Realtime publishers

"Leading the Conversation"

The Definitive Guide™ To

Windows Desktop Administration

SCRIPTLOGIC

Bob Kelly

Introduction to Realtimepublishers

by Don Jones, Series Editor

For several years, now, Realtime has produced dozens and dozens of high-quality books that just happen to be delivered in electronic format—at no cost to you, the reader. We've made this unique publishing model work through the generous support and cooperation of our sponsors, who agree to bear each book's production expenses for the benefit of our readers.

Although we've always offered our publications to you for free, don't think for a moment that quality is anything less than our top priority. My job is to make sure that our books are as good as—and in most cases better than—any printed book that would cost you $40 or more. Our electronic publishing model offers several advantages over printed books: You receive chapters literally as fast as our authors produce them (hence the "realtime" aspect of our model), and we can update chapters to reflect the latest changes in technology.

I want to point out that our books are by no means paid advertisements or white papers. We're an independent publishing company, and an important aspect of my job is to make sure that our authors are free to voice their expertise and opinions without reservation or restriction. We maintain complete editorial control of our publications, and I'm proud that we've produced so many quality books over the past years.

I want to extend an invitation to visit us at http://nexus.realtimepublishers.com, especially if you've received this publication from a friend or colleague. We have a wide variety of additional books on a range of topics, and you're sure to find something that's of interest to you—and it won't cost you a thing. We hope you'll continue to come to Realtime for your educational needs far into the future.

Until then, enjoy.

Don Jones

Copyright Statement

Chapter 1: Desktop Administration Overview

The latest computers, the fastest network, and the best-rated software can quickly turn from a good investment into a money pit without proper planning and implementation of desktop administration practices. So what is desktop administration? As you'll discover in this book, desktop administration is the methods and technologies used to deploy, configure, maintain, and track client workstations. It encompasses system deployment, user settings and data management, application management, security, support, and asset management. How you handle these critical facets determines the success of your workstation support. From a small office of 10 machines to a worldwide network consisting of tens of thousands of systems, the desktop administrator must address the same issues.

GartnerGroup studies have found that 82 percent of implemented management systems fail to meet user expectations. The reason is that the expectations of the solutions on the market are often simply unrealistic. There are no point-and-click solutions to deploying new systems or applications. There is no way of automatically backing up and managing user settings and data. Access to reports showing just the asset management, software inventory, or usage metrics you want to see isn't going to be displayed on your screen. You won't meet any of these mission critical needs *unless* time is taken to identify, plan, evaluate, test, and implement the right desktop administration solutions for your organization. As if meeting these mission-critical needs isn't motivation enough, let's delve into additional benefits of desktop administration.

> Keep in mind that implementing desktop administration solutions don't necessarily require new software purchases. You can take advantage of a number of solutions built-in to Windows and the resource kits. Although, commercially available solutions are often easier to work with and provide more robust feature sets, if your needs are basic, built-in tools and scripts might result in the right solution for you.

The Benefits of Desktop Administration

The potential rewards for implementing sound desktop administration practices can be great. Although we will cover the details of several critical areas of desktop administration throughout this book, the following benefits are common to all of these desktop administration components: a lower total cost of ownership (TCO), increased user productivity, and rapid and accurate system recovery.

Lower Cost of Ownership

The most significant benefit of desktop administration is the potential for reduced TCO. As the following list illustrates, there are many ways to measure the costs that may be saved by following desktop administration best practices.

- Increase support staff efficiency—For example, suppose that a hard drive in accounting just crashed. A user can no longer log on to the system after installing a new screen saver. It could be a very long day for the administrator who must address these kinds of problems—even a long week. However, with the help of an automated deployment solution, a machine could be rebuilt in minutes. In addition, policy and security restrictions could have prevented the user from breaking the workstation in the first place. With the right tools and skills, a support staff may solve (and hopefully prevent) far more problems than they could without them. A quickly solved, or prevented, support issue benefits both the support staff (who can move onto something else) and the end user (who can get back to work).
- Reduce Help desk costs—The ability to visually demonstrate a solution to a user can save the time it takes for you to explain how to perform a task over the phone, or worse, to take a trip to a remote office. A remote control solution provides this ability and is among the most common desktop administration tools implemented. For example, many organizations have implemented Microsoft's Systems Management Server (SMS) as a Help desk tool to make use of its remote control abilities.
- Increase support staff productivity—As you probably know, installing Windows, applications, and software updates across a large base of systems is tedious work. The time taken to visit each system and deal with user inquiries can greatly reduce the amount of work a support staff will be able to accomplish. By implementing desktop administration practices, tools or scripts may be utilized to greatly increase support staff productivity.
- Optimize software license requirements—Many inventory and metering software products provide a means of collecting statistics on application usage. This provides the ability to purchase only the updates that are needed, and to move licenses from those not utilizing them to those that need them rather than simply purchasing more licenses.
- Automate inventory to prevent failed deployments—Which systems fall below your standard configuration? Which systems do not meet the requirements for a software or operating system (OS) upgrade? A detailed hardware inventory can save you from doomed software and OS deployments that result from mistakenly targeting systems that do not meet the minimum requirements.

Increased User Productivity

Whether attempting to handle an issue themselves or waiting while you do it for them, users that aren't doing their work as a result of an IT concern cost the company. Although seemingly insignificant on an individual basis, the following scenarios occur often enough that they can affect a companies' bottom line. Solid desktop administration practices can prevent such situations, thus increasing user productivity.

Having users stand aside while updates and fixes are applied can be a frustrating inconvenience for those users and, more important, cost the company money. Without fail, an unexpected situation occurs and the half hour the user was going to have to wait quickly doubles. Automation of update and fix installations as well as proper testing of the automated installations significantly reduces the likelihood of unexpected problems and ensures that users return to productivity quickly.

In addition, solid desktop administration practices eliminate situations in which users handle installations on their machines. Such situations can result in problems because users don't correctly perform the installation. Rather than taking time away from their "real" work by messing with their systems, users can benefit from desktop administration that efficiently handles IT administrative tasks such as installing updates and fixes.

> Having users handle installations puts an undue demand on them and can have unpredictable results due to the varying degrees of technical aptitude users will possess. This situation will be frustrating to the end user as well as to management because of the inconsistent results and higher support costs.

Unfortunately, users who encounter a problem or are unable to perform a task are very likely to ask the people around them for help before calling the Help desk. Involving the users around them in solving system problems causes even more productivity to be lost. Enforcement of configuration management through a consistent Windows installation (that is, installation of tested software installation packages) and limiting a users' ability to cause problems will reduce the number of problems a user experiences and thus increase productivity.

As a technical individual, you might find it ridiculous that a user would want to hold onto an older, slower system. When faced with the option to replace their system, many users would rather stick with what they have than spend hours or days getting everything back to how they like it. Items such as wallpaper, color selections, and screen savers will be of great concern to many who will spend as much time as it takes—time that could be spent productively—to get their new system to look just like their old system. Thus, a script or tool to backup and migrate user data and settings can save the average user hours.

Finally, you can ensure that business computers perform only business functions through proper configuration and security management. The installation of unauthorized software and use of the games installed with Windows can be controlled through security restrictions and attention to Windows configuration options.

Rapid and Accurate System Recovery

Another benefit of desktop administration, automated and controlled system restores for systems or user data can save lost work and greatly reduce down time when repairing or replacing a computer. This desktop administration practice can be having an automated deployment process in place, having a method for restoring the applications that were there previously, or a combination of the two. The more automated the process, the faster and the less prone to inconsistencies it will be.

A system failure can mean extended down time and loss of data and settings in many environments. However, it does not need to be that way. Regular backups of user data and settings coupled with the speed of drive imaging software can mean that users are back up and running in minutes instead of hours or days. This process may be further engineered to automate the application of a drive image from a network management console. Taking this concept even further, you may hide a copy of the drive image on a local hard drive partition. In this way, the image may be instantly available without need for a lengthy network download.

When a system crashes and needs to be reloaded or replaced, even with days to perform the recovery manually, it might be impossible to get the recovery just right. Although many applications are common to most workstations, without an accurate software inventory, you might be forced to rely on the user to remember what other software existed on the system. With solid desktop management recovery and inventory practices in place, such doesn't have to be the case.

> ☞ Have you spent hours or days fixing a problem that is limited to a single machine? It happens, but if you have an automated system recovery process in place, you shouldn't spend more time on the problem than it would take to reload the system. Especially when an imaging and automated software deployment process has been implemented; in such a case, starting over can prove to be the most effective solution to many hard-to-solve computer problems.

When to Automate

As I stated in the introduction, small offices and worldwide corporations alike will face the same desktop administration issues. However, automation of desktop administration tasks might more clearly benefit larger networks. But what is a large network? When does spending the time and money to engineer an automated process outweigh the time and money spent to handle the task manually? Before we jump into a discussion of the scope of desktop administration, take into account the following considerations to determine whether automation will benefit your desktop administration practices:

- Regularity of administration tasks—If your organization reloads, migrates, or sets up a new computer more than a couple of times per week, the need to automate such a process becomes clearly visible.
- Speed of deployment—When taking into account additional application installation and administrative configurations, the time to install Windows might take as long as 6 hours. Imaging solutions can perform this process easily in less than an hour, with the possibility of reducing the operation to mere minutes.
- Configuration consistency—Even if Windows installations or upgrades occur just a couple of times a month, the benefit of consistency gained through implementation of an automated process has great value. Support issues may be more easily addressed if all systems start with a consistent installation of Windows.
- Lack of on-site technical staff—When there is little support staff available, having a system and application deployment process in place can help those less technical individuals reestablish a known good configuration. With an automated system deployment process is in place, people do not even need to know how to install Windows to get a system back up and running.

> I provided support for a daycare center where children would play learning games on computers. The kids would do all kinds of damage to the computers, mostly by turning them off while they were running. After fixing the computers for about 2 weeks, it became clear that I needed to provide a way for the non-technical people that worked there to fix the systems themselves. With a bootable CD-ROM that could apply an image of the hard drive, they were able to restore any problem systems themselves in just a few minutes. Of course, the kids still went through mice like candy, but troubleshooting Windows and the installed software took up no more of my time. This is just one example of how a desktop administration practice can not only save time and money, but also provide a permanent solution to a real-world problem.

Deployment Automation

Automating the deployment of systems and applications is critical to realizing both speed and consistency. Although speed is the most obvious and sought after benefit, the value of consistency should not be overlooked. You can thoroughly test and document the automated solution, providing invaluable data when addressing problems down the road. When all machines are not set up in the same manner, it is the inconsistencies (however small) that will make any issues that arise both hard to identify and to reproduce. If you cannot reproduce a problem, it becomes increasingly difficult, if not impossible, to identify. An automated deployment method ensures consistency.

Backup Automation

Particularly in situations in which user data and settings are stored locally, an automated backup implementation can save users the frustrations and delays that go with recovering them manually. Even when roaming profiles are in use, corruption of such profiles is not uncommon. An automated backup and recovery mechanism for maintaining a separate copy of the user profile can go a long way in the recovery process.

Inventory Automation

The decision to automate inventory tasks is greatly dependant upon how often, how accurate, and how detailed you need such reports. Even for a network of 30 machines, having an accurate picture of what is on the network can be invaluable, yet hard to obtain manually. Managing license requirements, deciding how many upgrades you need to purchase, and being able to report where software is installed are all difficult tasks to accomplish without the right tools. Software and hardware inventory can be detailed, up to date, and readily available with fairly little investment. Knowing what it is you have (hardware and software) is one of the first and most critical steps in performing large-scale updates and deployments.

Windows Management Instrumentation

When it comes to inventory, one of the most helpful new technologies is Windows Management Instrumentation (WMI). Microsoft has included more and more manageability with each release of WMI. Among other things, WMI provides a means of collecting an extensive amount of information about a system.

The WMI Object Browser, which Figure 1.1 shows, is one of the tools available in the WMI Administrative Tools. The WMI Object Browser lets you see all the WMI information available along with the associated values for the specified system (the local system is specified by default). Taking a look around in this tool lets you see just how much information is available to you.

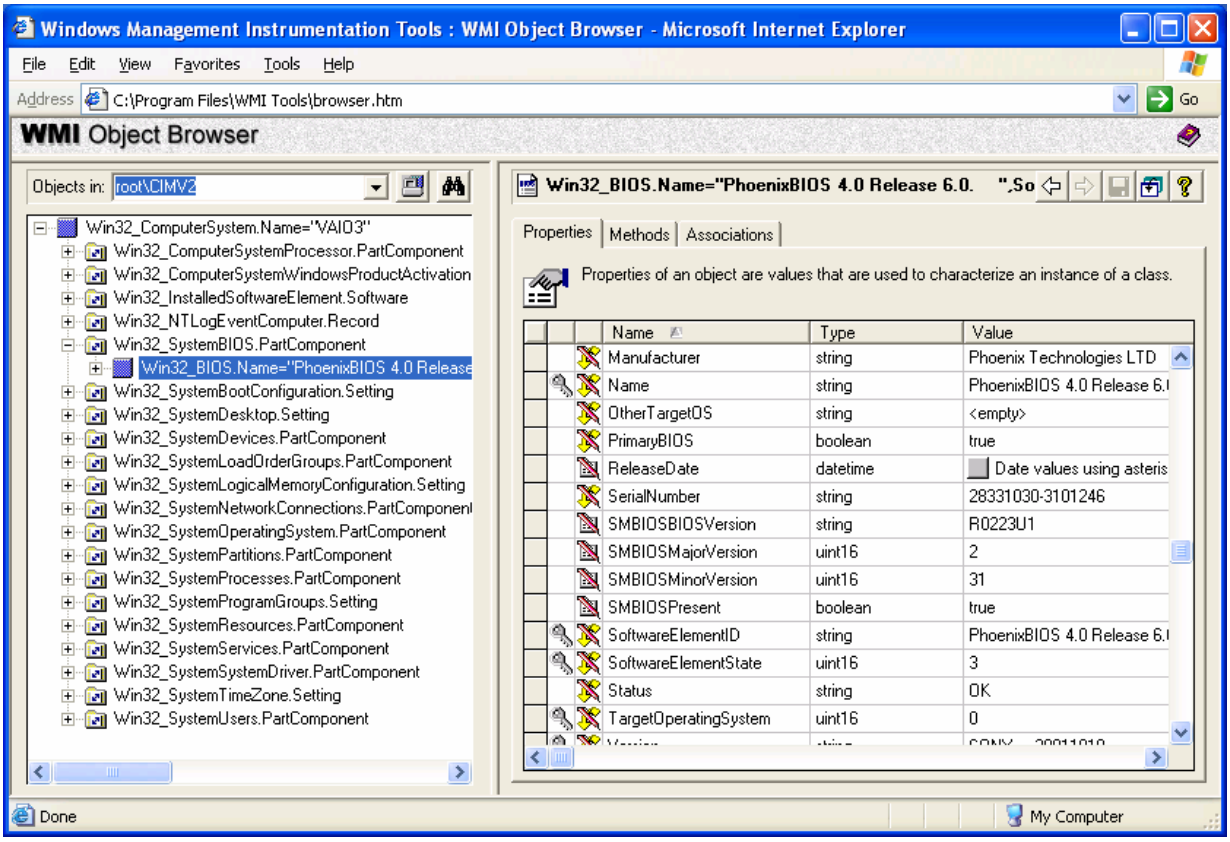

Figure 1.1: The WMI Object Browser.

> The WMI Administrative Tools are available at http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=6430F853-1120-48DB-8CC5-F2ABDC3ED314.

The Scope of Desktop Administration

As you're probably beginning to realize, desktop administration covers just about everything except servers and the network infrastructure. In this book, we will focus on the areas of deployment, management, and control of computers and users. In the following sections, I'll briefly introduce you to the areas of desktop administration that we'll discuss throughout this book as well as reference the chapters in which each topic will be covered in more detail.

System Deployment

The deployment of systems is the first step in desktop administration. A properly configured and well-tested Windows rollout is critical to the health of a system. Any problems with this initial system will need to be identified and corrected for all systems to which the deployment process is applied. Any issues not identified early might be compounded further with the addition of software and other configuration processes.

When deciding whether to perform manual installations, script automated installations (unattended installations), apply drive images, or make use of duplicated hard drives, there are several factors to consider. There are benefits and drawbacks to each of these methods.

- Manual installations—The benefit of manual installation is the ability to deviate from an established process when necessary. In contrast, inadvertently deviating from an established process is one of the primary reasons for automating installations. Manual installation is more costly in terms of time—it takes the administrator's time to perform the installation and takes time away from users' work while they wait.
- Unattended installations—Installation consistency and a reduction of time for the installation are benefits of unattended installations over a manual installation; however, unattended installations can still take more time than the other options. Microsoft provides a method for supporting unattended installations of Windows. There are also third-party tools available to assist in this process.
- Drive imaging—There are a handful of hard drive imaging products on the market that provide a means of distributing an exact copy of a hard drive across many systems. Many implementations provide support for both locally and remotely applying these images to a hard drive. Speed is certainly the primary reason for implementing such a tool on your network. (However, if you have multiple hardware configurations on your network, you might need multiple images.) The ability to include additional software applications in an image is also a key benefit to drive imaging, but taking advantage of this capability can further increase your need for multiple images. The drawbacks to this method include the fact that hardware incompatibilities can also make using a single image for all machines very difficult or even impossible. The more images you have to work with, the more difficult it will be to keep them all up to date and easily accessible. Additionally, in situations in which a network installation is to be performed, bandwidth restrictions may cause significant delays in applying something as large as a drive image. For this reason, it might not be possible to image more than a couple of machines at a time.
- Drive duplication—Hardware solutions exist that provide a means for cloning disks from one to another or from one to many drives at the same time. Speed is the primary benefit of this method. The negative aspects of drive duplication are the same as that of software imaging. In addition, the time spent removing and installing hard drives from systems must also be taken into consideration.

📖 In Chapter 2, we will explore these options as well as the tools available in more detail.

User Profile and Data Management

As I previously mentioned, user profile and data management is another aspect of desktop administration. You can manage user settings and data in many ways. Proper configuration of the default user profile and enforcement of settings via Group Policies can provide great control over a user's experience with applications and the OS. In addition, custom scripts and third-party tools such as ScriptLogic provide an ability to manage user profiles and data without some of the restrictions that exist in Group Policy. The following points highlight the user profile and data management practices that we'll explore:

- Default user profile—Creating and continuing to manage the default user profile can provide users with a customized default base of application and Windows settings.
- Implementing roaming profile size limitations—You can set a quota on the size of a roaming profile to avoid situations in which users create or drag large files into their profiles. Having too large a profile can result in extended logon and logoff events. In addition, this situation increases the likelihood of corruption.
- Folder redirection—Redirecting folders using the new capabilities offered by Windows XP can go a long way to keep the size of roaming profiles down. Even if you have not implemented roaming profiles, redirecting folders to a server location can still be very valuable in your efforts to have users store data on the server where it can be backed up.
- Troubleshooting—Simply delete the user's profile—that ought to fix it! We will discuss a more methodical approach than this to troubleshooting, including how to determine whether the profile is the problem and ways to identify where in the profile the problem might exist.

We will discuss these options and related topics in Chapter 3.

Application Management

Suppose you've completed your initial deployment of all systems just as they should be—with all the tools needed by everyone, before long a new version or update would rear its ugly head, which brings us to the next facet of desktop administration—application management. There a number of tools and methods available to avoid the need to visit each machine to perform updates and installations. As the following list points out, you can implement desktop administration practices to meet many of your application management needs:

- New software deployments—In a simplistic view, it is the software that allows many businesses to operate. The timely deployment of a mission-critical application can often be measured in dollars. There will always be newer and better software released, and a streamlined process for implementing these deployments must be carefully considered.
- Deployment of Windows application updates—Microsoft releases new hotfixes, security patches, and service packs every day. Although they might not all address the needs of your network, a great many of them will. The need to quickly implement patches, especially where security is involved, can be critical.
- Metering and usage metrics—Application metering provides the ability to monitor license usage, and in some implementations, even halt the execution of an application when a specified number of concurrent licenses are in use. This capability also produces a means of measuring the usage of applications. In many cases, you can extend this functionality to define restrictions and collect access times for specified Web sites or Web-based applications.
- Keep applications running—Help desk calls may be minimized and user productivity is maximized when the applications supporting their work run reliably. You can take advantage of the self-healing nature of Windows Installer (MSI) installations to avoid failure as a result of file corruption, deletion, or versioning discrepancies.

- Virus management—The presence of a damaging virus on one of your systems can quickly become the presence of a damaging virus on all of your systems. Ensuring that systems are up to date with the latest scan engines and the latest virus definitions can save your network from disaster.

These application management topics will be discussed in Chapter 4.

Desktop Security

In addition to system deployment, user settings management, and application management, desktop administration encompasses desktop security. Effectively managing desktop security can keep users from causing problems through improper configuration or installation of software. Solid desktop administration security practices include taking advantage of third-party logon solutions, file and registry access control, the restriction of user rights, and limited peripheral device access.

- Third-party logon alternatives—Policies and system restrictions that enforce complex passwords might not be enough in your environment. Access cards and biometric devices (for example, fingerprint, speech, and face recognition devices) are available to provide added security to control desktop access.
- File and registry access control—If you want all data stored on a server for ease of backup and recovery, simply directing users to put their data on that server might not be enough. Either intentionally or through ignorance, users might continue to store data on their local systems (despite your direction) unless their ability to do so is hindered through proper security configuration. Further benefits of file and registry access control include the ability to restrict software installation and the prevention of the proliferation of viruses.
- Restriction of user rights—Windows provides two distinct controls when it comes to security: permissions (to access and change the file system and registry) and rights. Rights dictate what a user is allowed (and not allowed) to do on the system. For example, a commonly restricted right is the right to change the system time. Although this limitation might seem an inconvenience to some, the effectiveness of security logs and audit logs can be greatly diminished by allowing users to perform such a change to the system.
- Peripheral device access—One way to take a step toward limiting exposure to viruses and stopping users from installing software is to limit the use of peripheral devices. One often overlooked way around device access restrictions is the potential for new devices to be easily attached to the system. A USB storage device might be automatically recognized by the system, bypassing your efforts to restrict peripheral device access.

Enforcement of security restrictions, particularly in the way of limiting or preventing the ability to customize or change a system, can cause some users to become quite upset. Remember (and perhaps remind users in a tactful way) that these are business computers and not personal computers.

I'll discuss desktop security in more detail in Chapter 5.

Desktop Support

Remote control products are one of the most commonly used desktop administration tools. Saving the time it takes to walk, drive, or fly to remote workstations has many obvious benefits. Through use of a remote control product, you may "take over" a user's desktop remotely and correct a problem or demonstrate how to perform an action without having to leave your chair.

The ability to remotely take over the desktop of a remote system is known as remote control. A call on the phone to report a problem can often lead to a manual visit, even if the user is able to articulate his or her problem well. For functions such as software installation and settings management, there are solutions that we will discuss that provide far more efficient and targeted solutions. However, when you need to perform a "one off" action with administrative privileges because you need to show them something (rather than try to explain it), or you simply need to see something with your own eyes, there is no tool more valuable than a remote control utility.

In addition, a remote execution service or tool, even the built-in Windows Task Scheduler, can be a very powerful solution. The ability to remotely execute a command could mean implementing a fix across thousands of machines without a need to visit them. Even when you only need to work with a couple of computers, this capability can save you from multiple trips of varying distances.

Another desktop administration support option is the ability to shut down and turn on computers over the network through Wake on LAN (WoL) support. By taking advantage of this capability, you can save power at night by shutting down all systems on the network and still perform those nightly maintenance tasks by turning on the necessary systems.

Finally, you can avoid problems by using proactive alerts for critical conditions and advise administrators of problems before the calls start coming in. Event notification and alert software can help you head off and even prevent problems.

We'll discuss these and related desktop support topics in Chapter 6.

Scripting Custom Solutions

Windows has provided increasing support for the automation of desktop administration tasks from the command line. Aside from those built-in to Windows, the Windows resource kits are loaded with even more helpful utilities. The Windows Script Host (WSH) provides Windows with support to run Visual Basic Scripting Edition (VBScript) and JScript. This support has resulted in an increased use of these languages by administrators. The simpler yet powerful KiXtart scripting language provided in the Windows resource kit provides easy-to-learn syntax for automating most administrative actions. In addition, you can implement customized scripted solutions, such as those highlighted in the following list, with little to no scripting knowledge by using tools such as ScriptLogic or WinBatch.

- Logon and logoff scripts—When you want to automate a task for a user, logon and logoff scripts are available to trigger such actions. The system executes the logon script automatically during the logon process, and this script can be used to perform user-centric actions mapping network resources such as drives and printers. Want to clear the Recycle Bin or perform a file backup of user profile data? The logoff script provides an excellent opportunity to perform such actions.
- Startup and shutdown scripts—When you want to automate a task for a system, the startup and shutdown scripts are available to trigger such actions. The system executes the startup script automatically during the startup process before the user is presented with the option to log on to the computer. The shutdown script performs its actions during the shutdown of the system. Both perform actions in the security context of the local system account, which allows for the triggering of actions users might not have the permissions to perform (for example, software installations or system level changes).

> Though support for logoff, startup, and shutdown scripts has only recently been introduced as a standard feature of Windows (NT and earlier do not provide native support), there are third-party tools that offer similar functionality, including ScriptLogic and ShutdownPlus from WM Software.

- Automation scripts—Migrating user data, creating accounts, checking for files, and most other repetitive tasks provide an excellent opportunity for automation via a custom script.

> I'll cover scripting custom solutions in more detail in Chapter 7.

Asset Management

Do you have enough licenses? Do you have too many licenses? You have four copies of a program—where are those copies installed? Are they being used? Keeping a tight grip on asset management can provide you with quick answers to these questions, helping to save money and avoid potentially steep fines for violating license agreements (which are far more than the cost of implementing an asset management solution).

Often, you review software inventory and licensing reports to make sure that enough licenses have been purchased. However, just as costly and sometimes less apparent, is that you have purchased too many licenses. If software is not being used, why pay for it? You could save a substantial amount of money by taking licenses from those who do not need them instead of just continuing to buy more. When it comes time to purchase an upgrade or new version, you can use solid asset management processes to determine how many you need to purchase.

> In Chapter 8, we'll explore asset management.

The Politics of Desktop Administration

An oft-overlooked aspect of desktop administration is politics. It can be frustrating to be in a situation in which you cannot make a change you feel is necessary as a result of the political setup in your company. Determining who has ownership and control of the network and its resources can cause problems in some organizations, particularly in situations in which there are remote sites that employ their own administrators. In this section, I'll explain how establishing responsibilities and providing appropriate access can go a long way in minimizing problems between business groups. In addition, you will be better able to implement strong desktop administration practices if you consider how decisions will affect users, how control is delegated, and how you plan to work with users to ensure success.

Making Decisions that Affect Users

Someone or some group of people must have the final say in what can be done on a network. If the ability of users to install software on their systems is causing problems, can you simply take that ability away? If you want to take control of a workstation, do you have to ask first? It is these types of issues that must be identified and documented by each organization.

For example, any limitations imposed on users are very likely to cause complaints. In such situations, it is not normally the role of the administrator to make the decision to implement limitations, but simply to recommend and then enforce those decisions. Having this type of delegation ensures that users have an outlet other than the Help desk to voice non-IT complaints.

If users want administrative control over their workstations, they might come to you, the administrator, for that access. Without an established process in place, it might be difficult to point the user elsewhere for permission to make such a change. This situation has been the downfall of many networks. Without a reason to say no to users' request for administrative control, over time, there can easily become too many individuals with administrative control and configuration management is lost. You can avoid such a situation by establishing how decisions are made that affect users' systems.

> ☞ Particularly in environments in which users have had unrestricted control of their systems, change in the way of implementing restrictions can mean loud complaints. Be sure to have management completely onboard before implementing this type of change to avoid becoming very unpopular very quickly.

Delegation of Control

As a desktop administrator, what you control and how much control you have will vary greatly depending on your organization. Do remote sites have their own administrators? If yes, what roles are performed onsite? Responsibilities must be agreed upon and then laid out on paper to avoid problems down the road.

Ideally, the same way users are given enough access to their systems to perform their jobs and no more, administrators should have access to perform only the tasks they must. Everyone that requires permission to back up a file, access a certain share, or join a computer to the domain does not necessarily need to be granted membership to the Domain Administrators group. Creating groups for administration roles and providing only what is required to those groups is an effective method for the delegation of administrative controls.

Depending upon the versions of Windows deployed in your organization and the presence of Active Directory (AD), there are several options available for you to define administrative roles. The following sections detail three options: via Windows NT domains, using AD, and through third-party solutions.

NT Domains

Domains represent administrative boundaries at which security policies and settings (such as administrative rights, security policies, and access control lists—ACLs) can be administered separately for each domain. Administrators of a domain have rights to set policies only within their domain (by default). Therefore, different domains in your organization can each be managed independently.

NT 4.0 has a limit to the number of user accounts that a directory can store. Therefore, in large computing environments, it is often necessary to create and manage several domains, each with its own directory of user accounts. Domains are typically organized into two types: master domains (used to store user and group accounts) and resource domains (used to store files, printers, application services, and so on). This multiple-domain computing environment is known as a multi-master domain model, which Figure 1.2 shows. A multi-master domain model means that resource domains need multiple trust relationships with all master domains. These trust relationships allow users in master domains to access resources in resource domains.

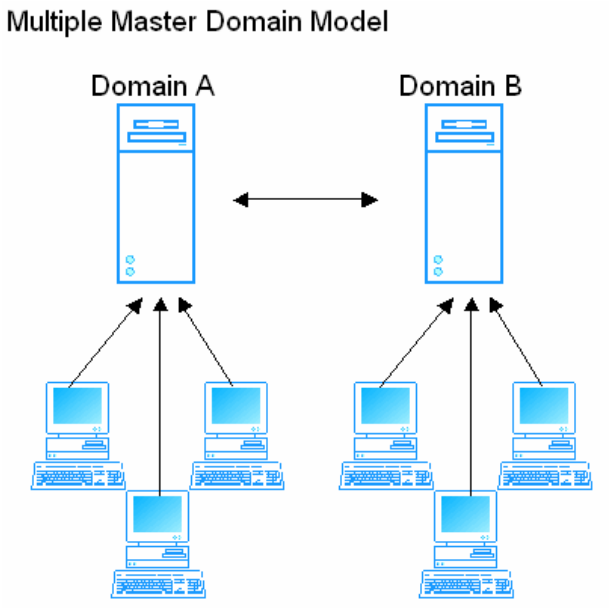

Figure 1.2: Multiple master domain model (trusted domains)

AD

AD replaces the need for multiple domains by providing the ability to store a large capacity of user, group, and computer accounts. With AD, administrators can consolidate all accounts (which formerly had to be stored on master domains) and all resources (which formerly had to be stored on resource domains) across domains into a single domain. To further maintain logical groupings of objects for administrative purposes, you might group computers and users into organizational units (OUs) within the domain.

Third-Party Solutions

Without the benefits of AD, assigning rights to administer some users means assigning rights to administer all users. However, there are tools available, such as Aelita Software's Enterprise Delegation Manager, that provide a more granular structure for defining administrative roles. Though Win2K addressed many of the limitations inherent in delegation of control within an NT environment, the available third-party solutions enable the creation of a Win2K-like directory in an NT network.

Working with Users

When things go wrong, there is a tendency for users to point to the unknown as the cause. The magic you perform with your desktop administration tools may be seen as the malefactor behind errors, latency, and just about anything else that is difficult for a user to explain. Simple practices such as keeping users informed and providing explanations will go a long way to ensuring successful desktop administration as well as office harmony.

> Perhaps it goes without saying, but even more important than education is that you perform the planning and testing necessary to minimize the potential for causing problems for users. A single failure can brand the desktop administration team with a bad reputation that might be difficult to recover from.

When a deployment is planned, notify users beforehand, then advise them of your success or any problems you encountered that might affect them. It is a common mistake to keep users out of the loop. Doing so ultimately means that they end up hearing about the occasional failure and not the successes of your work. Some administrators worry that problems will be blamed on management tools and remote updates if the users are informed of such installations. However, it is very likely problems will be blamed on desktop administrators even when operations are not being performed. Build a positive reputation by educating users and taking credit for your successes.

> A weekly or monthly newsletter may be overkill; however, normal announcements about what is happening as it happens can open communication with users. Consider setting up a simple Web page that users can visit to discover what's going on and what is scheduled for the near future.

In particular, a capability such as remote control can be very intimidating to users. Does this mean that someone is watching everything I do? Decide what to tell users about your tools, but avoid saying nothing at all. Put some thought into this process—for example, you might want to prevent paranoia by explaining tools but ensure that you don't eliminate the slight fear of software usage reporting (metering)! It might be a good thing that users worry about being discovered playing Solitaire for half of the day.

Factors in Effective Desktop Administration

Company politics aside, your ability to implement effective desktop administration practices might be limited for several reasons. You can address some factors, such as standardization, but others, such as geographic dispersion, are simply something you must learn to work with. In the following section, we'll explore the possible challenges you face as a desktop administrator.

Standardization

Whether your organization lacks standardization is the number one factor when measuring the complexity involved in desktop administration. Such is particularly true when dealing with the deployment of applications and updates. Varying OSs and hardware can multiply the effort required to engineer a deployment.

> ☞ Almost without exception, the more you buy the less it will cost. Choosing one software package that satisfies your users will be far less costly than making several smaller purchases from different vendors.

The more versions of Windows on your network, the more difficult certain desktop administration tasks will become. Particularly in the area of application deployment, you will need to develop, test, and manage the distribution of packages separately for each OS on the network. Measuring the minimum requirements of software will also require much more attention in environments in which multiple OS versions and service pack levels must be considered.

Chances are that most every workstation in an organization requires largely the same applications and utilities on each. By establishing a common baseline from which to start all workstations, you can more easily realize an increase in the consistency between systems. Whether through documentation, unattended installation, or imaging, start all users out with an identical common workstation load and then install additional software on an "as needed" basis.

> ☞ With a common software baseline established, Help desk personnel and users alike may become proficient with the same tools. Users are better able to assist one another and training programs may be instituted that benefit the largest possible number of users.

Determining the minimum requirements for versions of Windows and its applications may not be too large of an issue for many organizations. However, if a drive cloning (hardware- or software-based) method of system deployment is in place, consistency in the hardware on the network can greatly reduce testing requirements and the number of images required. With NT and earlier systems, even a different model network card could mean the creation of a new image. The plug-and-play (PnP) capabilities of newer versions of Windows provide less of an issue with inconsistent hardware, but the problem has not gone away. Even when creating an image with Windows XP, the reference and destination computers must have compatible Hardware Abstraction Layers (HALs).

We'll discuss standardization in more detail in Chapter 2.

When purchasing hardware, ask vendors about their update cycles. In some cases, the same model of a computer is shipped with slightly different hardware configurations. Even a slight change has the potential to cause you considerable troubles with imaging or accounting for available device drivers. Some vendors address this potential problem by offering computers aimed toward businesses. For example Dell offers the OptiPlex series for business and Dimension series for home use. Hewlett-Packard (which acquired Compaq) offers an Evo series for business use and a Presario series for home use. Each vendor provides documented assurances that hardware configurations will remain consistent in a given shipment of systems or for a specified number of months.

Geographic Dispersion

The ability to walk down the hall and put your hands on a system is easily taken for granted unless you must account for a single remote site (often with decreased bandwidth and stability). Remote sites can be the cause of much consideration and engineering. Often separate desktop administration policies and procedures are put in place to handle these environments, sometimes resulting in a reduction or loss of configuration management as well as an increase in the time spent planning and supporting these remote systems.

In Figure 1.3, we see a few desktop administration problems to overcome. First the wide area network (WAN) connection (the remote Tampa site) might be slow or costly to utilize, and second, the remote site does not have a local server. This scenario is not unusual for sites with few workstations and little to no onsite desktop administrative support. Virtually all network-intensive services may be subject to reduced network speeds. Windows installation, software deployment, profile management, and other desktop administrative tasks will often need to be altered for such remote sites.

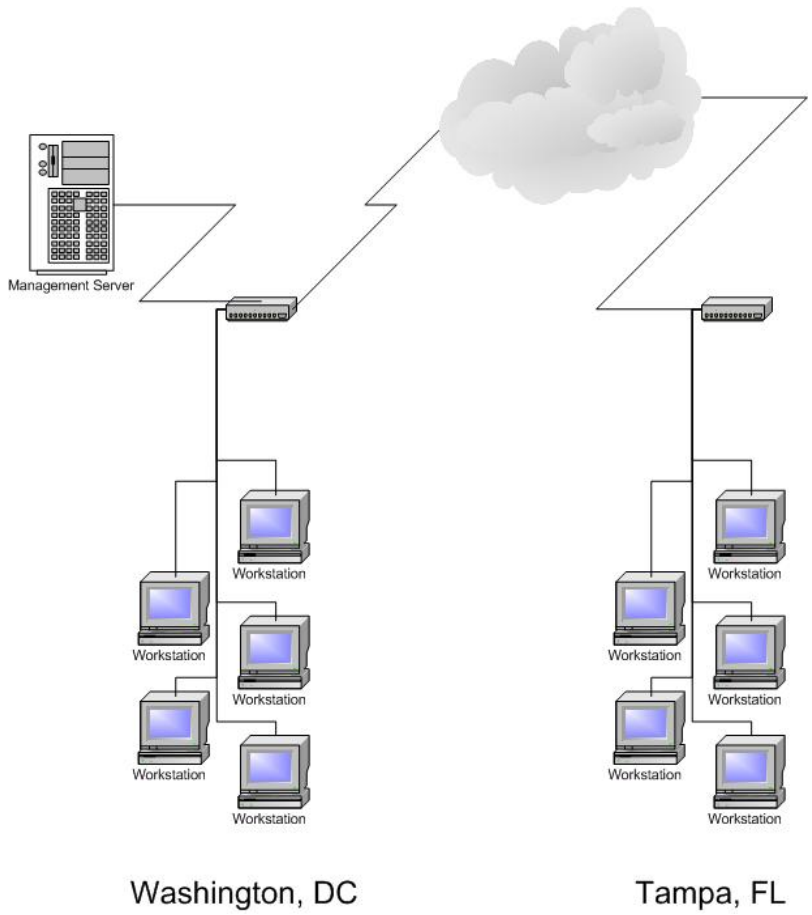

Figure 1.3: Geographic dispersion and connectivity considerations are elements that can add significant levels of complexity to desktop administration.

Dedicated Staffing

The larger the network, the more likely a dedicated desktop administration staff is present. A staff able to focus and build proficiencies will clearly have an increased opportunity to succeed. In situations in which desktop administration is one of many tasks you must perform, desktop administration operations might take longer to perform and risk the possibility of being forgotten completely.

Budget

The first limitation to come to mind for management, and thus for the desktop administrator, is usually budget. One way to ensure that your desktop administration tool purchase request is approved is by drafting a business case for the request (in contrast to an IT case—usually that such tools will make your life much easier). If you can demonstrate that the $5000 management software package that you want to implement will save many times that amount in just the first year, it is far less likely your request will be denied. If your purchase request for administration tools is considered too great a cost to spend and you are left to spend weeks or months developing the same capability with free tools and scripts, consider the situation a windfall—many careers have been built on the development of customized solutions to desktop administration issues.

Connectivity

As I mentioned in the geographic dispersion discussion, remote and disconnected clients significantly increase complexity when it comes to desktop administration. Deploying new and updated software and enforcing software-metering policies are a challenge when working with notebook computers that dial up only a couple of times a week. Rest assured, there are products that tackle these and most every problem you will encounter—including the pesky problem of connectivity.

Process

On one hand, if your organization doesn't have a process for implementing changes, problems such as undocumented and untested software is installed or approved and needed software isn't implemented. On the other hand, having too detailed and lengthy a process can slow (sometimes indefinitely) the ability to effect change on a network. This need for a balanced process management plan is sometimes overlooked as a factor in effective desktop administration. You need a process, such as the one that Figure 1.4 illustrates, that ensures that ad hoc changes as well as months of meetings, boards, and reviews are avoided.

Obviously, this situation is yet another in which the size of a network takes a significant role in a sensible approach. If a change affects very critical systems or a very large number of systems, a more detailed and rigorous process for test and review becomes a necessity.

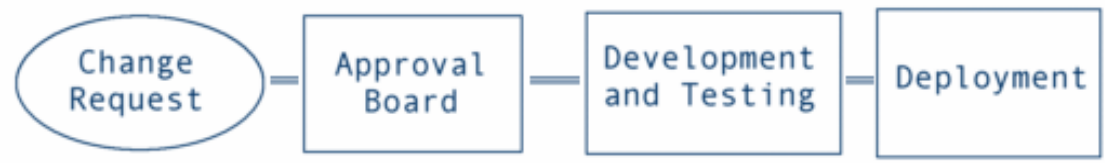

Figure 1.4: Example of a simplified change and deployment process.

Security

Security is also a factor in how effectively you can implement desktop administration practices. Some organizations lock down their desktops so that users can do no more than run their required software. Others provide everyone with local administrative access to the system to avoid errors or warnings when even the most basic, built-in security restrictions are violated. What a logon script can accomplish is an indicator of how security can affect your ability to administer the network. With default security implemented, a logon script can do very little outside automating user settings and environment. Changes to the machine must usually be handled via an administrative service account or the local system account. In this book, we will later discuss many of the utilities and third-party solutions available (such as SMS and ScriptLogic) to address the security limitations inherent of the logon script, which run under the security context of the local user.

For those that work in a secure environment, it might be hard to imagine that there are networks in which users have full control of their systems. However, it is not uncommon for users to be granted administrative access to their systems so that they may handle administrative functions themselves. Obviously, there is a significant potential for damage to the system, both by users who do not know what they are doing and users who *think* they know what they are doing. However, despite the potential for problems, users who *do* know what they are doing can reduce Help desk calls and prevent time wasted waiting for support staff to make a change on the user's behalf.

In contrast, some networks are locked down beyond the default settings to protect users from themselves and to maintain control of system configuration and content to the greatest degree possible. Keeping users from being able to modify certain settings can account for a direct decrease in support costs. Although this security method might sound favorable, there is a definite downside to this approach. Significant testing must be performed on new applications and updates to ensure that they are able to operate in the locked down environment. Deployment capabilities may also be hindered and will result in an increased complexity in many areas of desktop administration.

Desktop Administration in Action

Although these factors are very real, perhaps a few examples of how you'll actually use desktop administration would be helpful. The following situations provide a more situational overview of problems you might face and how desktop administration tools and practices will help you overcome them.

Help Desk Calls

One of the most often-employed desktop administration solutions is the Help desk. Whether you are implementing tools to help engineer solutions for a Help desk or you are the Help desk, the problems you face can be broken down into just a couple of categories.

Probably the most common Help desk call is for help with a specific application. Not necessarily because there is a problem, but because the user simply does not know how to perform the action he or she requires. The frequency of this type of problem will vary depending on the level of technical knowledge of your user base, but rest assured that every administrator has at least a handful of individuals that just like to ask questions. For such individuals, a remote control tool can save you considerable time. Show the user how to accomplish the required task while talking them through it on the phone to avoid the trip to the user's desk and the inevitable other distractions that come with that trip.

Requests for software installations and upgrades are yet another common type of Help desk call. The ability to remotely satisfy this request using a documented and tested installation package is invaluable. With such a solution in place, the support call can be reduced from a half hour or longer to just a couple of minutes.

Handling New Systems

As your organization expands and grows, you will encounter a constant need for the deployment of new systems. Whether you have a staging area in which such systems are configured beforehand or the new computer is already at the user's desk, new system rollouts can be a lengthy endeavor.

Requests to add a new machine or move an existing one can be among the most time-consuming desktop administration tasks if your organization lacks an automated process. Having a new employee wait while a fresh machine is built from the ground up would be the worst way to handle such a situation. Here, the new employee gets a bad impression of the way things work and the administrator spends hours or even a full day away from his or her other tasks. An automated process for setting up a machine and applying required software goes a very long way to reducing the impact of this demanding administrative chore.

Outsourcing Desktop Administration

There are a growing number of management service providers (MSPs) that provide desktop administration services as a monthly service. With the capabilities of remotely managing and supporting systems, it is easy to see how such operations could be handled as an off-site service. The more complex the operation, the more cost-efficient it becomes to outsource such an operation.

There are positive and negative aspects of this approach. Regardless of whether you outsource or internally handle desktop administration, it is important to understand the best practices and methods available so that you can be sure that your organization's desktop administration practices (even if they're outsourced) make sense for your company.

The Benefits of Outsourcing

The following points highlight the considerations for outsourcing desktop administration.

- Reduce implementation risks—There is risk in taking on the implementation of remote and automated solutions. Many such systems take a very long time to plan and implement, others might fail to perform the implementation altogether. An MSP focuses on such deployments on a regular basis, which translates into an ability to perform implementations of these systems with a much higher degree of success.
- Gain access to subject matter experts—A company that focuses on desktop administration issues and performs such tasks for multiple networks will have a proficiency in this area that would be difficult to develop in-house.
- Known costs—Desktop administration is an afterthought for some organizations that, in turn, apply little resources to its success. This shortcoming often leads to failures, which increase the costs associated with desktop administration. With a service approach, the costs and services are laid out and are therefore easier to plan a budget around.
- Process integration speed—Planning, implementing, and administering desktop administration tools can be challenging and time-consuming. A company that does so for multiple clients, such as an MSP, will be considerably faster implementing desktop administration tools and processes than a group of individuals who are doing so for the first time.
- Focus on core competencies—In the IT field, a certain specialization of skills can be a necessity. With technology changing on a constant basis, it is easier for many companies to outsource some part of their IT operation to a firm with expertise, particularly in an area as challenging as remote management.

The Drawbacks of Outsourcing

If the benefits listed sound too good to be true, that might very well be the case. Like every other decision, you must consider the drawbacks of outsourcing.

- Level of control—By out sourcing desktop administration operations, you inevitably loose some control of the network. To meet certain service level agreements (SLAs), policies may be required that restrict your ability to perform these tasks.
- Security—Though you would never take on such a service without reasonable assurances regarding security, there is still a controlled security hole being opened when you outsource administration. Desktop administration services may involve remote control, imaging, and application deployment—all of which are very damaging in the hands of the wrong people.
- Staffing conflicts—If you are reading this book, there is a good chance that your job could be threatened by outsourcing desktop administrative tasks. It will be critical that the existing IT staff be either moved on to more advanced work or that the outsourcing service work with the existing IT staff as a secondary authority. Placing powerful day-to-day operations in the hands of an external company might cause friction between internal and external IT staff if not handled in a very delicate manner.

☞ For a directory of companies that provides these services visit http://www.msplocator.com.

Summary

In this chapter, we covered some of the basic concepts behind desktop administration. We discussed the many benefits of implementing sound desktop administration practices including a reduced TCO, an increase in user productivity and the ability to realize rapid and accurate system recovery. In addition, we touched on whether the size of your company justifies automating desktop administration practices. As we explored, desktop administration encompasses a large scope of tools, tasks, and practices, and I let you know which topics will be covered in future chapters. I provided information about the need to establish a process to determine how decisions are made that affect users' systems, how control is delegated, and how users will be informed and educated about desktop administration activities. Finally, we considered the factors in effective desktop administration and whether avoiding these potential pitfalls through outsourcing makes sense for your organization.

In the chapters that follow, we'll delve deeper into the detailed aspects of desktop administration, and discuss the technologies and tools available to meet these mission-critical needs. Specifically, in Chapter 2, we'll discover system deployment considerations, options, and tools.

Chapter 2: OS Deployment

In the last chapter, we touched on each of the primary areas of desktop administration. In the remaining chapters, we will drill down into each of these topics in much more detail. This chapter covers the first of these subjects, operating system (OS) deployment. The deployment of new systems is the foundation on which the stability of the workstation will be set, no matter what the makeup of that new system—a new computer, a new version of Windows, or simply a new configuration. A poorly built image, a bad script, or poorly researched configuration options can result in wide-scale problems when you consider that all of your systems might reflect these same troubles. In this chapter, we will discuss some best practices and methods for handling the deployment of new systems, including the benefits and drawbacks of each.

The Workstation Baseline

A baseline is an established, common starting point from which all workstations begin. Regardless of whether you have a manual or automated process in place, there is normally some process that defines the makeup of your initial workstation installation and configuration. The typical makeup of a workstation baseline is the OS and applications common to all systems. From this point, things might change considerably, with each department (or even individual workstation) getting the additional software packages required for those users to accomplish their jobs. However, the installation of a consistent starting point, automated or not, is the most common first step in the workstation life cycle.

Benefits

Planning and discussing the many options that make up the installation of Windows and its common software is an important step in establishing a common baseline. With planning, a baseline allows for a reliable and consistent starting point, which brings the following benefits: increased reliability, faster deployments, and easier troubleshooting.

Increased Reliability

Through the consistency that comes with the implementation of a baseline configuration, you can establish a firm starting point for all workstations. When all systems start from this proven and tested configuration, there is a decreased likelihood of problems that result from misconfiguration of the system. A well thought out and documented baseline configuration can be directly attributed to a lower number of support calls (and therefore decreased user downtime.)

Increased Deployment Speed

The amount of time and effort needed to get a workstation to a desired state is greatly reduced through the implementation of a workstation baseline configuration. Whether the changes that follow are automated or manual, a repeatable proven process of getting the workstation to a baseline state will reduce overall setup time. In a case in which a duplicated or imaged hard drive is being used in the deployment process, this benefit is more obvious. However, even in cases in which the deployment process is merely a detailed check list, administrators will naturally become proficient in handling this baseline portion of the setup process.

Ease of Troubleshooting

Reproducibility is a key factor in the ability to troubleshoot and solve any computer-related problem. Being able to mimic a problem in a lab environment is often crucial to identifying and testing potential solutions. With a baseline implementation in place, you can more easily troubleshoot problems; you simply apply the baseline installation and add the same delta of additional software packages, and you have a good chance of reproducing a user's problem. Additionally, solutions to problems might be more commonly applied to all workstations. If a problem on one machine is discovered, it is likely the same resolution will apply to other systems as well.

Drawbacks

To be fair, we will also explore the other cons. As with any solution, even the use of a baseline can have negative aspects in certain situations. For example, a problem for one is a problem for all. If you don't get the baseline correct, you risk a problem on all machines. It is for this reason that extensive testing is a necessity in the implementation of a baseline by any means (automated or manual).

In addition, manual repetition can be a problem. When the process of establishing a baseline configuration is not automated, repetition can lead to carelessness. A check list is your best defense against manual misconfigured systems, but an administrator can still grow to feel they have memorized the process and ignore the check list. Forcing administrators to turn in such a check list as a deliverable document as part of the deployment process can result in administrators checking off items blindly simply to produce the required paperwork. Automating the development of your baseline through scripts or a combination of a manual check lists and execution of scripts will also help to ensure consistency in establishing your baseline.

Baseline Components

Planning what should be included in your baseline is obviously a very important step in the process. However, deciding which items to leave in the installation and which not to include is not always an obvious decision.

Table 2.1 shows a simplistic view of what might make up the software requirements for the development, engineering, and graphic art departments at your organization. The baseline configuration to be established here would be all but the last application in each list. Based on this planning document, you might enjoy the benefits of a baseline for these common components and use your application deployment method of choice to add the necessary delta in order to satisfy the requirements of each user or group of users.

Developers	**Engineers**	**Graphic Artists**
Microsoft Win2K	Microsoft Win2K	Microsoft Win2K
Microsoft Internet Explorer 7	Microsoft Internet Explorer 7	Microsoft Internet Explorer 7
Microsoft Office XP	Microsoft Office XP	Microsoft Office XP
Adobe Acrobat Reader	Adobe Acrobat Reader	Adobe Acrobat Reader
Virtual Studio .NET	**Visio Technical Edition**	**Adobe Photoshop 7**

Table 2.1: A sample baseline software makeup.

What to Leave In

You can avoid having to deploy and configure multiple instances of the software that is going to everyone (and therefore avoid the inherent risks that come with that process) by simply including the software in a baseline image or unattended scripting process. Even if your deployment process is entirely manual, you should still plan to create a baseline that includes all common software to avoid the need to return to workstations more than once.

What to Leave Out

Due to the popularity of drive imaging or cloning software, many client/server applications that have their own management consoles will provide specific recommendations regarding how they should be handled. These applications might not easily lend themselves to inclusion in a system baseline image or script, and you should handle them via the provided management console. Pay special attention to software with client agents or services such as antivirus software and desktop management tools.

Common Pitfalls to Avoid

As desktop administration becomes more and more transparent to the end user, its complexity can increase for you, the administrator. As a result, there are a growing number of pitfalls to watch out for. We will cover some of these issues in this section.

MSI Source Resiliency

Windows Installer (MSI) introduces a new element to consider as a desktop administrator—installation source availability. When software is installed, Windows Installer will check its installation location for installation files if and when they are needed in the future. If a new feature needs to be installed or if missing files need to be reinstalled, Windows Installer will look to the software's installation source to obtain these files. Thus, when creating images and duplicate drives, be sure that the location where Windows Installer will look is available.

When installing MSI setups from a CD-ROM, mapped drive, or UNC path, you must consider that this path is the one that will be checked by Windows Installer. Do you want all systems looking to the same network share for source files? Did you install Microsoft Office from the CD-ROM when you built your image? Failure to plan for a resilient MSI source might result in user prompts for setup files that will likely result in an increase in support calls.

For an overview of Windows Installer technology and recommendation for how to deal with issues unique to Windows Installer, see *The Definitive Guide to Windows Installer Technology for System Administrators* by Darwin Sanoy and Jeremy Moskowitz available from a link at http://www.realtimepublishers.com.

Non-Essential Data

Another pitfall to watch out for: Avoid inclusion of user- and machine-specific data in your baseline. Although this issue is more of a challenge when you're imaging or duplicating drives, be aware that your user profile might contain information that need not be deployed to all systems. Take a good look around by searching the file system (pay special attention to profile directories and temporary folders) and the Windows registry for your username. The Windows built-in search function, which Figure 2.1 shows, allows you to look for text within any file in the specified path. By specifying all files on all drives and entering your username as the text to search for, you might uncover some files that you do not want sitting on every new machine you roll out. In some environments you might want to avoid including the company name or network name in the baseline image.

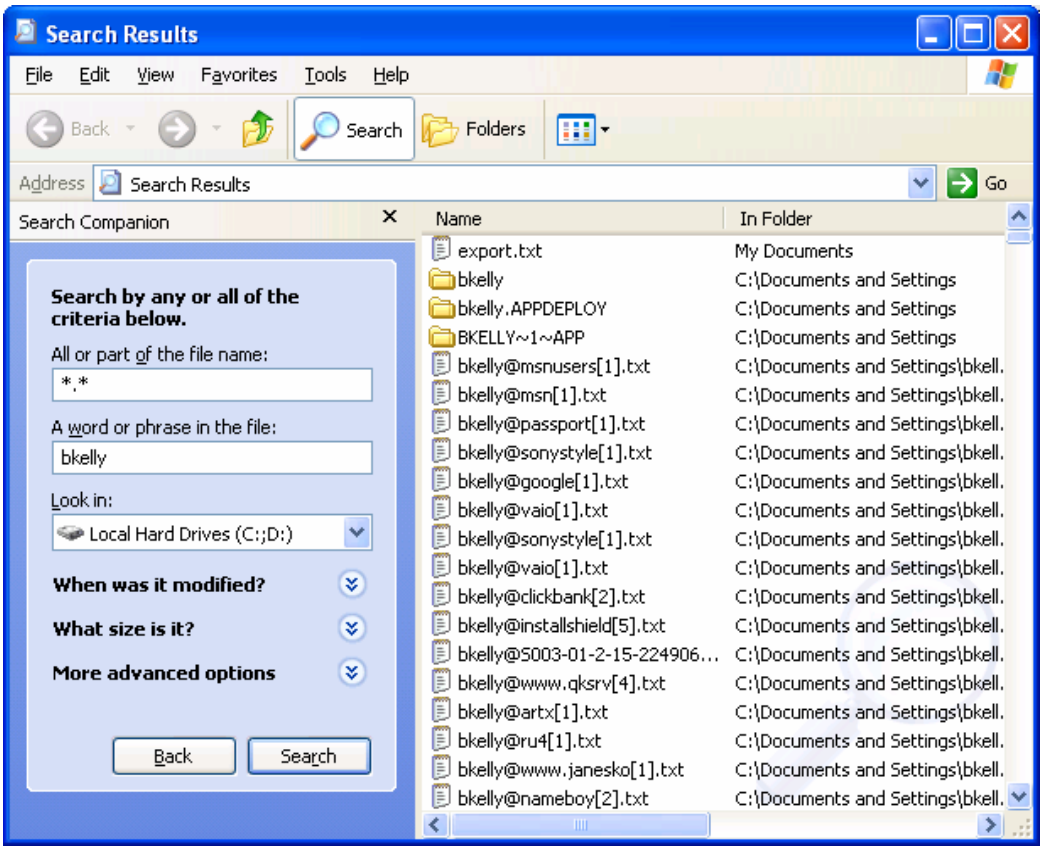

Figure 2.1: Windows' search view.

Classified or Company Proprietary Information

Accidentally including classified or company proprietary information in your baseline might be more than undesirable—doing so could get you in big trouble! If during the development of your image you have opened or accessed classified or company proprietary information, remember to clean up temporary files and cached data before finalizing an image or even before walking away from a manual setup. Though you might have application-specific areas to focus your search for such information, a good start will be to empty the Recycle Bin, delete cached Internet files and temporary folders, and remove unnecessary profiles.

Globally Unique Identifiers

Globally unique identifiers (GUIDs) are being used more and more these days. Be aware that if you include them in your image, they will no longer be globally unique! Client/server applications (client agents or services that communicate with a server console or management system) commonly use GUIDs to identify their client-side systems (such is particularly true for antivirus and desktop management client software). These unique identifiers are often established during installation and might be used to uniquely identify their clients on the network. Have a look at the documentation for any client/server software to ensure that GUIDs will not be a problem for your baseline.

Including Microsoft's SMS client in an image is another common way that a GUID might be inadvertently introduced. SMS assigns a GUID to a client system upon installation, and it is this number that identifies the computer to SMS. This situation can cause a lot of problems for SMS (software installing on the wrong computer, inventory information merged with other clients, and only the last computer to use the GUID will be listed in the management console). One way to end up with the same GUID assigned to all clients is to include the following registry entry in your image: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\SMS\Client\Configuration\Client Properties\SMS Unique Identifier. In addition, there are two files in the Windows System directory that can result in duplicate GUIDs: SMSuid.dat and SMScfg.ini.

The best way to avoid duplicate GUIDs with SMS is to not include the SMS client software in your baseline. SMS and most other client software agents provide their own method of deployment. Alternatively, you can make use of whatever application deployment process you have in place to install SMS as you would any other item that is not part of your baseline. This particular problem is only applicable to imaging and drive-duplication deployment methods. If you're using a manual or scripted process to roll out your baseline, you might be able to automate the installation of the client software so that it might be installed (and generate its own GUID) as designed.

> ☞ Non-unique GUIDs are such a common problem that Microsoft provides newuid.exe, a tool to help remove duplicate GUIDs after they occur. You can read more about dealing with duplicate GUIDs in SMS in the Microsoft article "Managing Duplicate Microsoft Systems Management Server Unique Identifiers."

Profiles

In addition to the fact that leaving user profiles in your baseline takes up unnecessary space, failing to do so might result in personal files on every machine. To avoid this pitfall, on the baseline systems, right-click My Computer, select Properties (or launch the System applet from Control Panel), and use the User Profiles option to delete user profiles from the computer, as Figure 2.2 illustrates.

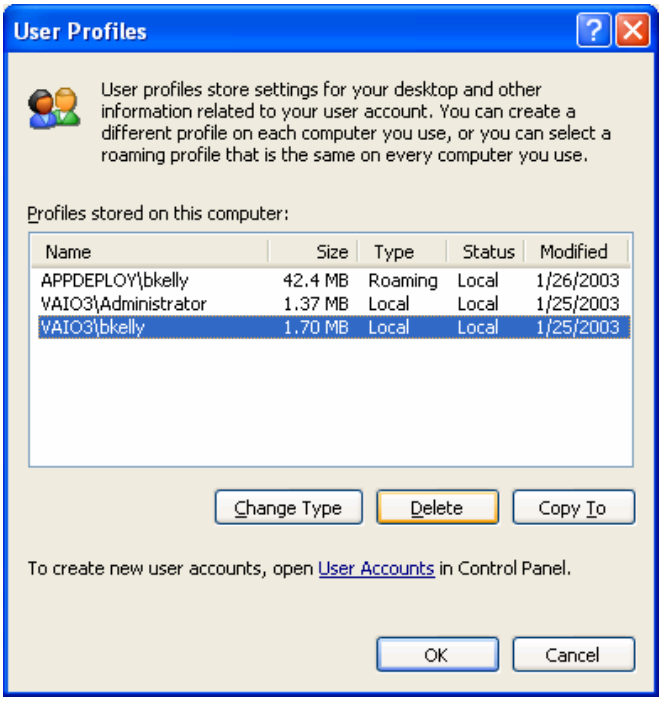

Figure 2.2: User profile dialog box of the Windows System applet.

> Simply deleting the user profile directory (C:\Documents and Settings\<userid> for Win2K and later and C:\WinNT\Profiles\<userid> for Windows NT) does not effectively remove all traces of the user profile.

To automate the process, use the DelProf.exe utility from the Win2K resource kit. However, this task is usually one that you perform manually before finalizing the creation of a duplicate disk or image. Therefore, using the previously mentioned method (illustrated in Figure 2.2) should work fine for most situations and will allow Windows to properly account for the profile and all references to it.

Security Identification Numbers

Users, groups, and computers are referred to by the security system of NT and later using their security identification numbers (SIDs). As Figure 2.3 shows, Windows does not see me as *bkelly*, but as S-1-5-21-592864255-1439916374-227697207-1031. It is important to the Windows security system that these user SIDs are unique so that Windows does not confuse one account or system with another.

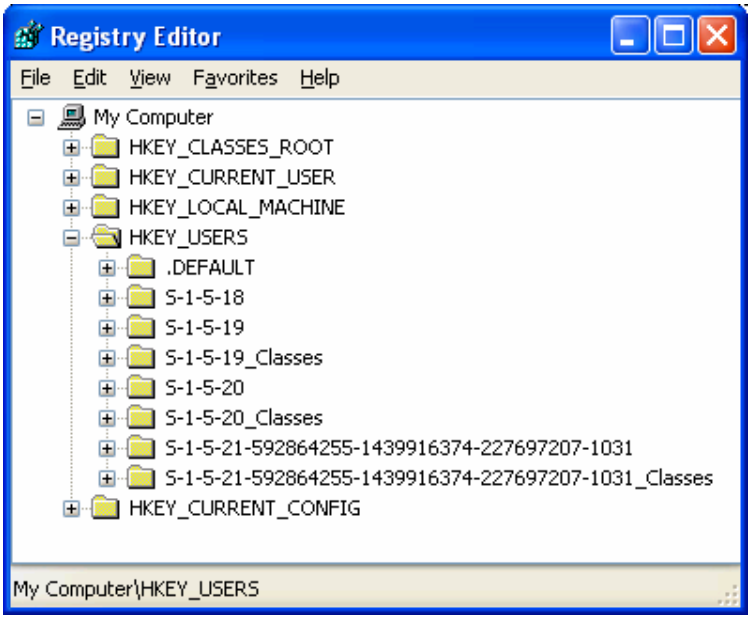

Figure 2.3: Registry editor view of HKEY_USERS.

When a machine is imaged, so are the unique SIDs for the computer and its local accounts. For most environments, this problem isn't as serious as you might have been lead to believe. When a workstation joins a domain, a domain SID is generated for the computer, which results in a unique SID for that workstation. Where this can be a problem is with any computers that do not join a domain (workgroup configurations). When you have two or more computers with the same computer SID, local user and group accounts created on these different computers will generate the same SIDs as they are created. The first account created on one computer and the first account created on another computer (with the same computer SID) will each be assigned the same user SID. In this scenario, both users will have the same security credentials and will therefore be able to access each other's data, even when it is restricted with NTFS file permissions. There are a number of both third-party and Microsoft tools available to address the issue of duplicate SIDs, as the following sections show.

Sysprep

Microsoft Sysprep once did little more than generate a new SID for a system; the latest version of the tool does quite a bit more. The answer file used by Sysprep (Sysprep.inf) provides much of the functionality used to perform an unattended installation, including naming the system, identifying drivers, and setting the local administrator password. For Win2K and later systems, Sysprep can also instruct the system to rebuild its Plug-and-Play (PnP) driver database. I provide more details about Sysprep later in this chapter.

Ghost Walker

Ghost Walker is provided with Symantec's Norton Ghost tool suite to "walk" through a system and replace duplicate SIDs with newly generated SIDs to ensure uniqueness. Ghost Walker also changes the SID for all user profiles on the computer to a unique, randomly generated value. Because both the imaging utility (Ghost) and Ghost Walker run in DOS, changing the SID with Ghost Walker does not require an additional restart as most similar tools do.

SIDgen

SIDgen from Altiris provides a means of changing SIDs in the registry and the file system. SIDgen works by backing up the registry, then generating a unique SID for the computer. It searches all registry hives and replaces appropriate entries with the newly generated SID. SIDgen then replaces SIDs found in the file system (NTFS permissions). You run SIDgen from within Windows, and though it is automated, it does require a reboot when complete.

SIDchanger

SIDchanger from PowerQuest is yet another tool that comes with PowerQuest DeployCenter. SIDchanger supports only NT; PowerQuest recommends using Microsoft's Sysprep for Win2K and later systems.

ImageCast SID Creator

ImageCast from Phoenix Technologies offers SID Creator as its tool to automatically assign a unique SID on each NT and Win2K workstation after imaging. However, the company recommends using Sysprep for this task.

NewSID

Sysinternals' NewSID is a popular free tool for dealing with SID duplication problems. The most recent release includes support for Windows XP and Windows Server 2003, a wizard-style interface, and the ability to specify the SID that you want applied in addition to other enhancements. NewSID also provides full source code.

With so many tools available, it can be a bit confusing to determine which is best for you. If you are not joining a domain or have local user accounts in your baseline, vendor-provided tools such as SIDgen and Ghost Walker provide more thorough scans of the computer, including the file system. However, for most, Sysprep is more than sufficient and is a very powerful tool in the deployment of new systems.

Initial Build Size

Keep your image as small as possible so that it takes up less space (particularly when you need it to fit on removable media). Most tools allow you to span media, but it is certainly more manageable if you can keep your image on one CD-ROM or DVD. With a simple baseline image containing little more than the OS, a compressed image might fit on a single CD-ROM. Although a single CD-ROM image is ideal, the compression you will be able to attain will vary and a single CD-ROM might prove a difficult goal.

Another reason to keep your baseline image small is to increase the speed with which you can apply the image. A smaller image will obviously take less time to process than a larger one. This processing time is of particular concern when it comes to network downloads, as bandwidth limitations will be a factor for a longer period of time. The following list explores ways to keep your initial build size small.

- Reduce Windows File Protection cache—The default size of the Windows File Protection (WFP) cache is 400MB. The same files will be protected by WFP regardless of the cache size, so you can reduce the cache size to decrease the size of your initial build. The downside to doing so is that you might be prompted for an installation CD-ROM or network shared installation files in order for WFP to do its job restoring replaced or removed files.

For information about WFP and controlling the size of the cache, see the Microsoft article "Description of the Windows File Protection Feature."

- Turn off hibernation support—The hibernation file, Hiberfil.sys, is roughly the size of the amount of RAM on the local system, which can be hundreds of megabytes that you need not include in an image. To be rid of the hibernation file, turn off hibernation mode support. To do so, in the Display Control Panel applet, click Power in the Monitor Power area of the Screen Saver tab. Next, select Never from the drop-down list for the *System hibernates* option, and press OK. If you really want hibernation support, consider automating the setting after the system is imaged. Listing 2.1 shows an KiXtart script that activates Windows hibernation support.

```
Run "%Comspec% /c RunDLL32.EXE shell32.dll,Control_RunDLL ups.cpl,2"
Do Sleep 1 Until SetFocus ("Power Options Properties") = 0
$rc = SendKeys("~H")
$rc = SendKeys ("{TAB 4}")
$rc = SendKeys("{RIGHT 4}")
$rc = SendKeys("~H")
$rc = SendKeys("{ENTER}")
```

Listing 2.1: KiXtart script to activate Windows hibernation support.

- Empty the Recycle Bin—By default, the Windows Recycle Bin is set to 10 percent of your hard disk. If you are creating an image, the Recycle Bin certainly should be nowhere near full, but ensure that it is empty to save as much space as possible.
- Clear the event logs—By default, the event logs can be as large as 512KB. In addition to eliminating wasted space, clearing the event logs before you create your baseline image will avoid confusion that arises when events that were inadvertently deployed to systems where the events didn't take place show up in the event logs.
- Delete user profiles—As discussed earlier, deleting user profiles just makes good sense. They can potentially be quite large, but regardless of the size, it is unprofessional to include your own documents, settings, Internet cache, and who know what else in a baseline image to be installed across the network.
- Empty temporary directories—Check the local admin temp directory and the system temp directory for files that might have been generated and not cleaned up during application installations or other setup operations. Use the NT and later built-in utility for cleaning up your hard drive by typing

  ```
  cleanmgr.exe
  ```

  from the Start menu's Run dialog box or at a command prompt. The Disk Cleanup tool will scan the specified drive for temporary files and facilitate their deletion (see Figure 2.4).

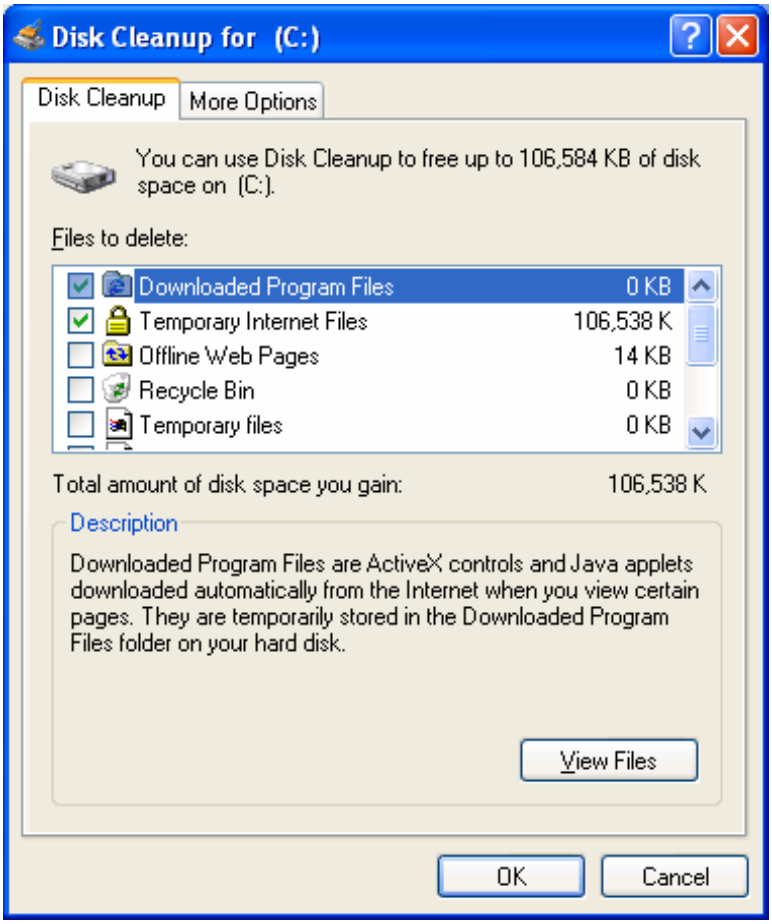

Figure 2.4: The Windows Disk Cleanup utility.

Staging a Deployment

It is a common and effective process to establish a staging area for the installation and initial configuration of new systems. Often an area is set aside to power up computers, apply an image via a local network switch or CD-ROM, and pile them up for distribution.

Benefits

As with many OS deployment considerations, the benefits of putting together a staging area for a deployment will depend on your organization. The following sections explore a few of the common benefits to such a process.

Less Time Spent at the Users' Desks

Having a staging area for your deployment will mean less time spent at users' desks. When you ask users to stand aside while you give them their new (or at least different) computer system, the less time you take the better. The faster you are, the faster you can be on to the next system and the faster the user can be back in business. In the end, the users are your customers, and speed is something they will be impressed with and will result in greater productivity.

Controlled Network Environment

When a network installation (imaged or unattended) is being performed, network speed will be a critical factor in the amount of time each system will take. In the setup of a staging area, you are in a better position to optimize network topology in your favor.

Most imaging software supports multicast network distribution in which one data stream is received by multiple clients simultaneously. Figure 2.5 illustrates a typical data transfer in which individual streams go to all systems.

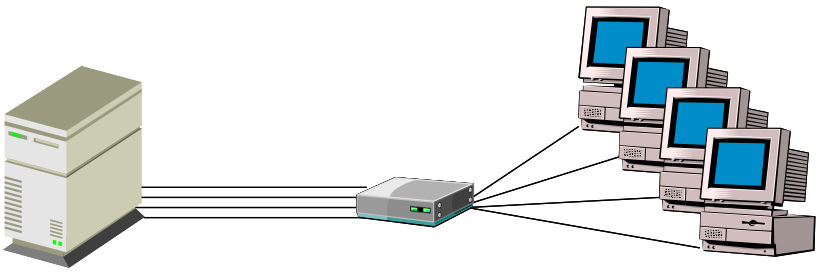

Figure 2.5: Typical data transfer (individual streams to all systems).

As Figure 2.6 shows, multicast is much easier on the network and the server as compared with managing multiple data streams. Although not all network environments support this technology, in a lab environment obtaining and configuring support for multicast should be a much easier implementation. Lack of support is normally in the form of outdated or improperly configured routers, which might require extensive purchases or configuration changes that would make this technology a challenge to implement network-wide. Imaging 10 computers from the same source will prove a considerable advantage over the congestion that comes with 10 individual data streams (all fighting against one another for bandwidth).

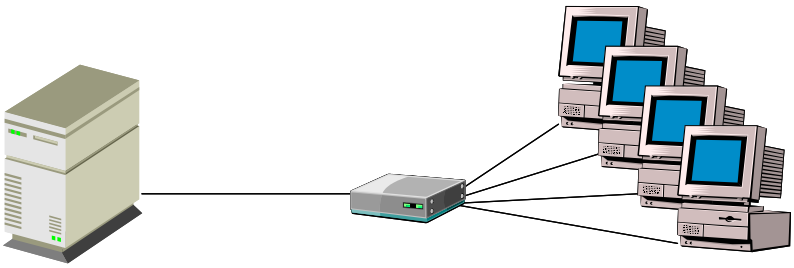

Figure 2.6: Multicast data transfer (single stream to all systems).

Early Identification of Failed Systems

If you purchase 10 computers will they all work? One hundred? One thousand? It is safe to assume that through manufacturing, parts, and shipping (or while you're taking it out of its box) things are going to be broken. It is much more desirable for everyone involved if these systems are identified and addressed in a staging area before you carry them through the building and plugging them in at someone's desk.

Drawbacks

In some environments, a staged deployment simply might not be an option. Your own organization's reasons might be unique; the following sections discuss a couple of reasons that staging might not be a good idea.

Location-Specific Actions

Custom scripts or client/server applications in use on your network might perform automatic actions based on location. In many cases, you can "fool" the system into thinking it is at its final location for staging if such actions are based on the name you give it, computer group membership, or organizational unit (OU) membership. However, when the network subnet is a deciding factor, there may be nothing you can do.

Slow Delivery in a Changing Environment

In a deployment staging area, workstations might sit unused as active computers receive network-wide changes. If you control the rate at which new machines are prepared for deployment, this kind of problem might be minimized. Depending upon your deployment mechanisms, the systems might simply update themselves when connected to the network. In this case, the issue may be less critical. However, the time of the deployment might still be a factor while you wait for the new updates to be applied.

Deployment Methods

There are several deployment methods available; which one is best for you will depend greatly on your budget, size of the deployment, and the skill sets of the deployment team. Much of what we have covered applies to the more common mass deployment methods of imaging and drive duplication. In the following sections, I will describe the common methods used and summarize the benefits and drawbacks of each.

Manual Installation

For those with only a handful of systems to set up, a manual installation is still a common way to go. However, more and more organizations have implemented more robust means of deploying systems that are faster and more consistent even when considering very few workstations. Every method has its up side, and manual installations are no exception:

- GUID uniqueness—Accidental inclusion of GUIDs or other machine-specific items need not be a concern when performing a manual installation. After all, this was how the vendor intended the deployment to be performed.
- Customize to individual needs—Also a negative aspect of manual installations, the ability to customize the installation on the fly is easiest when a manual installation is being performed. If the user has a peripheral that you did not plan on or if there is a software package that must be installed prior to those in your baseline process, a manual process will allow for handling these situations with the least difficulty.

- Hardware installations supported as designed—PnP operations, loading of the appropriate hardware abstraction layer (HAL), and legacy driver detection and installation are all designed to take place dynamically during the installation process. Therefore, both manual and unattended installations enjoy the increased success that comes with performing an installation as it was intended to take place. This support can be a significant obstacle for imaging and drive duplication implementations.
- Windows upgrades—When upgrading Windows to a newer version, manual and unattended installations are the only methods that will allow you to take advantage of an in-place upgrade. Testing is naturally very important here, but the primary benefit of an in-place upgrade is that most applications, data, and settings can be preserved. Of course, any existing problems with the system might also be preserved, but if this is not a concern, an upgrade may be the way to go.

There are a few drawbacks to performing installations manually, some of which are quite obvious:

- Inconsistency—A key reason to avoid manual installation of systems, and applications for that matter, is that the process becomes susceptible to human error and inconsistencies. All the benefits of a tested, planned, consistent baseline discussed earlier are at risk when manual installations are being performed.
- Slow installation times—Among the options available, a manual installation is also the slowest way to go. Even when an administrator is very proficient at installing the baseline configuration, nobody can compete with the speed of an unattended script.
- Multiple reboots—You will have to manually perform multiple reboots, particularly when considering the installation of other baseline software outside the installation of Windows itself. Internet Explorer is well known by administrators to be a real challenge for unattended deployment primarily due to its requirement for a reboot along with a second logon by an administrative account. Windows service packs and hotfixes require reboots as well.

> Microsoft has instituted a directive in its logo certification program that requires vendors to avoid the need to require a reboot of the computer. Far too many situations have surfaced in the past in which reboots are requested but not really needed. With Win2K (and even more so in Windows XP), there are far fewer events that require a system restart. Those events that still require a restart might present alternative ways of getting the job done, which eliminates the need to reboot. What all this means to us as administrators is that with any luck, we will see far less prompts for a restart today and in the coming years.

There are a number of ways to handle a manual installation aside from relying upon the administrator's familiarity and experience to capturing screen shots of every step. When it comes to a manual installation, strong documentation is about the only "solution" available to you.

Documentation and check lists are critical components of consistent and accurate installations. Although some organizations include detail on every file version, location, and source, others will simply list the applications to be installed and the order of their installation. Too much detail might be a necessity for some and an unnecessary waste of time for others, but lack of documentation spells trouble for all. Your own configuration management process will dictate the level of detail necessary. Keep in mind that staff changes and reassignments affect an undocumented build process dramatically.

Unattended Installations

Unattended installations are an improvement over manual installations; the benefits and drawbacks of this method will be covered in the following section. As with the other methods available, the benefits of an unattended installation might outweigh those of other methods depending upon your organization. The following list provides common benefits to unattended installations:

- Consistency—A scripted installation is a consistent installation. Removing the human element from the equation is certainly a strong benefit of an unattended installation. Even if the same individual is setting up every machine on a network using a check list and comprehensive documentation, some degree of deviation (deliberate or accidental) becomes inevitable over time.
- Speed of installation—An unattended installation might not be as fast as applying an image or duplicating a drive, but it is certainly faster than following a manual check list and addressing each element of the installation manually. In an unattended installation, many of the options normally presented are not even displayed on the screen. When compared with responding to every option as it is displayed, it becomes clear that an unattended installation is far faster than a manual installation.
- Manual input—You can design unattended installations to give administrators the flexibility to let users provide input during the installation process. By leaving out certain information such as the computer name and product ID, a mini-setup can be utilized that prompts for only these items, while the remainder of the installation is automated as you specify.
- Cost—Out of the box, Microsoft provides built-in and well-documented methods for implementing an unattended installation. We will discuss these options in addition to some third-party solutions that aim to improve upon Microsoft's offerings; however, for many, a Microsoft out-of-box solution can be developed at little or no additional cost.

The following list provides the drawbacks of an unattended installation. Every method has its benefits and drawbacks; the correct solution lies in what is important to your organization. Here we will cover some of the drawbacks to unattended installations.

- Speed of Installation—While this was also listed as a benefit, it is still considered a strike against this method when compared to imaging and drive duplication. Depending upon the software to be installed after the operating system, an unattended installation could easily take over an hour to complete. The source files and scripts may be stored on a CD (or set of CDs.) The problem may then become the space limitation of just one CD or hanging around to insert the next CD. Alternatively, such installations are often handled over the network, where limitations may be imposed by network bandwidth. It will also be necessary to limit the number of simultaneous installations take place before network saturation brings the file transfer process to a crawl.
- Potential Complexity—Depending upon what it is you want to accomplish during your unattended installation (application installations, driver installation, reboots, automatic logons, etc.) an unattended installation has the potential to grow into quite a complex process. An out of the box installation where Windows is simply installed and waits for a logon can be surprisingly easy to accomplish. However, the more you attempt to automate, the more complex the entire process will become. The primary problem with a very complex approach is there is more to go wrong and, at the same time, fewer people will understand and be able to troubleshoot the process.
- Multiple reboots—As with manual installations, multiple reboots can be a problem for unattended installations.

Microsoft Solutions

Slow to adopt support for imaging, Microsoft's preferred method of automated installation has always been unattended installations. To support this perspective, Microsoft has made a handful of tools available that work together to help you out. We will briefly cover each of these utilities and processes in the following section.

Sysprep

The Sysprep.inf file is used by the Microsoft System Preparation tool (Sysprep) to answer the questions normally posed in the graphical wizards and prompts presented during the manual installation of Windows. Sysprep also allows for the clearing of the PnP database, inclusion of additional drivers, and much more. Take advantage of this powerful tool when utilizing any of the automated methods covered in this chapter. Sysprep offers a host of other configuration options as the Sysprep.inf file in Listing 2.2 shows.

```
[Unattended]
UnattendMode = FullUnattended
NoWaitAfterTextMode = 1
NoWaitAfterGUIMode = 1
OemSkipEula = yes
InstallFilesPath = "%systemdrive%\Sysprep\i386"
OemPnPDriversPath = "\Sysprep\drivers"
ExtendOemPartition = 0

[GuiUnattended]
AdminPassword = *
TimeZone=10
OemSkipWelcome = 1
OemSkipRegional = 1

[LicenseFilePrintData]
AutoMode = "PerServer"
AutoUsers = "5"

[UserData]
FullName="FullName"
OrgName="PC"
Computername=*

[Networking]
InstallDefaultComponents = Yes
```

Listing 2.2: A sample Sysprep.inf file.

In Listing 2.2, the asterisk (*) after the computername value results in a random computername based on the orgname value (for example, PC3149361729). Additionally, the number 1 represents True for many of the parameters shown in the listing.

> Designed for Win2K, a version of Sysprep was made available for NT only to Microsoft Enterprise partners. This version of Sysprep provided limited functionality; namely, SID generation. Many administrators still think of Sysprep in this limited capacity, but it is a very powerful tool for automated deployments of Win2K and Windows XP. You can find Sysprep documentation online at http://www.microsoft.com/windows2000/techinfo/planning/incremental/Sysprep11.asp and http://www.microsoft.com/windows2000/techinfo/reskit/en-us/deploy/dggb_aut_cwrh.asp.

Sysprep can be used to apply an image to dissimilar hardware configurations. Among the items that need not be the same on target computers are video cards, network adapters, audio cards, and mass storage device drivers. However, the system HAL and Win2K kernel (Ntoskrnl.exe) must be identical.

To use Sysprep, extract the Sysprep.exe and setupcl.exe files from the support.cab file located in the support/deployment folder of the Win2K and Windows XP installation CD-ROMs. You can download Sysprep from http://www.microsoft.com/windows2000/downloads/tools/Sysprep. Copy the files to the computer to be imaged and run Sysprep; doing so will configure the computer to launch a mini-setup at the next startup. What is presented by this mini-setup is determined by the contents of the Sysprep.inf file used by Sysprep to prepare the system for imaging (for an example, see Listing 2.2). Once the machine shuts down, create your image or duplicate drives for delivery to other computers.

Unattend.txt

The Unattend.txt file provides answers to the questions typically posed during a manual installation. For Win2K and later, many think only of the new support for Remote Installation Services (RIS), but these OSs still provide full (and even enhanced) support for use of an Unattend.txt file.

> You can find Unattend.txt file format and available parameters at http://support.microsoft.com/?kbid=155197.

You need not create the Unattend.txt file from scratch. Along with the Sysprep and RIPrep information files (discussed later), the Setup Manager Wizard is provided to populate this file, as Figure 2.7 shows. The Setup Manager is included with Sysprep in the deploy.cab file on the Win2K and Windows XP installation CD-ROMs in the Support folder.

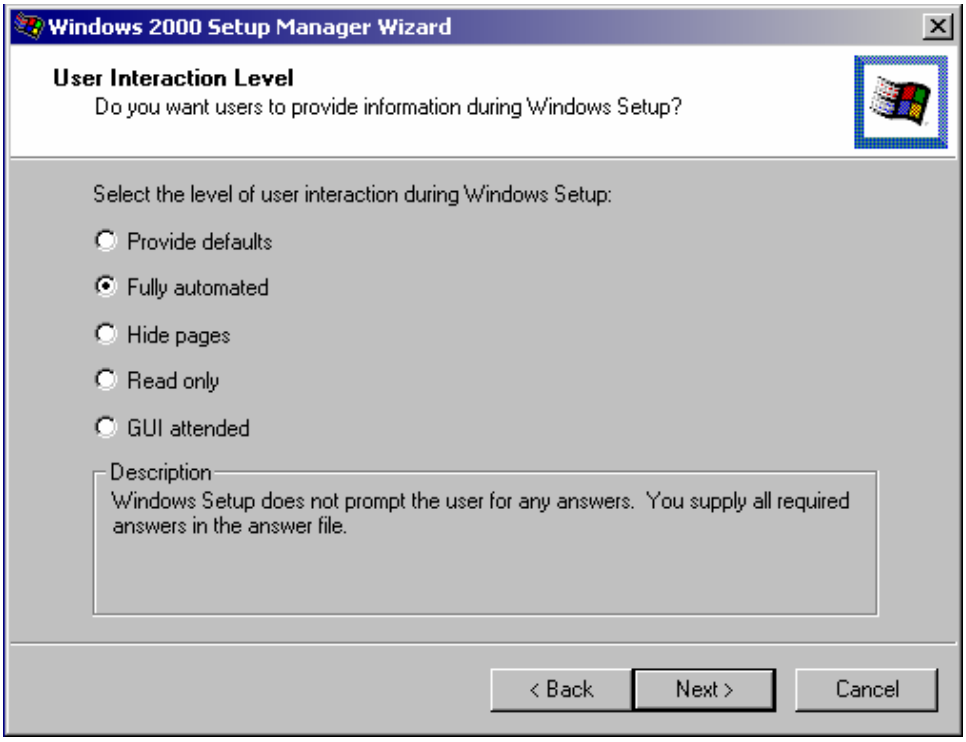

Figure 2.7: The Setup Manager wizard.

> Windows Product Activation (WPA) might look like a process that must be completed manually. However, Microsoft has implemented WPA only to deter the casual rollout of installations without the licenses to allow it. Microsoft fully supports the automation of this process and provides documentation and recommendations for doing so at http://www.microsoft.com/windowsxp/pro/techinfo/deployment/activation.

OEM

By setting the OEMPreinstall value to Yes in Unattend.txt, you can include additional files in the automated installation process. You can employ the OEM directory structure to add to the source files to include additional drivers and even application installation files. The structure for including these additional files is shown here:

```
<drive>
      /i386
            /$OEM$
```

From within the Setup Manager tool, use the Additional Dirs page to supply your own set of files to be deployed. Doing so will generate the previously shown folder structure for the inclusion of any specified files. For example the files you want placed in the System Drive tree are placed in OEM\$1. Each of the special keys that are displayed on the Additional Dirs page, maps to a special subdirectory of the OEM file structure.

> Windows is hard-coded to look for the OEM folder below the /i386 directory. Because a CD-ROM–based installation copies files to a local temporary directory, the path to OEM might become incorrect. For more information about this potential problem, see the Microsoft article "Cannot Obtain Access to oem Folder During Unattended Setup."

Cmdlines.txt

You can use the Cmdlines.txt file to specify command lines for execution as part of the setup process. Stored in the root of the OEM directory, this file is processed at the end of the mini-setup wizard before saving any settings. To make use of this capability, the OEMPreinstall option must be enabled in the specified unattended answer file.

> For more information about Cmdlines.txt, see the Microsoft article 238955 "HOW TO: Use Cmdlines.txt File During Sysprep.exe Setup Wizard."

Third-Party Solutions

It might seem that Microsoft has deployment tools pretty well covered, but there is always going to be room for improvement. By integrating the native unattended support into a desktop management product or by introducing support for an alternative means of accomplishing the task, you have many alternative solutions available.

- SystemPrep—Sys-Manage's SystemPrep is designed for the automated deployment of Windows as well as additional software and drivers. This tool facilitates the creation of shares and network boot disks and the customization for the installation of Windows releases as early as Windows 95. (Figure 2.8 shows the SystemPrep Configuration Wizard interface.)

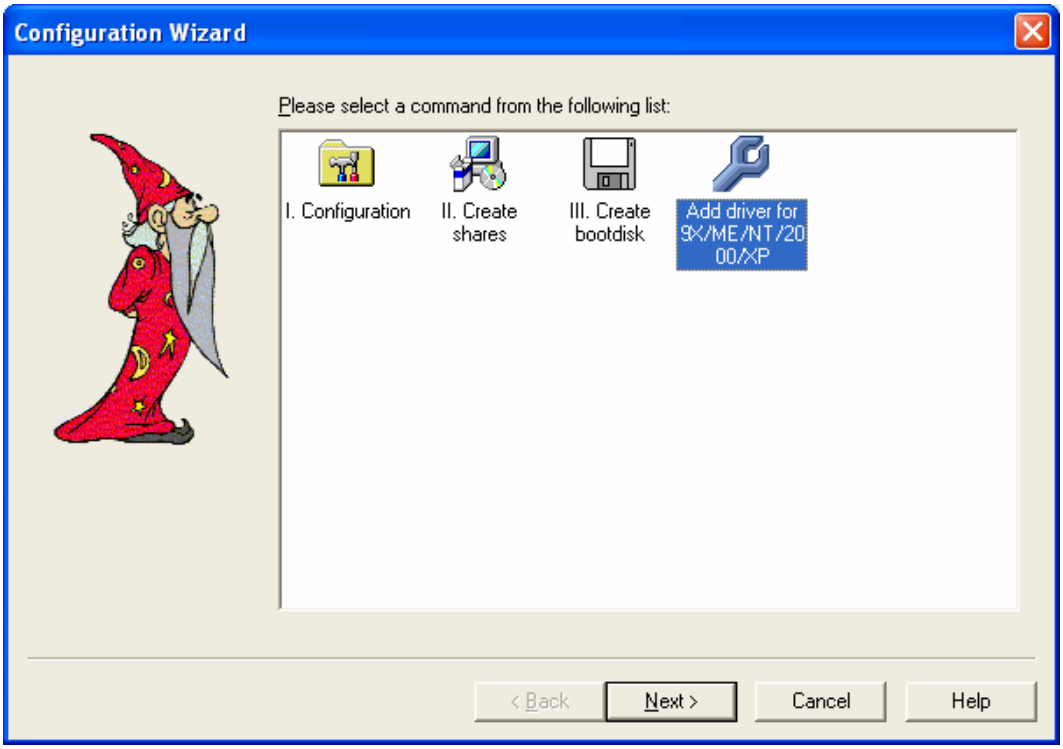

Figure 2.8: The SystemPrep 1.1 Configuration Wizard.

- CapaInstaller PC Creator—CapaSystems' CapaInstaller provides a product to simplify the process of customizing an unattended Win2K or Windows XP installation by asking all the necessary questions and generating a network boot disk to trigger the process. CapaInstaller also features a PC configuration tool, a driver library, a hotfix installation service, a remote reinstall service, and hardware and software inventory.

Drive Imaging

Drive imaging is one of the most popular methods of OS deployment for several reasons, but primarily for its speed and simplicity. But both those benefits assume a successful image has been generated, which can sometimes prove difficult depending on your environment. We will discuss some of the benefits and drawbacks to imaging in the following section.

The benefits of drive imaging include the following items, which have already been discussed:

- Speed of installation—An image can be applied from CD-ROM or a high-speed network connection in just minutes. Considering this image might also include baseline software applications and configurations, speed is a major benefit when compared to other methods available. Factors include the speed of the target system, the method used to provide the image to the system (CD-ROM, network, local partition) and the size of the image itself.
- Consistency—Consistency is a key benefit enjoyed through the imaging process. At least until the next reboot, all computers will be binarily identical at the time the image is applied. When a well-tested baseline is in use, you have a very solid foundation for your new system.

Although a very common means of deploying a large number of systems, imaging has its drawbacks:

- Upgrade is not an option—Imaging constitutes a complete replacement of drive data; therefore, utilizing imaging to perform an in-place update is not possible. One can naturally replace the OS with a newer version, but doing so does not constitute an upgrade in the traditional sense, in which applications and settings are retained or migrated for use on the updated OS.
- All user data and settings are lost—To retain user data and settings, you must arrange to back up this information prior to the installation, then restore it after the installation. Many of the imaging products have addressed this shortcoming of imaging by providing tools to automate this process, even going so far as to incorporate it into a single automated deployment event—backup user data and settings, apply new image, apply user data and settings. Even with such support, added configuration and planning must be applied to the deployment process.

> To address this major drawback of imaging (and duplication as well), many of the imaging products available either include or have partnered with other companies with products designed to facilitate the migration of user data and settings between systems. In many cases, these tools are also able to move between versions of Windows and some cases even support the migration of application-specific information and data to newer versions of the application.
>
> In addition, migration of user data between systems becomes increasingly less important (if not completely unnecessary) if folder redirection is used properly. You can perform folder redirection using ScriptLogic, Group Policies, and/or manual registry edits.

- HAL incompatibilities—With the release of Win2K, the number of HAL incompatibilities has decreased significantly to just one key potential incompatibility: Advanced Configuration and Power Interface (ACPI) and non-ACPI. When an incompatibility is present, the lowest common denominator prevails. Thus, a non-ACPI image should (theoretically) work on any ACPI system (but you would lose its ACPI support in the process). In practice, though, you may find it is not so easy. Trial and error dictates the number of images you require to support all systems in your organization. Despite dissimilar system HALs, you might be able to create an image on one machine that applies to several other machines, but an image created of a newer machine is less likely to work on older ones. Here especially, the fewer hardware differences in your environment, the easier your life will be.

> NT SCSI and IDE drive images present yet another incompatibility that has thankfully been removed from the picture with the release of Sysprep 1.1.

- Multi-image management—Dealing with multiple images is a necessary evil for many organizations. The desire for multiple baselines and HAL incompatibilities are the core reasons that the number of images to be managed may quickly increase. You might find a need to update images on a regular basis as new applications and updates are added to the deployed baseline. If your desktop management solution is designed to dynamically update the older baseline image after it is applied to a new system, this drawback might be less of a concern. However, it is very likely that there will come a time when the existing network baseline and your original deployment image differ to such a degree that updating the image becomes necessary. When image updates become necessary, the desire for a small number of images becomes evident.

Available Solutions

There are a fairly small number of vendors providing imaging software on the market today. Many of these solutions are briefly discussed in the following section.

> ☞ An independent review of the imaging products available was performed in 2000, which included a comparative matrix of features in several categories. You can find this competitive matrix at http://www.appdeploy.com/comparisons/imaging.

- Remote Installation Services (RIS)—RIS looks more like an unattended installation when you watch it in action, but it is actually a file-based image. Like all of the other imaging solutions available, it can access the network via a network boot disk or PXE (a hardware-based network boot). RIS requires Active Directory (AD) and it supports only Win2K and Windows XP. In addition, it does not support multicasting, which might eliminate it as an option for many enterprise networks planning for large-scale deployments. RIPrep, which Figure 2.9 shows, is a utility provided for the preparation and uploading of images for deployment via RIS.

> 🕮 You can find more information about RIS from the following resources:
>
> RIS installation and configuration check list
> http://www.microsoft.com/windows2000/en/server/help/sag_RIS_Install_Checklist.htm
>
> Technical Guide to Remote Installation Services
> http://www.microsoft.com/technet/prodtechnol/windows2000serv/deploy/depopt/remosad.asp

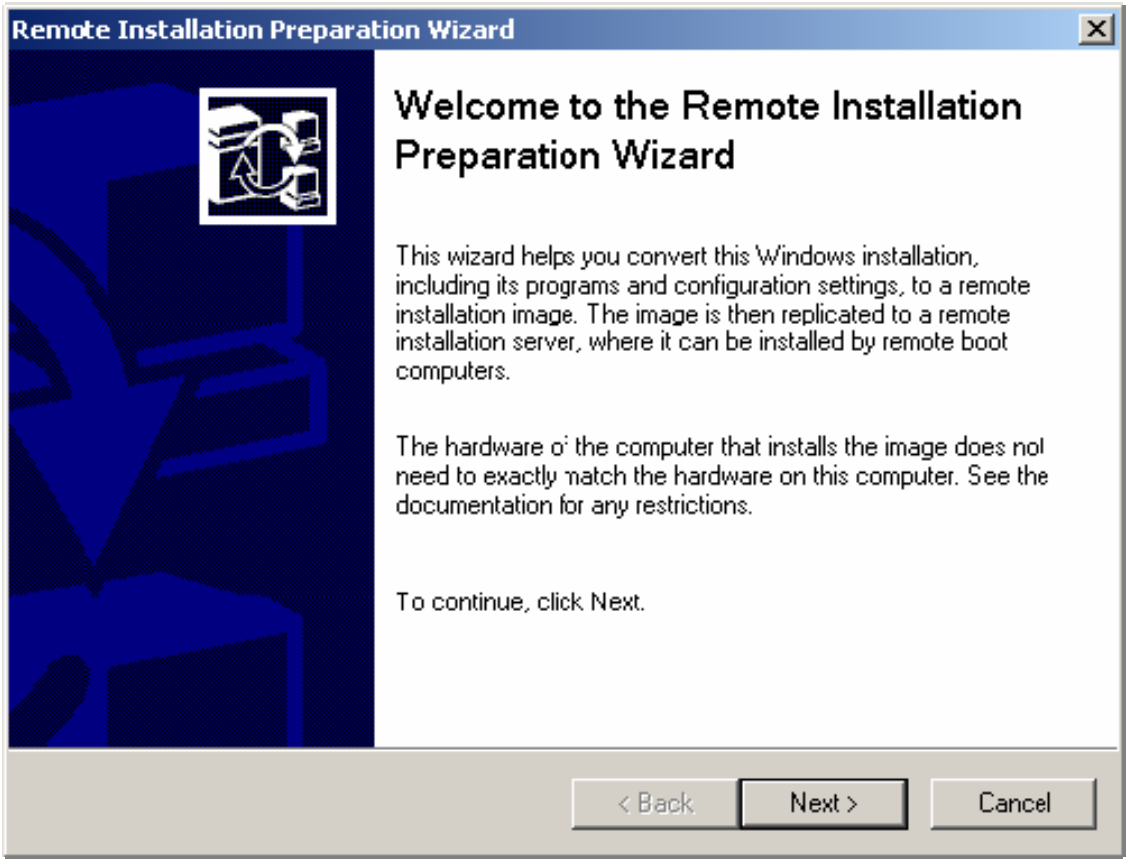

Figure 2.9: The Remote Installation Preparation Wizard (RIPrep.exe).

> ☞ You can customize the text menus presented by the RIS installation process. You can edit the menus or create them with a simple text editor such as Notepad using the OSCML file format. You can find information regarding the modification of these menus as well as its file formatting at http://www.microsoft.com/technet/prodtechnol/windows2000serv/reskit/distsys/part5/dsgappg.asp.

- Ghost—In the imaging market, Symantec Ghost is certainly the most well-known product. In fact, the term *ghosting* has become synonymous with imaging. Ghost provides the ability to image directly to CD-ROM or other removable media. You can use a recordable CD-ROM drive as a place to write an image file to at the time the imaging of a drive is being performed. In addition, Ghost provides USB and LTP peer-to-peer imaging support, offers remote deployment of client management service, and is very fast when compared with other imaging solutions.
- DeployCenter—Formerly Drive Image Pro, PowerQuest's DeployCenter is another very popular imaging tool with plenty of strong features, including PXE and multicast support, image direct to removable media (including CD-R), and a Web-based management console.

- RapiDeploy—Altiris has changed the name of its imaging product from ImageBlaster to RapiDeploy, and the tool is now available as a component of the company's desktop management suite, Altiris Deployment Solution (formerly eXpress.) Although the name changes can be confusing, the product is very intuitive and offers several great features, including complete remote deployment capabilities via its Web-based management console, the ability to create an image of a hard drive and simultaneously distribute that image to other machines at the same time the image file is being generated, remote deployment of client management service support, and direct editing of compressed image files functionality.
- ImageCast—The last release of Phoenix's ImageCast was 4.6.1; however, it is still available in enterprise and manufacturer (MFG) editions. ImageCast's Post Configuration Injector is used to customize each workstation with its individual settings (such as the user name, computer name, and IP address). This tool offers a 100 percent scriptable interface as well as the ability to generate restore CD-ROMs with RestoreBuilder.
- TrueImage—Acronis is a newcomer to the market; its TrueImage product is geared more toward the individual user than an enterprise deployment. However, it is worth mentioning that TrueImage provides the ability to create an image of a machine from within Windows (including the system partition), as Figure 2.10 shows.

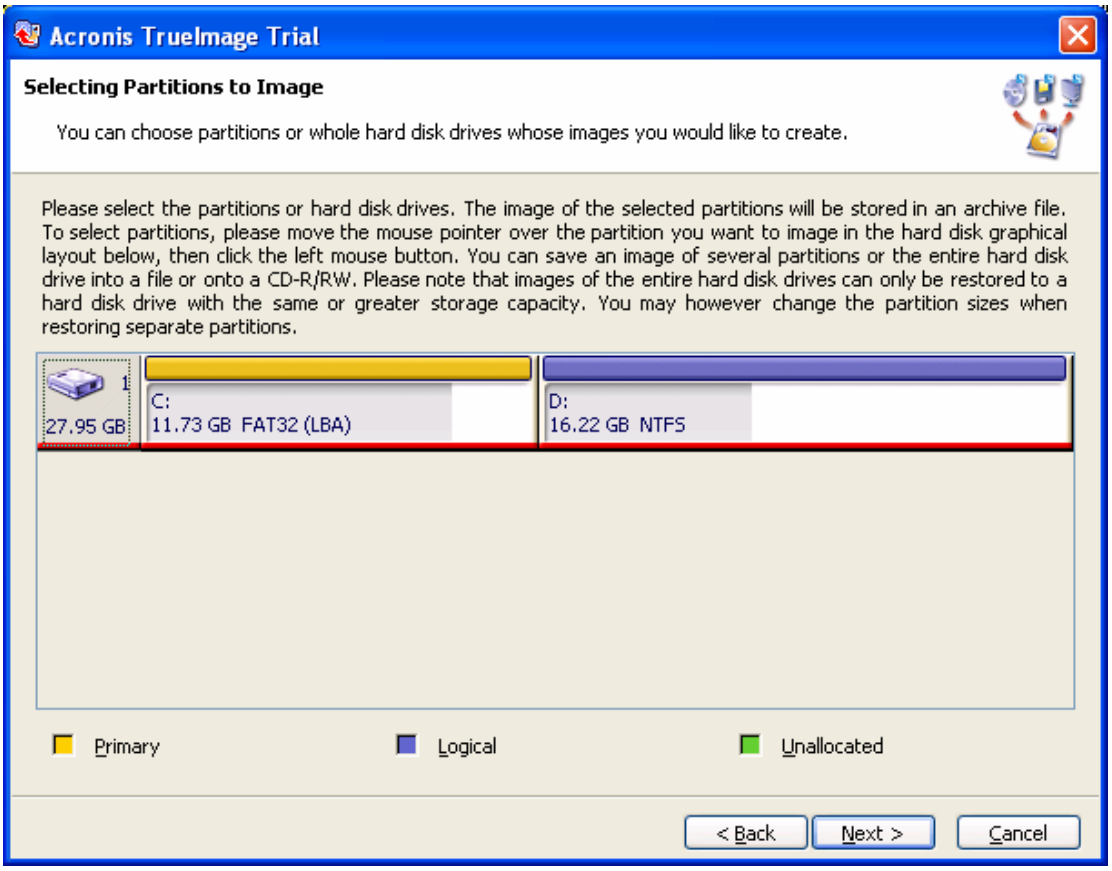

Figure 2.10: The TrueImage partition selection dialog box.

- Auto-Deploy—The main purpose of Rembo Auto-Deploy is to deploy an OS on client computers by replicating a reference system. The deployment process is driven from a central database containing unique parameters for each PC (including the rules that decide which images and software are applied to each computer). The deployment process can be fully automated (without interaction on the client computer), but RAD can also let the final user choose the OS and the applications to deploy. Report logs are then sent to the central console to help the IT administrator control the deployment.

Which Is Right For You?

In many cases, the tool you will use is the one that you are most familiar with. For the most part, imaging software products provide the same feature set. I've listed many of the key differences and standout features, but if you are looking to implement a new solution, do your homework and be sure to get the features that are important to you. Take a look at the features available, create a requirements document stressing what is important to you, and the right choice for your organization will become much more clear.

> We will discuss application deployment tools in Chapter 4.

Drive Duplication

Drive duplication is the process of copying hard drives from one to another. This method is a hardware-based solution. The benefits of drive duplication as opposed to the other methods available will depend greatly upon your deployment process. If you want to replace the drive of existing machines with larger ones, drive duplication will likely become a more desirable solution when compared with imaging. The following list provides common benefits of drive duplication:

- Ideal for pre-staging scenarios—There might already be a step in your deployment process that involves opening the case for the installation or removal of components.

> Some environments remove floppy drives or modems for security reasons. Others might need to install zip drives or security devices. Even when buying thousands of machines, it is often not cost effective to order the machines customized by the manufacturer—even ordering that a part be removed from a system will likely result in additional costs because it does not fall in line with the manufacturer's process.

- Speed of deployment—Many duplication systems allow you copy from one drive to more than one destination drive at the same time. At faster than 2.3GB per minute, this solution is the fastest available. Add to this the fact that you can simultaneously image as many as 16 drives, and duplication becomes a clear winner where speed is an issue.
- Network bandwidth requirements—Unlike software imaging, and even unattended installations, drive duplication happens off the network, thus not consuming network bandwidth.

- OS independent—With the drives being physically duplicated, the contents of the drives are irrelevant. Therefore, there is no limit to the OSs you can duplicate—any version of Windows, Linux, UNIX, Mac, or any other OS.

All the negative aspects of drive imaging apply to drive duplication. In addition to these drawbacks, drive duplication requires you to open each machine—some systems lend themselves to drive replacement better than others.

Available Solutions

There are several companies providing duplication equipment. We will briefly discuss some options as well as factors you might want to consider when making your decision.

Features to look for in a drive duplication solution include support for IDE and SCSI, the ability to duplicate a large number of drives at one time, and the ability to duplicate dissimilar hard drives. Easy loading and unloading of drives and, of course, speed are also key factors. Some duplicators include diagnostics and utilities, and if you are cloning very large drives, you might want to verify support before purchase (particularly those over 137GB). Some of the larger manufacturers include ImageMASSter, Logicube, Greystone Data Systems, Corporate Systems Center, and Wytron Technology.

OS Deployment Best Practices

OS deployment is a function performed by many desktop, systems, and network administrators. Thus, a number of common sense guidelines are available, many of which were learned the hard way—through experience.

Workstation Naming Conventions

Definitely give some thought to the make up of your workstation names. For identification visually and for use by scripts, a well-engineered naming convention should be a major consideration when starting up a new network or replacing a significant number of systems. This decision is one that you will have to live with for some time. Depending on your environment, different identifying data will make sense. Table 2.2 provides some factors to consider.

Factor	Consideration
Multiple sites	Include a site code or name. If you are using a systems management system such as SMS, you might already have designated site codes in use. See if a site code or identifier is in use on the network before generating new ones.
Multiple buildings	A site might include more than one building. In this case, including something to identify the building number might be a good idea.
Multiple floors	If your network is dispersed between floors or might be in the future, including the floor can be of great help. Does the room number identify the floor already? If so, the floor number might be an unnecessary element.
Multiple wings	Which wing a system is in is also something often identified in a room number, but if it is not and it will help people locate a machine by its name, you can include the wing location as a valuable bit of information.
Multiple rooms	If physically locating a machine is of any interest to you, the room number is likely the most valuable information to include in a workstation name.
More than one computer per room	When there are two or more computers in the same room, and the room number is being used as identifying data for a computer name, you will need to come up with a consistent way of identifying computers within the same room. If there are two, perhaps the one closest to the door is A and the one further away is B. If there are several, you could identify them from distance to the main entrance or perhaps clockwise as you stand in the doorway.
Primary function	If a computer is a shared resource for Internet access or to connect to a certain network, the system's primary function can be useful identifying data to include in the computer name.
Primary user	If computers are assigned to individuals and not rooms, include a UserID or initials. When using initials, keep in mind that the likelihood of duplicate instances will increase along with the number of users. Also keep in mind, people move and company turnover rates might fluctuate. Try to avoid a situation in which you constantly have to change computer names.
Asset tag number	If you have an asset management system in place, using just the asset tag code might suffice. This decision would weigh greatly on the availability and accuracy of your asset management database.
OS	If you have multiple OSs on the network and this information is vital, including an identifier for the OS might make sense. Remember that the OS can change and there are other (possibly better) means of determining and/or documenting the OS for the machine.

Table 2.2: Naming convention factors and the reason to consider them.

A structured computer name can not only help you identify machines visually, but it can also help scripts and other management systems identify machines. For example, if you want to perform an action on all computers at a specific site, and the site is identified as part of your computer name, you can use the computer name as a condition within a script. Several commercial products allow for computer names to be utilized in identifying systems by accepting a computer name as a parameter and offering wildcard support. For example, ScriptLogic's Validation Logic offers the ability to identify which computers should perform a specified action on computer names. Microsoft Operations Manager (MOM) bases the remote deployment of its management agent on computer name. And, when using Deployment Solution from Altiris, you can specify events (such as software deployment) on the computer name. These are just a few examples to help illustrate the value of a meaningful naming convention; check your management software for such support. Even if your software only supports sorting systems by name, a structured naming convention can be of value.

Some example computer name structures include:

- S01B2F1G210A—Site 1, Building 2, Floor 1, G Wing, Room 210, the first or only computer in the room
- ENG-KIOSK03—Engineering group common area workstation number 3
- WNT-KKELLY—An NT workstation that belongs to Kayla Kelly
- PC1032-327G—Asset tag number PC1032 and room number 327G.

Don't include everything—too long and complicated a name might defeat the purpose. You might want to be able to know where the machine is, what its function is, or whom it belongs to. However, computer names are limited to 14 characters. Whatever the decision, try to think beyond your current network configuration. The likelihood that your network will grow in the future is very strong.

> Win2K DNS supports the underscore (_) character as a valid DNS character, but other DNS solutions might not, so to avoid possible problems, Win2K Setup recommends underscores be changed to dashes. As a result, it is strongly recommended that you abstain from using underscores in computer names—you might never encounter a problem as a result of using one, but if and when you do, it might not be clear that the underscore is the problem. It is best to simply not use them from the beginning.

System Build Information

Include baseline identification or similar relevant information to the system as part of your rollout process. Best implemented by script, when you include information in the registry for later use, you will be thankful step later.

Image Identification

Perhaps you are lucky enough to have an image that works on all hardware platforms that you need to support. This image is still likely to be the first of many, and a naming convention will prove helpful to distinguish images from one another. If you have different baseline images to track, image identification fits well into an image naming convention. The simplest method is a versioning system. The first image released is version 1, the second is version 2, and so on. If fixes or minor changes have to be made to an image, include a sub-version number (such as 1.2, 1.3, 2.0, and so on). It is also a good practice to include a date on any CD-ROMs that are created, as there may be times during image development at which new and updated images are routinely generated.

Summary

We started this chapter by discussing the concept of establishing a common baseline. We also covered the major deployment methods for rolling out new systems as well as the benefits and drawbacks of each. In the next chapter, we will discuss user profile and user data management.

Chapter 3: User Profile and Data Management

In the last chapter, we covered the deployment of new systems. With your baseline identified, tested, and deployed, the real challenge of desktop administration comes into play—satisfying your users. Your goal in this area is clear: provide your users with the software they need as well as helpful default settings to ease its use. If a user has to search for an application, configure the application, search for his or her data, then work to identify and install the nearest printer, they will not be productive (or happy) users. Even when users are fairly experienced in the use of Windows, searching for documents and printers can be a real headache. A well-implemented desktop can save users from these speed bumps, reduce Help desk calls, and increase user productivity. In this chapter, we will cover user profiles and various ways you can implement and control them (for an overview, see the sidebar "Group Policy, Scripting, and Third-Party Solutions"). We will also discuss what might be considered the most critical aspect of desktop administration—user data management.

Group Policy, Scripting, and Third-Party Solutions

Many of the issues covered with regard to desktop administration lend themselves to one of three primary solutions: Group Policy, scripting, and third-party solutions. Group Policy provides a specific set of solutions to certain problems; where appropriate, I've pointed out these capabilities throughout this chapter. Group Policy is popular with its users; however, as a result of Group Policy's strict requirements, a great many organizations can't take advantage of the benefits of this desktop administration solution. Group Policy requires AD, and its managed clients must be running Win2K or later—thus, Windows 9x, Windows ME, and other clients are out of luck in terms of using Group Policy.

Even if you use Group Policy on your network, this solution can't do it all. In environments in which you need to address specific requirements (for example, your network has legacy desktops), scripting is often the tool that will get the job done. In Chapter 7, we will explore how you can script solutions to several desktop administration issues.

Finally, if scripting isn't within your skill set or your needs are robust enough to demand a supported, documented, and evolving solution, third-party software is the choice for you. Although scripting your own solution is a viable option in many situations, developing and maintaining a very large and complex script can be expensive. In such cases, a third-party solution makes sense.

The problems you face will often direct you to the most reasonable solution for your scenario. My point is to show you that you have desktop administration solution options. Depending on the size and complexity of your network (and the complexity of the problems that you are sure to face as a result), it often makes sense to employ a combination of Group Policy, scripting, and third-party solutions to address all of your needs.

User Profiles

Many settings affect the entire system (and all who use it). For this reason, NT and later versions of Windows provide default security settings that restrict users with non-administrative permissions from modifying such items as system time, display resolution, and software installation, to name a few. (For information about Windows 9x support for user profiles, see the sidebar "Windows 9x Support for User Profiles.") Conversely, there are many settings that are user-specific, such as printer and drive mappings, software settings, desktop colors, shortcuts, and wallpaper. These types of items are stored in an area separate from those that users don't have permissions to change, and the user is given full control over these settings, which are referred to as the *user profile*. The user profile is made up of a directory structure identified with the root folder name of the user to which the profile belongs. This root folder also contains a file named User.dat or Ntuser.dat, which contains the HKEY_CURRENT_USER registry hive used to store settings specific to the user (see Figure 3.1). There are many ways to configure, modify, and even force settings within a user profile.

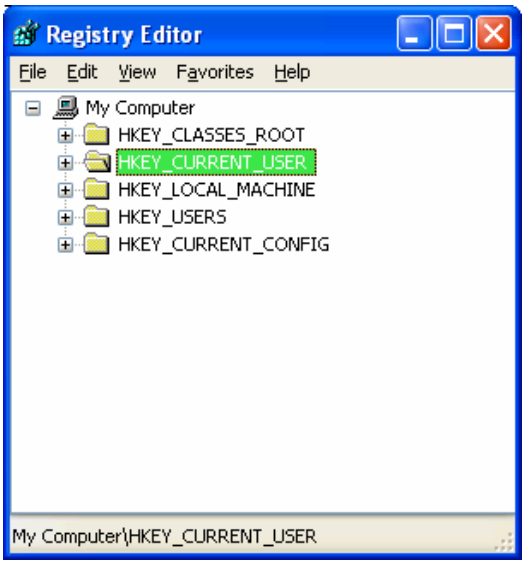

Figure 3.1: The user profile registry settings are loaded as HKEY_CURRENT_USER.

Profiles provide the ability for multiple users to work on the same computer. When each user logs on, the user has the same desktop settings as were available the last time the user logged off. Another key benefit is that customizations made to the desktop environment affect only that of the current user and not the environment of other users that might share the same computer. By implementing roaming profiles, such settings and data can be stored on a server from which it can be backed up and made available to users anywhere on the network.

As a means of profile recovery, Windows 95 uses a file named User.da0 and NT 4.0 uses a file named Ntuser.dat.log. Although these files are similar, they provide slightly different functionality. Windows 95 writes a copy of User.dat to User.da0 each time the user logs off as a simple backup copy. NT uses the Ntuser.dat.log file as a transaction log file, which allows for fault tolerance in the event that a user profile must be recovered.

See "Using Profiles with Multiple Versions of Windows" at the end of this chapter for information about the use of roaming profiles when logging onto multiple versions of Windows.

Windows provides four types of user profiles: local, roaming, mandatory, and temporary. In the following sections, we will discuss the purpose of each as well as considerations for local and roaming profiles to help you identify which implementation makes the most sense in your environment.

Local Profiles

The first time a user logs on to an NT or later system, a local profile is created and stored locally based on the default user profile (discussed later in this section). Changes made by the user are saved and subsequent logons will make use of this same user profile (if it exists). Table 3.1 provides the default locations for user profiles.

Windows Version	Default Location
Windows 95	\Windows\Profiles*username*
Windows 98	\Windows\Profiles*username*
Windows NT	\WinNT\Profiles*username*
Windows ME	\Windows\Profiles*username*
Windows 2000 * (English Language)**	\Documents and Settings*username*
Windows XP * (English Language)**	\Documents and Settings*username*

* If the installation is an upgrade from NT, user profile folders are stored in the same location as in NT.

** The localized base profile path can be determined by reading the registry

Table 3.1: User profile locations by Windows version.

Where a roaming profile is not found, local profiles are created and stored on each computer a user logs onto. The settings remain for use next time the same user logs on. To have the same settings available to users logging onto multiple systems, you must implement roaming profiles.

Roaming Profiles

Roaming profiles are copied down from the server (to the location as indicated in Table 3.1). When a user who is using a roaming profile logs off, the profile is copied back up to a specified share on the network.

The copying of the local profile to the network is determined by the file and object timestamps. Any files or objects in the local copy of the profile that have a different timestamp from the corresponding file on the network are copied back to the target profile server location. At a minimum, the Ntuser.dat and Ntuser.ini files are copied up to the server because the act of logging on and off the system will cause modification (and therefore a timestamp update) of these files.

The following steps walk you through the procedure for managing users on a Win2K server. NT provides very similar functionality as it pertains to configuring roaming profiles through its User Manager for Domains utility.

1. Set up and share a folder on a server.
2. Open the Active Directory Users and Computers snap-in, and navigate to the container in which the user account exists.
3. Right-click the user's name, and select Properties.
4. Select the Profile tab (see Figure 3.3).
5. For the profile path, input the path to the network share where the user profiles will be stored (for example, \\ServerName\ShareName\UserName).

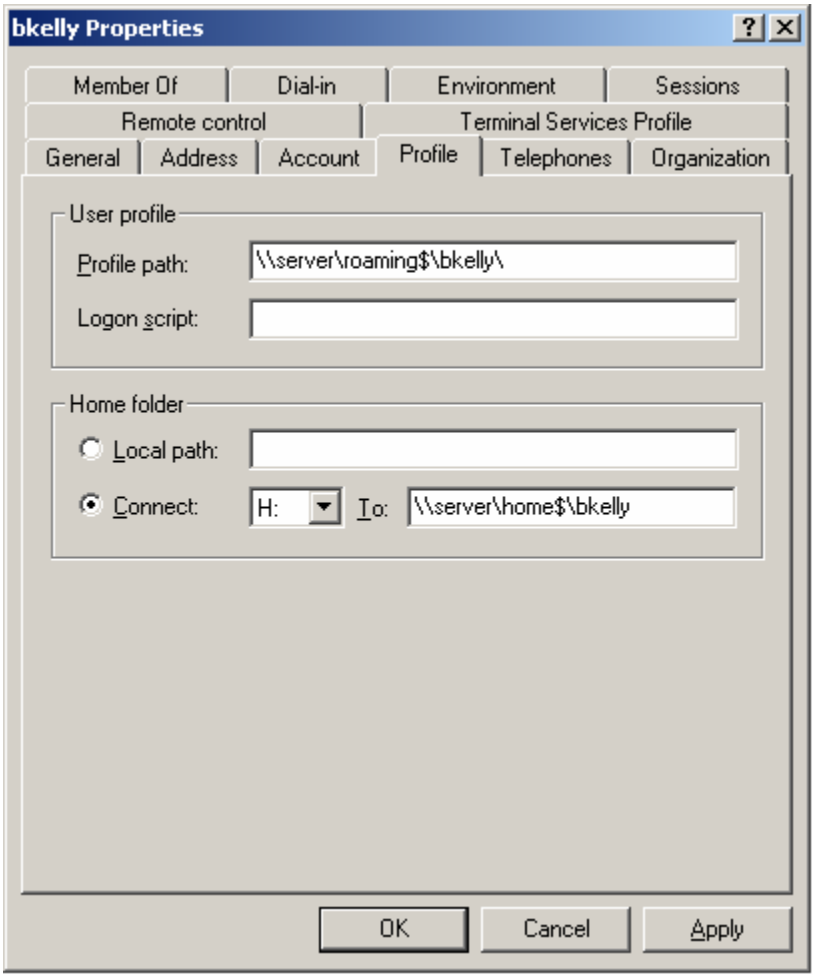

Figure 3.3: The Profile tab in Win2K Active Directory Users and Computers.

To create the roaming profile share, create and share a folder as the root folder for storing user profiles. Append a dollar sign ($) to the share name if you would like it hidden from users browsing the network (\\server/roaming$). Set the NTFS and Share level security to allow all users access to the share (the default security will be to allow full control to the Everyone group). In the user profile, enter the share path using the local path to the share and entering the username as the folder (for example, \\server\roaming$\bkelly). The roaming user profile directory will be created when the user first logs onto the domain (as opposed to user home directories, which are created at the time they are specified).

> The profile share can be stored on any system. The process of downloading the profile is controlled by the client computer—all the client needs is the correct path. However, it is strongly recommended that you use a server as opposed to a workstation for this purpose. One big reason is that NT Workstation, Win2K Professional, and Windows XP all have a limitation of 10 inbound network connections. However you may use a network attached storage (NAS) device or UNIX server share.

> ☞ You should not specify the same location for the user's profile and home directory. If you do so, all files that the user stores in his or her home directory will be copied up and down from the server in the profile copy process. If you want to use the same share, specify the roaming profile (as opposed to the root of the share) as a subdirectory in the user's home share to avoid this situation.

When logged onto more than one system at a time, it is the system you last log off of that will dictate the contents of your roaming profile. To demonstrate this idea, the following steps walk you through an example scenario. In addition, Figure 3.4 helps to illustrate the situation.

1. On COMPUTER1, you install software while logged onto both COMPUTER1 and COMPUTER2. This software installs required registry keys in your profile (such as licensing information).
2. You log off of COMPUTER1, and your profile information is copied to the server as expected.
3. You now log off of COMPUTER2, the registry entries and other profile changes that might have been made during the software installation on COMPUTER1 are lost when COMPUTER2 copies its version of the user profile over the profile copied up by COMPUTER1.
4. You log onto COMPUTER1, and the copy of your profile that does not know about the software installation is copied down to the local machine.

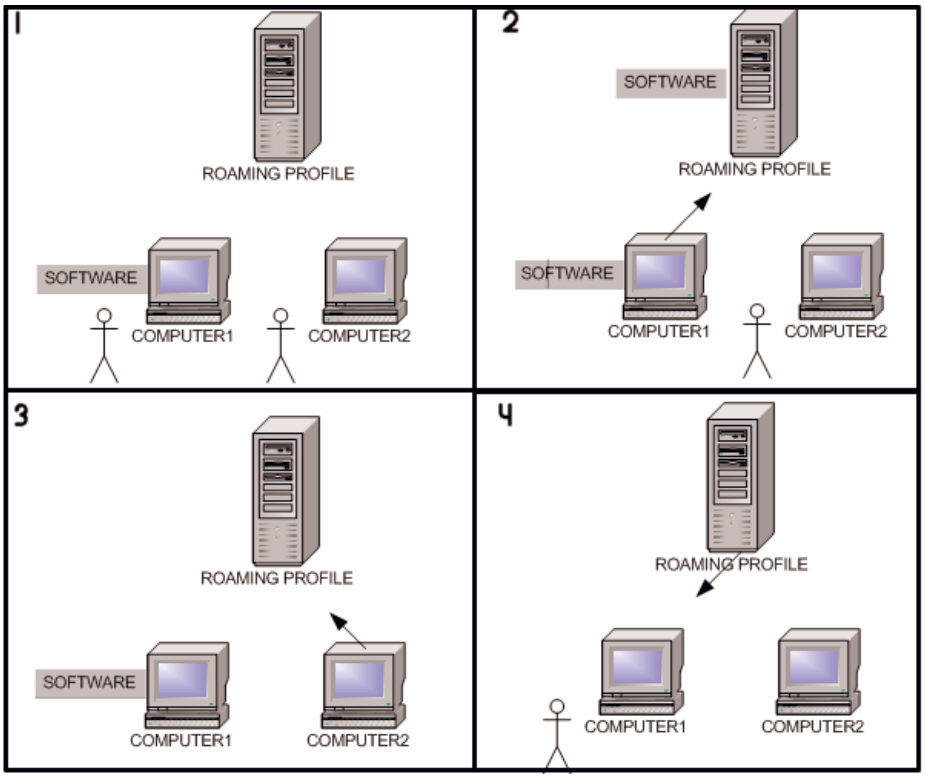

Figure 3.4: Multiple logons and roaming profiles can be a troublesome combination.

Additionally, roaming profiles do not work well over slow network links. In fact, Microsoft does not recommend using roaming profiles across a slow network link at all. Aside from extending logon and logoff periods, the risk of unsynchronized profiles increases.

One example of how slow network links can be trouble for roaming profiles occurs when a user logs on via a network connection that is slow enough to cause Windows to time out during the logon process. Windows instead uses the local profile (if the user doesn't have a local profile, one is created using the default user profile). If the remote server becomes available when the session ends, Windows will copy the local profile up over the roaming profile on the server.

> ☞ NT and earlier versions of Windows detect slow links by measuring the time it takes the server to respond to a request for the file attributes of the profile. This action is timed and compared with the value determining a slow network.
>
> Win2K and Windows XP determine whether there is a slow link by measuring the response time from a sequence of TCP/IP pings from the client computer to the server to determine the average transfer rate in kilobits per second. If the response time from any of the pings is less than 10 milliseconds, the link is not considered to be a slow one. Otherwise, the average transfer rate is calculated from three ping requests to the server with 2048 bytes of data. If this calculated average transfer rate is slower than the default (2000 milliseconds) or a value defined by the administrator, the connection is considered slow. The formula used in Win2K is
>
> link speed=16000/(average ping for 2048 byte packet)
>
> Roaming user profiles are not required to be stored on a server on which TCP/IP is an installed protocol. This method of pinging the server is attempted, and if the server is identified as not supporting the TCP/IP network protocol, it uses the method used by NT 4.0 and earlier versions.

By default, roaming profiles are stored on the system and remain until used again or manually deleted. You can, however, set policies to have roaming profiles removed from workstations if you desire using System Policies for NT and earlier (see Figure 3.5) or Group Policy for Win2K and later.

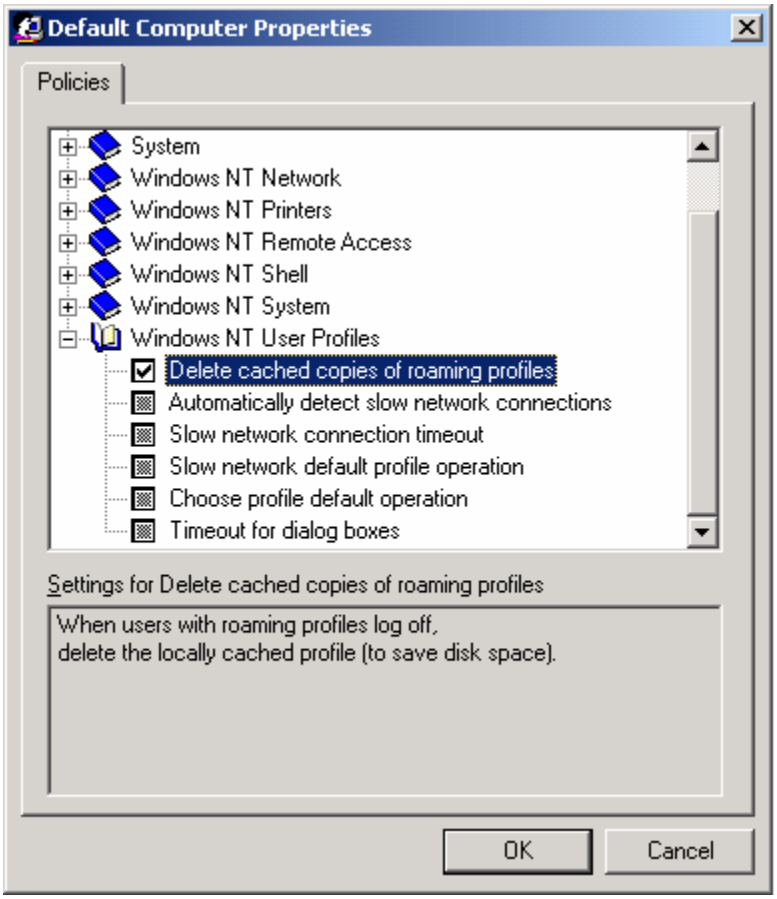

Figure 3.5: The System Policy value for deleting roaming profiles.

When the copy of the user profile on the local system is newer than the one on the network, the user is presented with a choice of which to use, as Figure 3.6 shows.

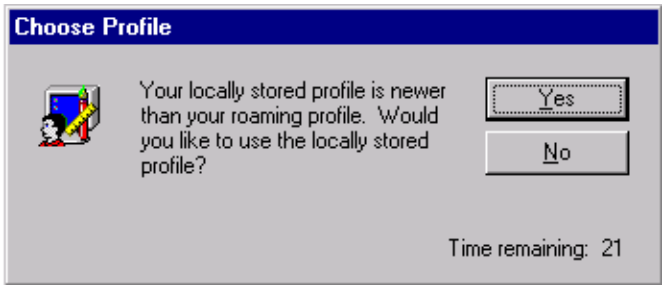

Figure 3.6: Choice presented by Windows when the network profile is newer than the local profile.

By default, this dialog box is presented for 30 seconds and the default selection is made automatically. This choice can be confusing to users, and it is common for them to select the default option (which differs among the Windows versions) or panic long enough that the default option is selected for them. For NT, the default is to download the network profile; for Win2K and later, the default is to use the local copy. To specify your desired default, add or modify the following registry value:

Key	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon
Value	SlowLinkProfileDefault
Type	REG_DWORD
Data	0 or 1 (1 = Download Profile, 2 = Use Local Profile)

You can edit the registry or use System Policy to exclude desired directories from being included in a user's roaming profile. This capability was introduced with NT Service Pack 4 (SP4). To use the registry edit method, add or modify the following registry value:

Key	HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon
Value	`ExcludeProfilDirs`
Type	REG_SZ
Data	Directories to exclude, using relative paths from the user profile root separated by semicolons
Sample Data	`Temporary Internet Files;Application Data\Microsoft\Outlook`

To use System Policy to exclude directories from a user's roaming profile, follow these steps:

1. In System Policy Editor, load the templates Common.adm and Winnt.adm (usually found in the hidden c:\winnt\inf folder).
2. Create a new policy.
3. Open Default User, then expand Windows NT User Profiles.
4. Select the *Exclude directories in roaming profile* option.
5. In the text box provided, enter the directory names you want to exclude. Again, the directories must be relative to the root of the user's profile and semicolons must be used to separate multiple entries (the default value is Temporary Internet Files;Temp).

Mandatory User Profiles

Mandatory user profiles are roaming profiles that can be used to specify particular settings for individuals or an entire group of users. Only systems administrators can make changes to mandatory user profiles. A mandatory user profile will not save changes made to the desktop by users during their logon sessions. Users can modify the desktop settings of the computer while they are logged on, but none of these changes are saved when they log off. The mandatory profile settings are downloaded to the local computer each time a user logs on. You can designate any profile as a mandatory profile simply by renaming Ntuser.dat as Ntuser.man in the root of the desired roaming user profile directory. Win2K provides support for mandatory profiles in this way, but in environments in which AD has been implemented, it is recommended that Group Policy be used to provide a layered, and more granular, control of user profiles.

> Although mandatory profiles are supported for Windows 9x clients, mandatory profiles cannot be shared. You must create a separate profile for each user.

Temporary User Profiles

Available only on computers running Win2K and later, a temporary profile is issued any time an error condition prevents a user's profile from being loaded. Temporary profiles are deleted at the end of each session. Changes made by the user to his or her desktop settings and files are lost when the user logs off. When a temporary profile is to be used, users are presented with a message during logon that advises them of this condition.

Profile Security

Local access to profile directories is restricted on NT and later versions of Windows where NTFS is in use. On NT computers, profile directories are created with the user account, local system account, and administrator's group having full control.

Directories containing roaming user profiles need at least add and read permissions for profiles to be read correctly. Because Windows looks for the existence of the folder first, having only add permissions will result in failure. For a roaming profile to be written back up to the server, the user must have at least change permissions. If a profile is mandatory, the user account must have at least read permissions on the network share on which the user profile is stored.

Contents of a User Profile

Several standard directories are included in a user profile, many of which are hidden. The following list provides the makeup of the settings and default directories included for a Win2K system (see Figure 3.7):

- Application settings—All user-specific application settings installed or generated by the user
- Windows Explorer file settings—All user-definable settings for Windows file explorer as well as persistent network connections
- Program groups and taskbar settings—All personal program groups and their properties, all program items and their properties, and all taskbar settings

- Printer settings—All network printer connections
- Control Panel settings—All user-defined settings made in the Control Panel
- Help bookmarks—Any bookmarks placed in the NT Help system
- Application data directory—Application-specific data, such as a custom dictionary for a word processing program; application vendors decide which data to store in this directory
- Desktop directory—Desktop items, including files and shortcuts
- Cookies—Internet Explorer (IE) cookies
- Local Settings—Application settings that do not roam with the profile; usually these settings are computer specific or are too large to roam effectively; the following subfolders are present by default: Application Data, History, Temp, and Temporary Internet Files
- Favorites directory —Shortcuts to program items and favorite locations
- NetHood directory—Shortcuts to Network Neighborhood items (hidden)
- My Documents directory—User data directory
- PrintHood directory—Shortcuts to printer folder items (hidden)
- Recent directory—Shortcuts to the most recently used items
- SendTo directory—Shortcuts to document storage locations and applications
- Start menu directory—Shortcuts to program items
- Templates—Shortcuts to template items (hidden)

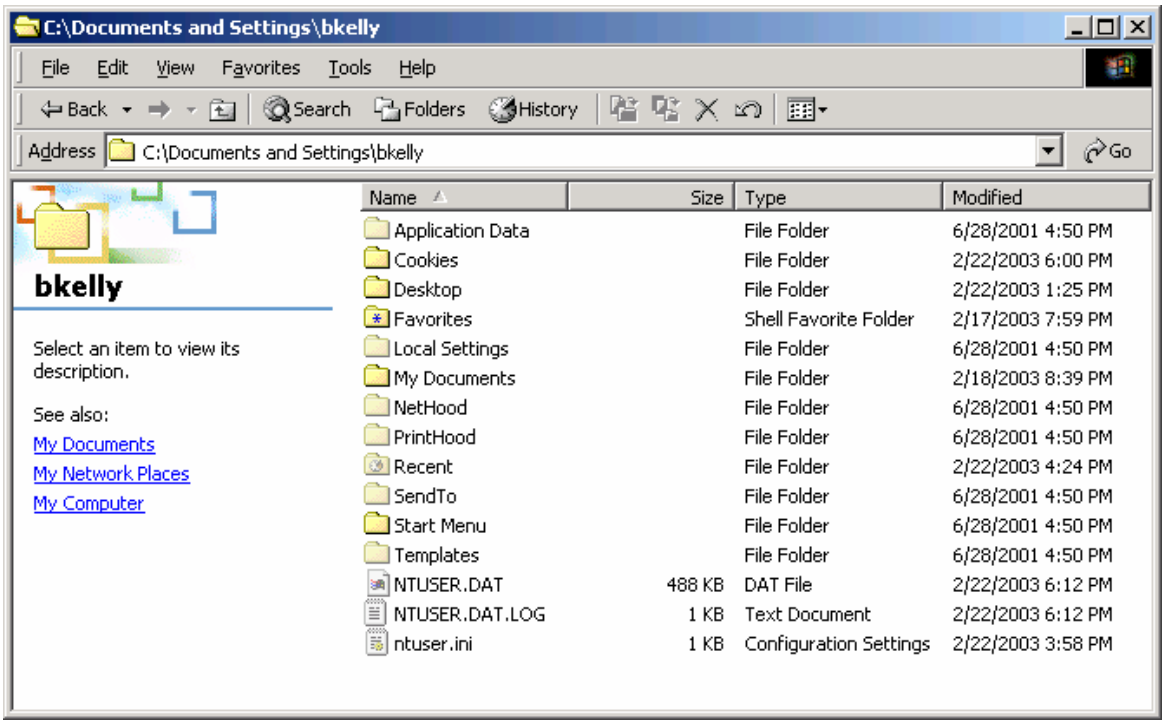

Figure 3.7: Root directory view of a user profile.

User Profile Storage

There are situations in which duplicate profiles names are avoided by the system. For example, when a user named bkelly first logs onto the computer, a standard profile directory structure is created:

```
Documents and Settings/bkelly
```

On NT systems, a different user logging on with the same name would have a unique profile path created by appending a three-digit number to the UserID, starting with 000 and incrementing each additional time another duplicate name with a different user SID logs onto the computer:

```
Documents and Settings/bkelly.000
Documents and Settings/bkelly.001
```

On Win2K and later systems, when a second account with the same name logs on, the account domain name is appended to make it unique:

```
Documents and Settings/bkelly.APPDEPLOY
```

On Win2K and later systems, in situations in which further duplication might occur, a number is appended that increments each time the situation occurs:

```
C:\Documents and Settings\bkelly.APPDEPLOY.000
C:\Documents and Settings\bkelly.APPDEPLOY.001
```

NT 4.0 records which profile should be used by which user by placing registry keys for the user's security ID (SID) in the registry in HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList. Each user who has logged on to the local machine will have a SID recorded here in its own subkey with a value that contains the path to that user's local profile (ProfileImagePath). Should multiple users with the same account name log on to the network, separate distinct profiles are created for each.

Profiles Templates

When manually creating user accounts, you might choose to create and use multiple default user profiles that cater to the software and tasks appropriate for individual users or groups of users. Using this method, you might apply a tailored default user profile to users in order to provide them with a more customized starting point:

6. Create a new user account that will be used as a template for the preconfigured user profile.
7. Log on as the new user, then customize the desktop and install applications to configure this user's profile for the user profile template.
8. Log off, then log on as the Administrator.
9. Open the System applet in Control Panel.
10. On the Advanced tab, under User Profiles, click Settings.
11. Under *Profiles stored on this computer*, select the user that you created in step 1, click Copy To, and enter the path specified for the user's roaming profile.
12. In the Copy To dialog box under *Permitted to use*, click Change, and in the Select User or Group dialog box, select the user to whom you are assigning the profile.

Default Profiles

The default user profile is used as the starting point for new user profiles that are generated when a local or roaming profile is not available for a user logging onto the computer. To create a custom default user profile, follow these steps:

13. Log on to the system as Local Administrator, and create a new local user account.
14. For NT, use User Manager and for Win2K and later systems, right-click My Computer and select Manage, to start the Microsoft Management Console (MMC) snap-in that allows local user account creation.
15. Log off of the computer, and log on again using the newly created local user account.
16. Customize the system as you want default users configured.
17. Log off of the system when customization is complete, then log on again as the Local Administrator.
18. From the Control Panel, launch the System applet.
19. On the Advanced tab under User Profiles, click Settings (for NT, select the User Profiles tab), click the user profile that you just created, then click Copy To. In the Copy To dialog box, specify the location as <*local user profile path*>\Default User to create a custom local user profile or choose your network replication folder to create a network default user profile.
20. Under *Permitted to use*, click Change, and select Everyone. Finally, click OK to complete the process.

> If you want a domain-wide default profile, enter the path to NETLOGON\Default User on the domain controller. Doing so creates the default user profile for the domain. To create a network default user profile, specify the replication directory (the Netlogon share is a read-only share). For Active Directory (AD) environments, choose the share location from SYSVOL. In either case, the profile should be saved as Default User.

> Need to open System from a command line as an administrator? On Win2K and later, don't log off—instead type
>
> runas/user:*computername*\Administrator"rundll32.exeshell32.dll,Control_RunDLLsysdm.cpl"
>
> Keep in mind that you cannot copy or delete a user profile that belongs to the currently logged on user or any user whose profile is in use.

You can store the default user profile in either the Netlogon share of domain controllers or locally on each system. It is common to utilize both, but understand the precedence taken to determine when each is used: when available, the network copy of the default user profile will be used and not the local copy. In fact, unless the network is unavailable or you are dealing with a user logging onto the computer locally (not the domain), the default user profile on the local machine might never be used. The following list describes the order used to determine which profile is employed for a user:

- Is there a central (roaming) profile defined in User Manager for Domains? If yes, any profile created locally will become a roaming profile (otherwise the profile will be created as a local profile).
- If yes, does the roaming profile exist? If yes, use the roaming profile.
- If no, does this user have a local profile? If yes, use local profile.
- If no, does Default User exist on the Netlogon share of the validating domain controller? If yes, use the network default user profile to generate a new user profile.
- If no, does Default User exist in the local profile directory? If yes, use the local default user profile to generate a new user profile.

Manual Profile Modifications

To manually customize a user profile, you can manipulate the file structure or registry hive locally or from a network stored roaming profile. Keep in mind that if you are manipulating a roaming profile, it might be overwritten locally when logging on or the network copy might be overwritten if the user then logs off. To edit a user's registry entries, load the user's registry hive file as a new subkey, as Figure 3.8 shows.

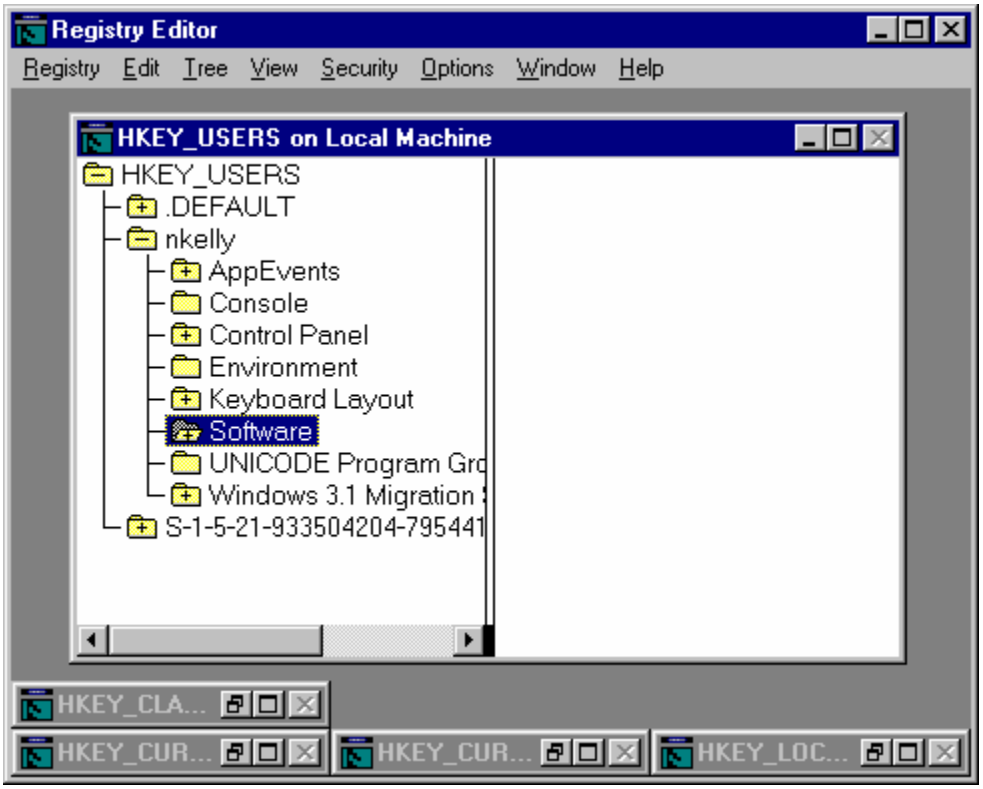

Figure 3.8: Loading another user profile as a new hive for manual editing.

21. Start regedt32.exe, and click on the root key of HKEY_USERS to highlight it.

22. From the Registry menu, select Load Hive.

23. Browse for the desired Ntuser.dat file to be manipulated.

24. A dialog box will prompt you to enter a key name. Enter the UserID to identify the profile being loaded, and click Enter.
25. Click Enter to add the profile registry hive as a subkey to HKEY_USERS, as Figure 3.8 shows (in this case, nkelly).
26. Edit the existing values as desired.
27. After completing the changes, highlight the root of the user's profile registry key, and from the Registry menu, select Unload Hive to save the changes to the user's profile.

Dictating Data Storage

Where do you want users storing their data? The answer to this question should be carefully considered. Once you decide, you need to determine how you'll enforce the storage policy—perhaps through written company policies or by denying users access to save to undesired locations. There are a handful of options that you can implement in several ways, as we will discuss in the following sections.

Network Share

Most environments prefer that user data be stored on a network share so that it might be easily backed up. You can create different shares to store data files by category or to share documents with other users in any specified group. Controlling who maps such drives and who should have access is often specified by group membership.

User Home Directory

You can specify the user home directory as a local directory or network share (see Figure 3.3). When a network share is specified, a drive letter to map the share to is also identified. During the logon process, the home drive is mapped to the designated drive letter.

The user home directory is created at the time it is specified in the user profile and its default security settings allow only the user assigned to the home directory to have access. Some administrators choose to create a separate share for each user; others choose to create a single share and include all user home directories within it. Either way, you can use file security to restrict unauthorized users from access.

> Having both a home directory and roaming profile set to the same path can result in all files in the home directory being deleted when a user logs off. However, it is acceptable to make the profile directory a subdirectory of the home directory (for example, \\Server\Share\UserHomeDirectory\Profile).

Single Share

For the single share method, create and share a folder as the root folder for storing user home directories. Append a dollar sign ($) to the share name if you would like it hidden from users browsing the network (\\server\homes$). Set the NTFS and Share level security to allow all users access to the share (the default security will be to allow full control to the Everyone group). In the user profile (see Figure 3.3 earlier in this chapter), enter the share path using the local path to the share and entering the username as the folder (for example, \\server\roaming$\bkelly). The home directory will be created at the time it is specified here.

On Win2K and later systems, you can use *deep mapping* support, which allows for drives to be mapped to a directory within a share. Particularly when dealing with user profiles, deep mapping is a very helpful feature. By mapping to, for example, \\server\roaming$\bkelly, a user sees his or her data directly within the user's mapped home drive. On NT and earlier clients, mapping directly to a subdirectory (*deep mapping*) is not supported. In this case, you could map to \\server\roaming$, for example, and the user would need to open the directory identified by his or her user name. Though security restrictions will keep users from accessing each other's home directories, in an environment with many users, digging for the proper folder can easily become a hassle. The alternative is to create a share for each user, as the next section details.

Individual User Shares

To use individual shares for each user, create a folder to hold all user home directories, then create subdirectories for each user. This step is typically done using the user name as the name of the folder for the corresponding user (for example D:\Homes\bkelly). Share each user folder and modify the default permissions to remove the Everyone group, then add the user account for whom the share is being created with Change permissions. Specify this location as the user profile directory in the Active Directory Users and Computers MMC snap-in.

> ☞ When the creation of a great number of home directories is on your task list, writing a script to do the job can be a real time saver. For a simple approach to automating this process, see the Microsoft Article "Batch Process to Create and Grant Access to Home Directories."
>
> Don't want to script it? There are a couple of tools to help you get the job done including AutoShare from ScriptLogic (http://scriptlogic.com/eng/Products/Autoshare/main.asp) and UserManagemeNT from Advanced Toolware (http://www.advtoolware.com/t4e/general/um_default.htm).

User Profile

Too much data in a roaming profile can result in extended logon and logoff times as well as increases the likelihood of corruption or impatient users powering off their computers during the copy process. For these reasons, it is best to limit the size of the user profile through implementation of drive quotas (discussed later in this section). Unless you manage the default location for the system and its applications, the user profile might well be where data is stored. Additionally, many novice users do not understand the difference between a shortcut and a folder. As a result, it is not uncommon for a user to copy or move a folder to his or her desktop for the sake of convenience.

Local Data Directory

Probably as the result of some past experience or paranoia, many users typically like to have their data stored on their local systems. Of course, if their machines suffer a drive failure or are re-imaged, their data could be long gone. In a small environment, network backups of local systems or scheduled scripts to synchronize the data with a network copy might be a sufficient solution to this problem. For larger networks, it is often policy not to allow users to save locally, which might be further enforced by restricting write access to the local hard drive. Generally, allowing users to store data locally increases the risk of loss and might restrict your ability to replace or upgrade systems in the future.

Setting Default Paths

One common way to steer users away from storing data in their profiles is to specify desired locations as default for any deployed applications. Although Microsoft products are usually very good about documenting and providing settings for the location of templates and user data, not all applications are so friendly. You will need to examine each application used to generate user data in order to determine how (or if) you can specify the location of user data. As discussed, using the default user profile to configure these settings is often a means of providing a helpful starting point (this should include the setting of default paths). However, as applications are added and updated, managing this via the default user profile becomes less and less helpful over time (affecting only new users). You can further specify paths via registry edits using logon scripts or by using other tools designed to provide similar functionality. Remember, these are default paths—the users may browse to any location he or she has access to when saving data. Finally, you can use Group Policies and System Policies to enforce settings such as default data paths. Keep in mind that when setting paths using System Policy or Group Policy, the settings will be enforced at each logon, and therefore should only be used to *enforce* settings (not to provide defaults).

Folder Redirection

Folder redirection is a term used to describe the process of locating standard profile directories in alternative locations, typically a network share. Folder redirection can be used to significantly reduce the size of a user's roaming profile (avoiding long logon times and even corruption of data. Doing so can also be valuable in that you can trick applications and users by letting them work as they normally would, while the directories they are familiar with are actually located elsewhere. Group Policy provides the ability to implement folder redirection. You can also implement similar functionality by manually editing the registry.

> Ultimately, all folder redirection will end up being the result of the manipulation of particular registry values that inform the system where to find those folders. There are, therefore, several ways to achieve this result and manipulate these registry values (for example, manually, automatically via Group Policy, or automatically via a third-party solution). In the third-party solution category, utilities such as ScriptLogic (http://www.scriptlogic.com) and Enterprise Configuration Manager (ECM) from ConfigureSoft (http://www.configuresoft.com) include folder redirection in their arsenal of skills. In the freeware/shareware category, tools such as Multi-Remote Registry Change from Eytcheson (http://www.eytcheson.com) offer a solution for registry editing en masse (for links to additional registry editing tools, visit LabMice.net at http://www.labmice.net/Utilities/registrytools.htm).

For Win2K and later clients in an AD domain, you can utilize Group Policy to specify folders you want to have redirected. You can specify a single path for all users or common locations based on group membership. When specifying the same path for multiple users, you can use the %username% environment variable to personalize the path for each user, as Figure 3.9 shows.

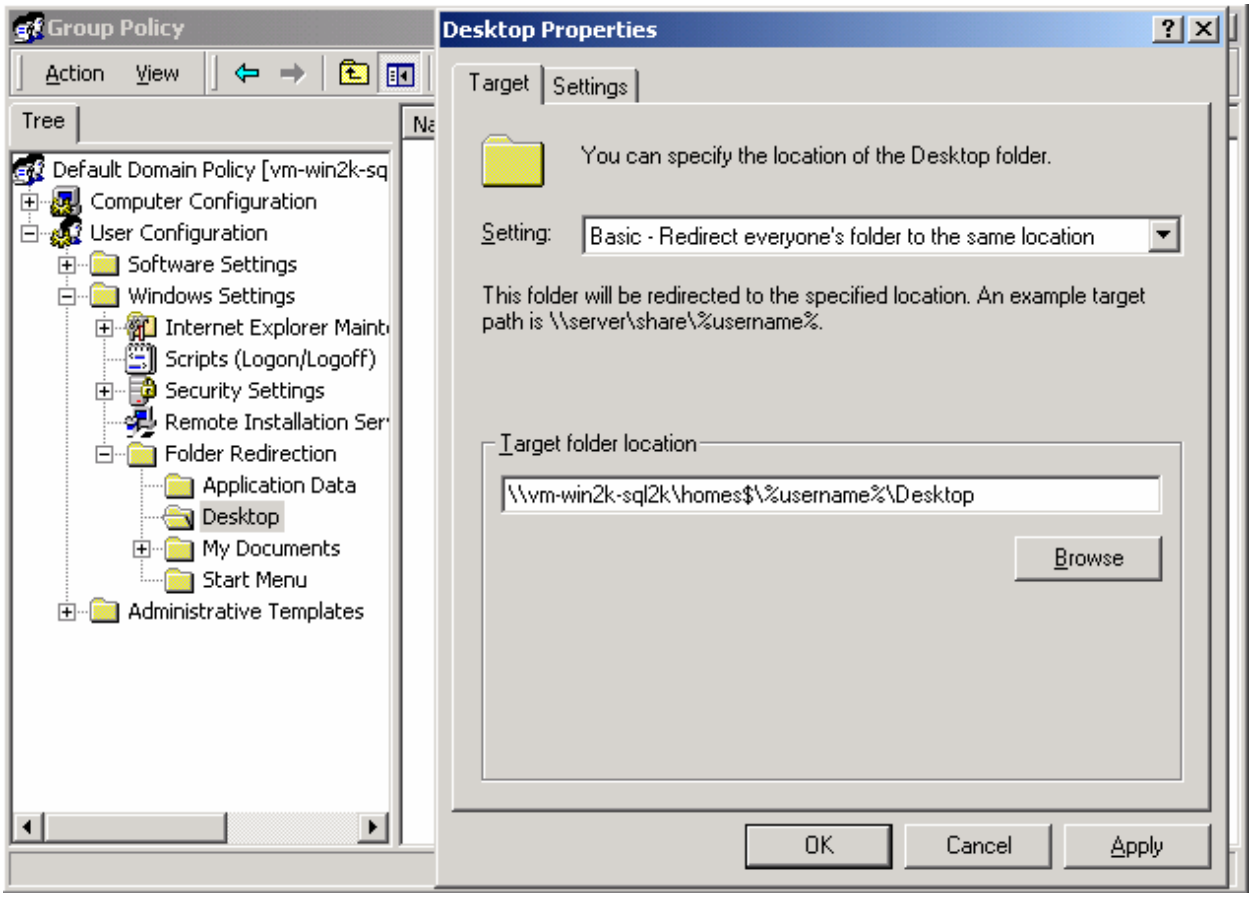

Figure 3.9: Folder redirection properties for the desktop folder in a Group Policy Object (GPO).

If you specify a folder that doesn't exist, the system will create the folder when the user logs onto the network. It is customary to utilize folders in the user home directory for this purpose to increase the use of this managed location and to avoid storing user data in too many different locations.

With the folder redirected, references made to the folder automatically interact with the redirected folder specified in Group Policy. For example, if the desktop folder is redirected (as in Figure 3.9), you will be able to see that any shortcuts created on the desktop are actually created in the redirected folder and not in the local user desktop folder. In fact, the local desktop folder normally found in the user profile will no longer be present.

The Start menu is another location that you might want to configure for folder redirection. For example, you might specify that all Domain Admins have their Start menus redirected to a common location where shortcuts to all administrative utilities are maintained.

Group Policies can only perform folder redirection on Win2K and Windows XP and later clients. For legacy clients such as Windows 9x and Windows ME desktops, you'll have to use a third-party tool or manually edit the registry to implement folder redirection.

Folder Redirection via Shell Folders

Alternatively, you can implement folder redirection by editing the *shell folder* registry entries. Even as far back as Windows 95, the location of several special folders (those normally located in the user profile) are identified in the following registry key: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders. Even in an AD domain, you can use this method to specify redirected folders, as Figure 3.10 illustrates.

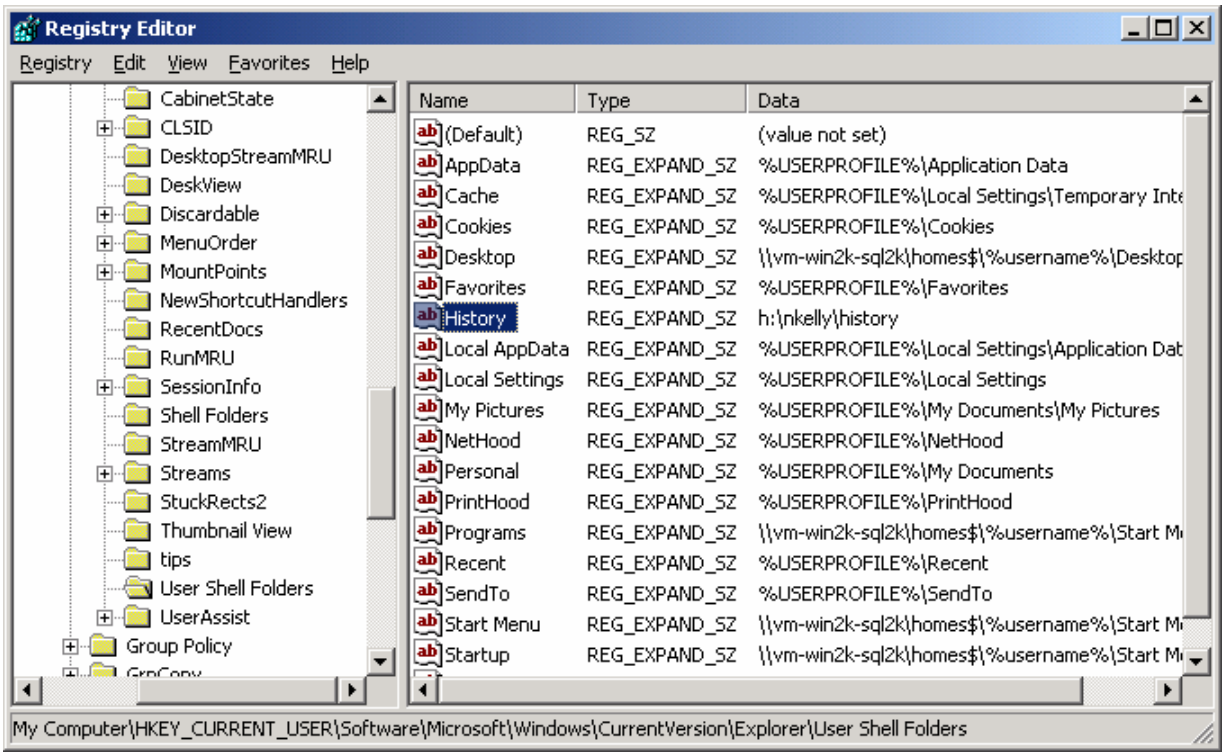

Figure 3.10: Redirected folders registry entries set via GPO or by registry edit.

In addition to the User Shell Folders key, NT and earlier include a Shell Folders key. You can find both the User Shell Folders and the Shell Folders key in NT in either the current user or local machine hives of the registry. The order of precedence is as follows:

- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders
- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders *

- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders *

* These keys are NOT used in Win2K and later.

This method of modifying the registry to achieve folder redirection works similarly to that of Group Policy but it is not quite as robust. Whereas Group Policy allows you to move the contents to the redirected location, manipulating the paths in the registry simply points the user to a second, new copy of the folder. You will need to move the contents yourself or the result might be duplicate entries (see Figure 3.11). In the following figure, the user's Programs folder location was modified in the registry, but the original still exists, resulting in two Programs folders.

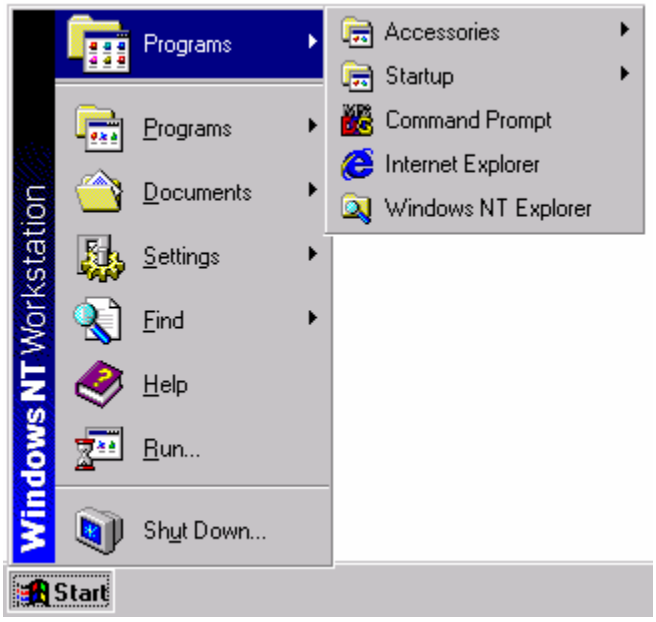

Figure 3.11: A redirected programs folder in NT results in two Programs folders.

If the original location is left, keep in mind that it is in no way linked to the redirected copy. Users might still navigate to the folder locally and any modifications will not be reflected in the redirected location.

> ☞ Like folder redirection via Group Policy, if a path is specified that does not exist, it is created when the user logs on.

User Profile Quotas

With the release of Win2K, built-in support for drive space quotas is now available. NT offers support for quotas via the ProQuota.exe utility (provided in SP4). You can prevent users from having roaming profiles that take up too much disk space or, more important, that are too large to be copied up and down from the network. When users' profiles exceed the limit defined in Group Policy (using the Limit Profile Size policy, as Figure 3.12 shows), they are presented with a message instructing them to reduce the size of their profile. Further, they are prevented from logging off until the profile has been reduced to a size less than that of the defined maximum size.

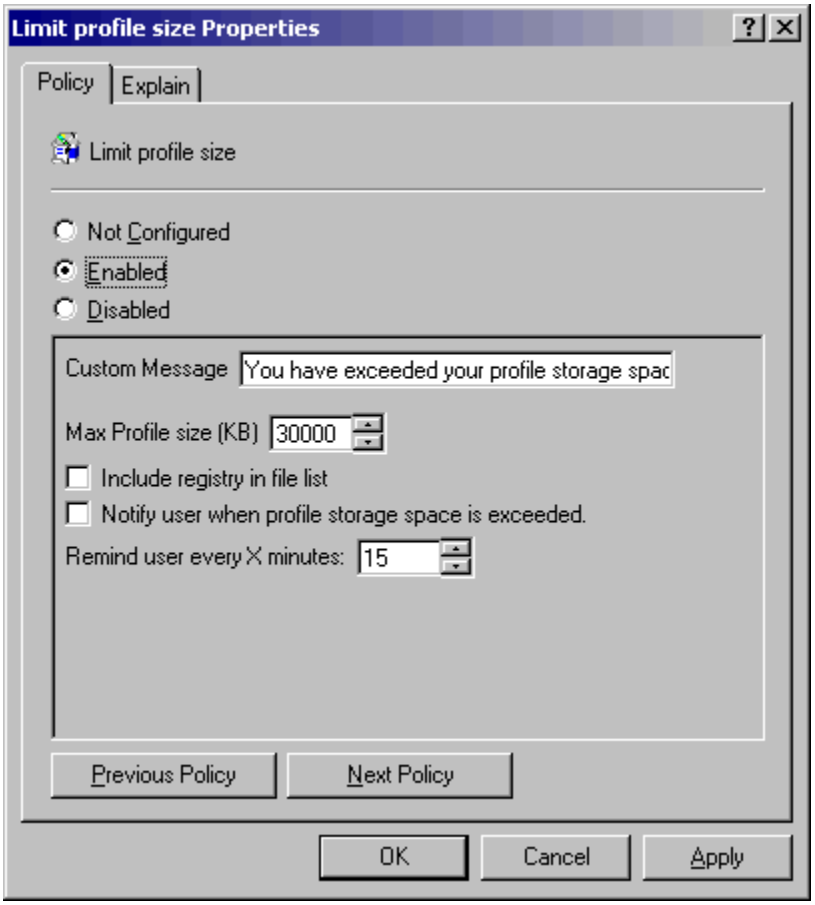

Figure 3.12: Win2K Limit Profile Size Policy.

With profile quotas in use, an icon is displayed that will show the available profile space when the cursor is placed over it, as Figure 3.13 shows.

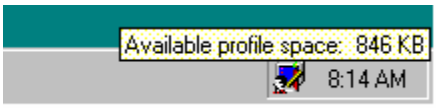

Figure 3.13: The Profile Quota icon.

To configure NT drive quotas, NT SP4 includes ProQuota.exe, which you can use with System Policies to enforce profile quotas. To configure profile quotas, perform the following set of tasks:

28. Ensure clients have the ProQuota.exe file installed. If not present, you can manually copy it to the %systemroot%\system32 directory.
29. Start the System Policy Editor (%windir%\PolEdit.exe).
30. Select Policy Template from the Options menu, and from the %SystemRoot%\inf directory, load the templates Common.adm and Winnt.adm.
31. Open Default User, and expand Windows NT User Profiles.
32. Select Limit Profile Size (see Figure 3.14), and specify the following values:

- Custom Message—This text will be presented to users when their profile size is exceeded.
- Max Profile Size—This field specifies the maximum size of a user's profile in kilobytes (the default value is 30,000Kb).
- Include Registry In File List—This check box specifies whether the Ntuser.dat file is to be presented to users in the list of user profile files that are currently in use (to help them determine which files can be removed from their profile). Obviously, you do not want a user removing Ntuser.dat, and whether you select the option to include it in the list or not, it cannot be deleted.
- Notify User When Profile Storage Space Is Exceeded—When this check box is selected, users will be presented with a warning message as soon as their profiles reach the quota size. By default, the message is only presented to users when they try to log off of the computer.

33. Save the policy as Ntconfig.pol, and store it in the Netlogon share (replication export directory on the Primary Domain Controller—PDC—for example, C:\WINNT\system32\Repl\Export).

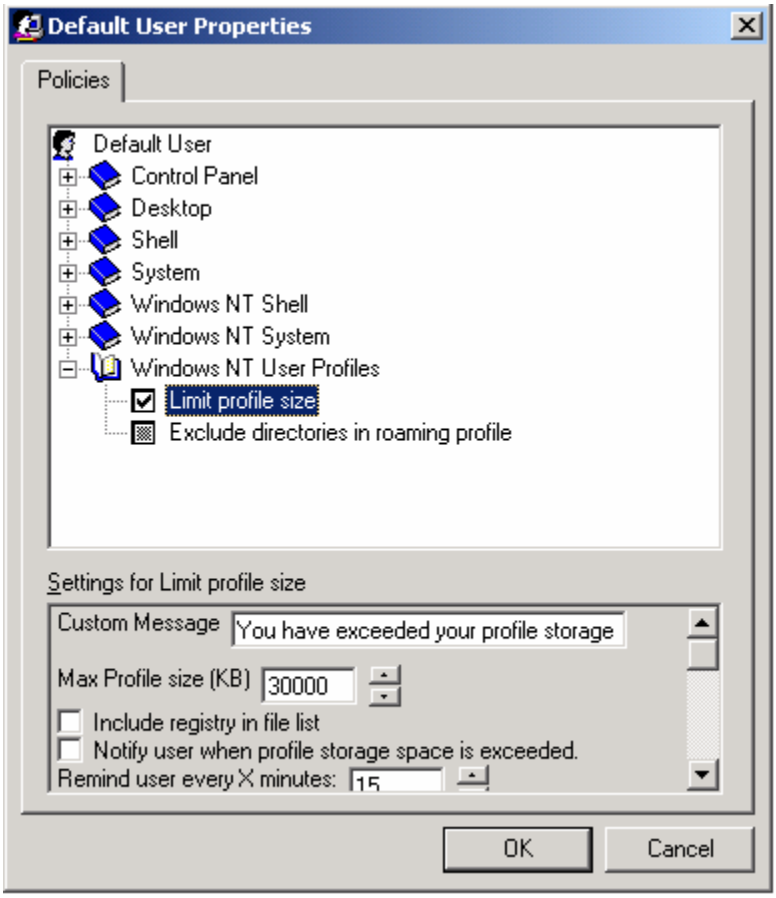

Figure 3.14: The Limit Profile Size policy.

Users will not be able to log off if the user profile quota is exceeded. By default, small files (less than 2Kb in size) are not listed in the dialog box that displays the files contained in the profile. If IE is installed, there could be a great number of small files that will affect the size of the user profile in the Temporary Internet Files folder. This cache uses a small percentage of the total drive space, but can easily grow to be several megabytes in size. To delete these files, the user will need to empty the cache through the IE Internet Options dialog box.

Be aware that a user can log off with an exceeded quota size by killing the ProQuota.exe process in Task Manager. Thus, defeating your imposed limitation.

Post Deployment Profile Configuration

A well-planned and tested default user profile is a good start for users, but over time any environment will undergo change: new printers, new software, new groups, and new shares. An implementation that dynamically dictates configuration and changes to settings will be best suited to handle these inevitable changes. Furthermore, the more that is taken into account dynamically, the less a default user profile is of any consequence.

Dictating Settings at Startup

A dynamic means of configuring machines or users, Group Policy provides centralized control of systems and can be easily modified for any or all users through its Startup and Shutdown script capabilities. By enforcing changes to a computer at startup, a potentially lengthy logon period can be dramatically reduced. However, changes made to the system at startup (and optionally, at shutdown) occur when no user profile is available for modification.

Despite this limitation, the more you work to customize your environment, the more you will realize a need to modify settings outside the user profile. Security settings, software installation, and changes to the All Users profile or HKEY_LOCAL_MACHINE hive of the registry are prime examples. Instead of working to bypass security and perform such actions during logon, changes might be more appropriately set to take place before a user even logs on.

Keep in mind that, like System Policies, Group Policy enforces settings. If your desire is to provide a default setting that might be modified by a user, policies are not the way to go. System Policies *enforce* settings at logon, and Group Policy *enforces* settings at startup and logon as appropriate. Further, Group Policy will reapply itself to the system at a specified interval. By default, this re-application occurs every 90 minutes, with a random offset time of as long as 30 minutes (to prevent all clients from requesting Group Policy at once).

Dictating Settings at Logon

Group Policy and System Policy can enforce settings at logon as well. Additionally, logon scripts run in the context of the user logging onto a system, providing the ability to modify the user profile. Typically implemented using scripting languages such as KiXtart, VBScript, or the DOS Shell, administrators can dynamically customize the user environment based on several criteria, including:

- Group membership
- Computer name
- IP address
- Windows version
- Organizational unit (OU—in AD)

Depending upon a user's location, you might want to map certain printers. Depending upon what group a user is a member of, you might want them to map a shared network folder. Depending upon the OU that the computer is in, you might want certain software configured. This customization can be a rather involved endeavor depending upon your requirements.

> 🕮 We will discuss custom scripting in more detail in Chapter 7.

Additional Tools

Third-party tools have been introduced to help administrators maintain control and configuration of user settings. Although a solution tailored specifically to your environment might be realized using scripts and other command-line tools, such an implementation can be challenging, complicated, and limited to the capabilities of your staff. In this section, we will touch on the capabilities of third-party tools designed to help employ a customized solution without the need to write scripts.

ScriptLogic

ScriptLogic provides a completely point-and-click interface for customizing a user's desktop environment based on a wide variety of conditions. Figure 3.15 shows the Printers tab, one of several items that you can customize.

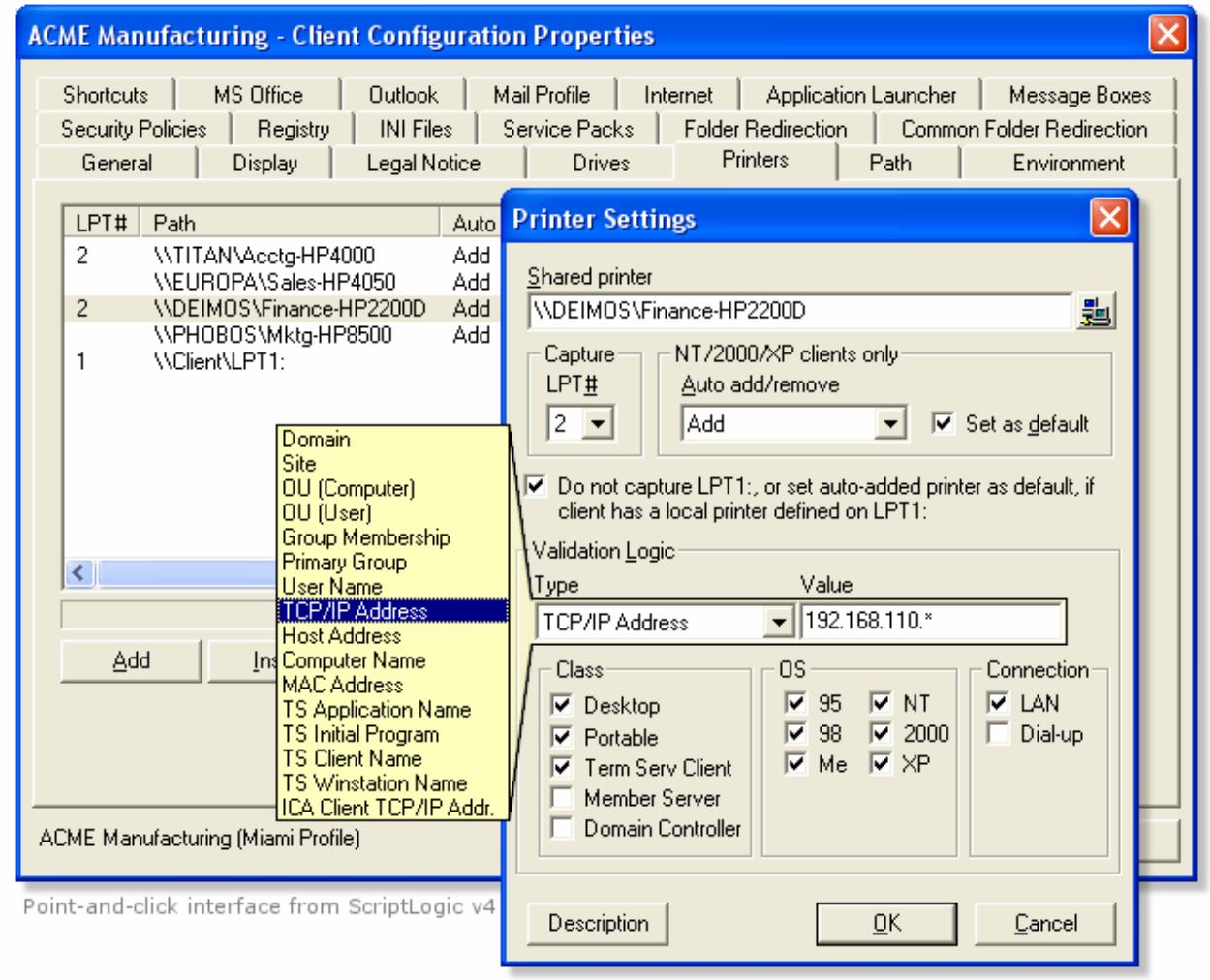

Figure 3.15: Using ScriptLogic to customize a user's desktop environment.

The following conditions, for which actions can be based (as supported by ScriptLogic), provide a good example of how you can implement a truly dynamic configuration:

- Domain
- Site
- Computer or User OU
- Group membership
- Primary group
- User name
- TCP/IP address
- Host access
- Computer name
- MAC address

- Terminal Services application name
- Terminal Services initial program
- Terminal Services session name
- ICA client TCP/IP address

Additionally, you may focus actions on specific versions of Windows, connection types (LAN or dial-up), and class of systems (desktops, portable systems, Terminal Service clients, member servers, and domain controllers). It also provides the ability to run at logon or logoff (on all Windows platforms) and can use an alternative security context to perform actions that you might otherwise be unable to carryout within the user security context.

Visual KIX

Another solution for gaining the ability to customize and dictate configurations without writing scripts is Visual KIX. This tool provides the ability to accomplish many basic actions one would typically include in a logon script, including drive mappings (see Figure 3.16), registry changes, program launches, time synchronization, and the installation of service packs.

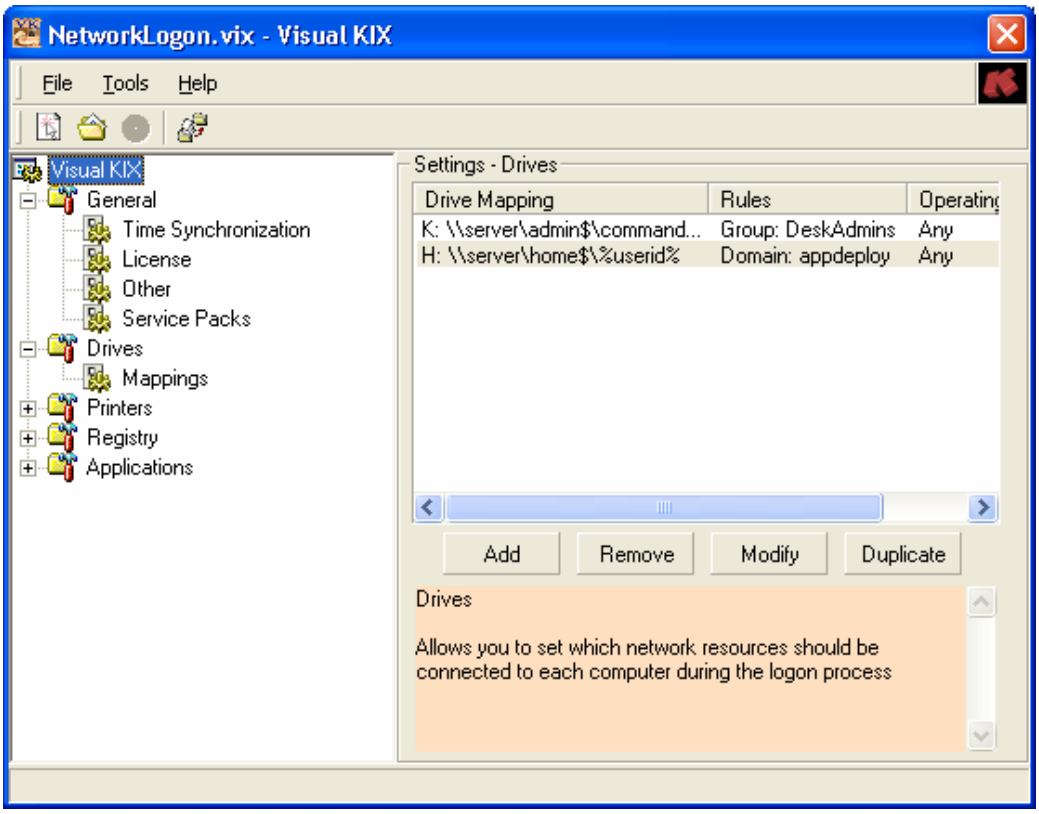

Figure 3.16: Visual KIX GUI.

Like ScriptLogic, Visual KIX also provides an automated method for replicating your final configuration to the network from which it is called through the normal logon script process.

Both ScriptLogic and Visual KIX utilize the KiXtart scripting language as their core engine, which illustrates the power and usefulness of KiXtart as a Windows logon scripting language. For more information about KiXtart, visit http://www.KiXscripts.com, http://www.KiXtart.org, or http://scriptlogic.com/kixtart.

Backing Up and Restoring Profiles

By now, we have established that user profiles are certainly critical to users, especially considering the potential for the inclusion of data created by the user. It therefore stands to reason that you would want to back up user profiles.

Traditional Backup Systems

In environments in which roaming user profiles have been implemented, simply backing up the share to which they are stored is a simple and effective way of maintaining a backup. Scheduling a regular backup of your roaming profile share to occur daily or weekly is recommended for any network implementation of roaming profiles. Since the release of NT, backup software has been included as a utility within Windows. Win2K introduced a significant update to the backup software through its licensing of Veritas's backup software (see Figure 3.17). Most networks have a backup mechanism of some kind in place; simply ensure that roaming user profiles are included in the backup and (if possible) given special consideration for quick recovery.

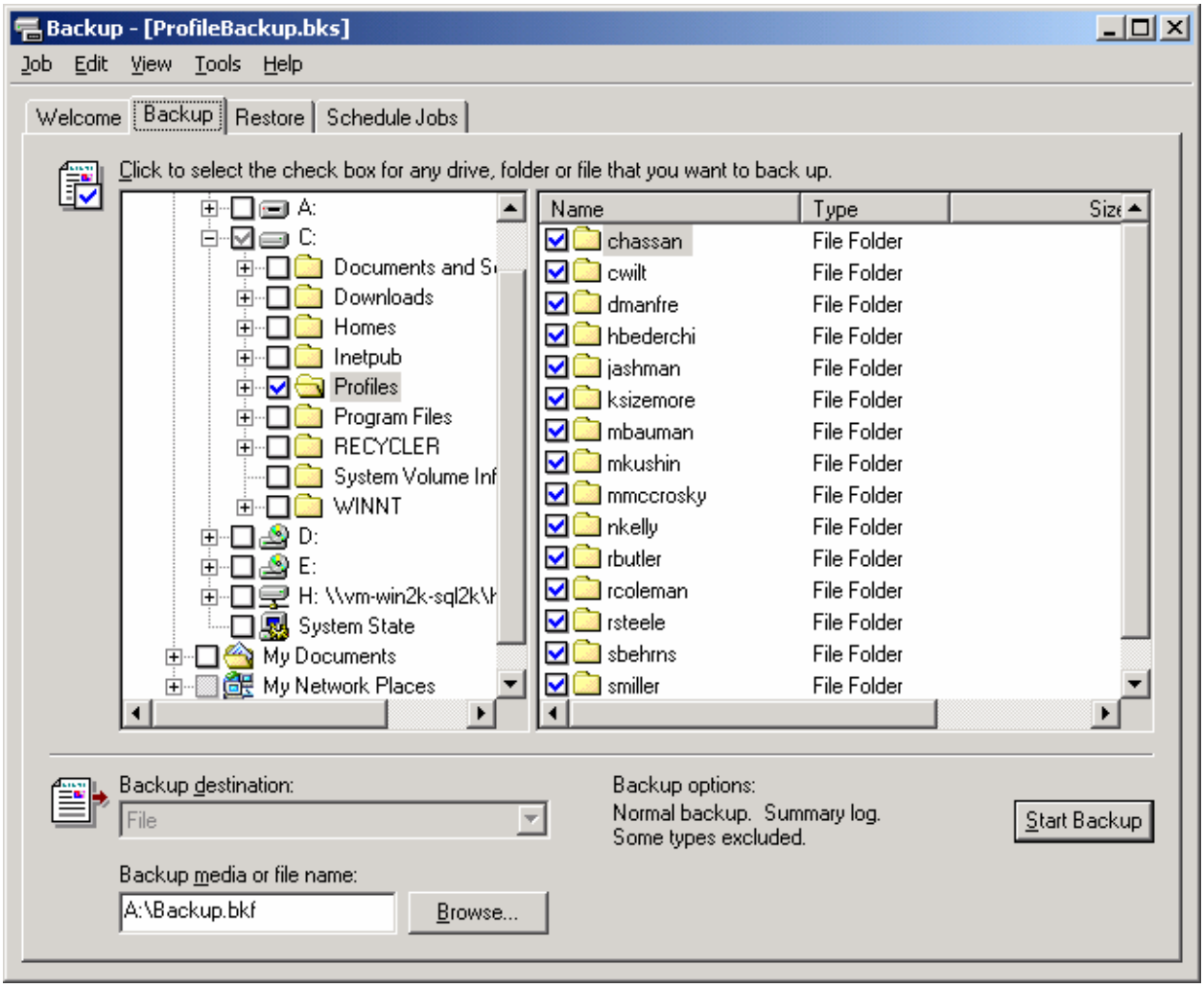

Figure 3.17: Microsoft Windows backup.

Scripted Copies

A scheduled script to perform a backup of user profiles to another folder or server share is another commonly employed method of maintaining backups of user profiles. Either copying a network share where roaming profiles are stored or remotely accessing each local workstation, keeping an online backup makes for quick and easy restore. The following text shows a simple example of how you can create a backup copy of your roaming user profiles:

```
Robocopy D:\Profiles \\BackupServer\Profiles /E /PURGE
```

This command line will copy all files from the local D:\Profiles folder to the \\BackupServer\Profiles share. The /E switch instructs Robocopy to include subdirectories, including empty ones, and the /PURGE switch instructs Robocopy to remove any files from the destination directory that do not exist in the source location.

> ✎ Robocopy (Robust Copy) is available from in Microsoft resource kits (NT and later versions).

Migration Tools

There are several tools on the market to help in the migration of user data and settings from one system to another. These tools are geared toward those moving to a fresh new system who want to retain their desktop appearance and settings (including data files stored on the system identified by either directory or file extension). A fresh system is typically introduced via the deployment of a new OS or baseline deployment. With the exception of manual or scripted updates to the Windows OS, user settings and data are typically lost. User data may be taken out of the equation if folder redirection is in use, but settings must often still be taken into consideration. You can use third-party migration tools to migrate from one version of Windows to another or as a part of your desktop recovery process. Additionally, most of the imaging tools on the market today include a user settings and data migration tool.

> 🕮 You can find links, information, and reviews of the many migration tools on the market at http://appdeploy.com/tools/migration.asp.

Troubleshooting Profiles

Because profiles are a distinctly separate element of the user environment, they can be used in troubleshooting. When there is a problem with someone's machine, some administrators promptly delete the user profile to see whether the problem goes away. Doing so can cause considerable trouble for users who have worked to define their default settings and even the appearance of their desktop. It might seem unimportant, but if users will spend time restoring these lost customizations, they are not doing there jobs—and as the one who deleted their profiles, it's your fault!

Is there problem with an individual application? Try removing just these applications' registry settings or try deleting the application data folder from the user profile. If you determine that it is necessary to delete a user's profile, it is usually sufficient to simply delete the user's registry settings (Ntuser.dat). Better yet, simply rename the file so that it can be easily recovered if it turns out not to be the problem.

Unable to Load Profile

If the problem is that a user is receiving an error and is unable to load the user profile, the troublemaker could be one of the following:

- Permission on the %SystemRoot%\Profiles directory has been modified; the Everyone group requires Full Control of this folder to load profiles.
- Not enough drive space or a registry size limit has been exceeded—a user profile will fail to load if either is true.
- If the local or roaming copy of Ntuser.dat (or .man) is corrupted, an error will occur and the profile will fail to load.

> ✎ Win2K and Windows Server 2003 include support for Microsoft Encrypted File System (EFS) but neither supports the use of EFS within roaming user profiles.

Troubleshooting Profile Problems with UserEnv.log

The UserEnv.log file is a very helpful tool for troubleshooting the process of loading and unloading user profiles. Each step in the user profile process is identified in this log file, including informational and error messages.

Support for logging is built into Win2K and later. Simply make the following registry edit to enable logging:

Key	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon
Value	UserEnvDebugLevel
Type	REG_DWORD
Data	10002 (Hexadecimal)

For NT, there is another version of the UserEnv.dll file, referred to as the "checked" version, which is identical to the retail version except that it contains debug flags that you can set and use with the kernel debugger. You can obtain the checked version of this file in either the NT Device Driver Kit (DDK) or the NT SDK. In addition to using this version of the UserEnv.dll, you must also set a registry entry. To enable UserEnv.log in NT, perform the following actions:

34. Rename the file UserEnv.dll in the %systemroot%\System32 folder to Userenv.old.
35. For the client machine to be debugged, copy the checked version of UserEnv.dll to the %systemroot%\System32 folder.
36. In the registry editor, add the following registry value and reboot the system:

Key	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon
Value	UserEnvDebugLevel
Type	REG_DWORD
Data	10002 (Hexadecimal)

> The UserEnv.log file is created in the root directory of the C drive. This file is a simple text file that you can view using a text file viewer. On Windows XP systems, the log file is created at %WinDir%\Debug\UserMode\UserEnv.log.

Listing 3.1 shows a sample log output from UserEnv.log on a Windows XP Pro computer.

```
USERENV(298.d0) 20:38:49:639 PingComputer: Adapter speed 100000000 bps
USERENV(298.d0) 20:38:49:639 PingComputer:  First time:  0
USERENV(298.d0) 20:38:49:649 PingComputer:  Fast link.  Exiting.
USERENV(298.d0) 20:38:56:981 ProcessGPOs: network name is
mindspring.com
USERENV(298.d0) 20:39:04:252 ProcessGPOs: Computer Group Policy has
been applied.
USERENV(298.d0) 20:39:04:252 ProcessGPOs: Leaving with 0.
USERENV(298.d0) 20:39:04:262 EnterCriticalPolicySection: Machine
critical section has been claimed.  Handle = 0x110
USERENV(298.d0) 20:39:07:867 LeaveCriticalPolicySection: Critical
section 0x110 has been released.
```

```
USERENV(298.d0) 20:39:07:877 GPOThread:  Next refresh will happen in
105 minutes
```

Listing 3.1: Sample log output from UserEnv.log on a Windows XP Pro system.

Using Profiles with Multiple Versions of Windows

With roaming profiles, users are not restricted from logging onto computers running different versions of Windows. However, NT user profiles are not compatible with Win2K systems A Win2K server can act as a repository for any type of profile, even an NT profile, but the system will not load an NT user profile, nor will an NT system load a Win2K user profile. One way to maintain roaming profiles on a network that has both NT and Win2K systems is to provide different accounts for logging onto each system.

Windows 9x profiles are not compatible with NT systems At first glance, the folder structure might look similar, but the incompatibility lies in the registry structure contained in the registry hive (User.dat and Ntuser.dat) files. The registry structures are completely incompatible between Windows 9x and NT. It is recommended that profiles for each of these OSs be stored in their own location. This implementation is fairly easy to configure and manage as Windows 9x profiles are stored in the user home directory, and NT profiles locations are specified in the user account profile itself.

Hard-Coded Profile Paths

Since NT, the %UserProfile% environment variable has been available as a reliable means of addressing the user profile directory. Despite this, some applications look to the %SystemRoot%\Profiles folder for user profiles. Because this had been the location of the user profile up until Win2K, it was not uncommon for software developers to assume this location. As a result, situations might arise in which a user logs onto NT without problems, but when logging onto Win2K, encounter a problem as a result of a hard-coded, invalid path.

System and Group Policy Assignments

NT enforces settings across the domain through System Policies. These settings are cached as part of the user's profile. Win2K enforces settings across the domain using (GPOs), which are added to the user's registry settings at HKEY_CURRENT_USER\Software\Policies, and Windows looks to this location for settings to be enforced on a user (overriding what may be contradicting settings elsewhere in the registry). If users log onto an NT system and have System Policies applied, then log onto a Win2K system and have GPOs applied, no technical limitation exists to prevent such a situation. However, the inclusion of both settings in the user's registry settings might increase its size.

> By default, NT calculates the registry size as 25 percent of the paged pool. The default page pool size is approximately equal to the amount of RAM. The maximum registry size is 152MB (80 percent of the paged pool, which is limited to 192MB). With Win2K, the default registry size limit is 33 percent of the size of the paged pool with the same maximum of 80 percent of the paged pool allowed. In Windows XP and Windows Server 2003, the registry files are mapped in the computer cache address space. Therefore, regardless of the size of the registry data, it is not charged more than 4MB. There are no longer any explicit limits on the total amount of space that can be consumed by hives in paged pool memory and in disk space.

Summary

In this chapter, we have discussed the user profile—its use and benefits. We covered the multiple types of profiles and the configuration of a custom default user profile. I discussed problems to avoid when using roaming profiles as well as ways to reduce their size through quotas and folder redirection. In the next chapter, we will explore another major part of establishing your users' work environment: deploying and upgrading the applications themselves.

Chapter 4: Application Management

In the last chapter, we covered the deployment of new systems. With your baseline identified, tested, and deployed, you must now work to keep that baseline up to date with the latest updates and new software requirements. Just keeping Windows itself up to date with service packs and a weekly release of hotfixes is enough to make some a little uneasy. Add to this task requests for new software and a constant barrage of updates, patches, and new releases for your existing software base and the need for a planned process and deployment mechanism becomes clear. In this chapter, we will discuss the creation of a process for deployment as well as the various methods available for the automated installation of software (and the benefits and drawbacks of each). Finally, we will cover a handful of the available software deployment systems with a summary of their capabilities.

The Software Deployment Process

Establishing a process to follow for the deployment of software is a crucial first step to success. When someone hands you a new application that they want sent to everyone in their office, what do you do first? Although larger networks demand a more structured process (largely for accountability), even the smallest of networks should have some sort of approval process. After approval, with which settings should the software be deployed? How should the software be deployed? Which machines should receive the software? These are all questions for which the answer should be crystal clear for every release. To realize this goal, a structured process must be in place.

Particularly when there are several people involved in software deployment, a process can keep everyone in the loop with what is going on. This communication is very important when the user comes back hours later to ask why new software hasn't appeared on his or her desktop yet. Having a documented process to show the user, even if it is a simple bulleted list, will help the user understand that there are steps to be taken—and that the process might take longer than he or she might have expected.

Additionally, a solid process will help to minimize deployments that negatively impact users. The deployment of packages that cause problems for users will hurt the reputation of your package and deployment team—the last thing you need is for users to fear receipt of new deployments and blame every problem they encounter on something you have "done to them."

We will discuss several elements that may be included in the makeup of your own deployment process. Use what makes sense in your own environment, but consider each:

- Receipt
- Integration testing
- Package research
- Package development
- Package testing
- Deployment testing
- Pilot deployment
- Full-scale deployment
- Baseline updates

Receipt

In most environments, there is a board (or individual) that approves software for a network. The request might get to this point through some chain of approval, but odds are you, as the desktop administrator, will not have to approve or disapprove software for use on the network. This, of course, is a good thing; after all, you have better things to do, right?

When software is approved and makes it into your hands to deploy, there are some basic first steps you should take before you start working on a package to deploy the software. The following sections detail these steps.

Deploying the Latest Version

Many organizations make it a policy not to deploy old versions of software. This approach is sensible considering that the update will require a second deployment more rapidly than if you had deployed the latest version from the start. Check the vendor's Web site to ensure that you have the latest version, upgrade, or patch.

> One helpful way to track the latest version of software is the online service from TechTracker called VersionTracker. This service notifies users of new versions by comparing the software inventory of a system with the service's online database. See http://www.versiontracker.com for more information.

Existing Older Versions

If a version of the software that you are to deploy already exists on the network, it is likely that this existence will affect how you deploy your new package. In some cases, previous versions must be installed before any new version is installed. If you are deploying software while users are logged on, you might need to ensure that the previous version is not in use before triggering your silent installation. Finally, if you have repackaged a previous version as an MSI setup, you will need to synchronize the component global unique identifiers (GUIDs) of the two packages and properly configure the new MSI as an update of the previous one. For more information about MSI update packages, see the sidebar "Creating Update MSI Files."

Creating Update MSI Files

Vendor-provided MSI files *should* follow Windows Installer rules for updates. However, if you are developing a setup for an internally developed application or repackaging a commercial off-the-shelf (COTS) application, there are some additional considerations for the proper configuration of an MSI update.

The version number of the MSI must be incremented. Major and minor updates are dictated by which version number is incremented. For example, 1.1.0.0 to 1.2.0.0 would be a minor update and 1.1.0.0 to 2.0.0.0 would be a major update. For more information about MSI version numbers, see the article by Matt Goedtel, "The Importance of the Version Field," at http://myitforum.com/articles/6/view.asp?id=5075.

Depending upon the type of update, a varying number of required GUID updates are necessary. GUIDs to be updated might include the PackageCode, ProductCode, and ProductVersion. See the Windows Installer SDK for more details (online at http://msdn.microsoft.com/library/en-us/msi/setup/patching_and_upgrades.asp).

GUID synchronization for like components is also important to the MSI update process. All elements of an MSI package are identified by GUID values. Your MSI authoring/repackaging software randomly generates these values for you. When the same file and version is deployed in one or more packages, such as in an update MSI file, it is important to use the same GUID to identify the files so that the reference counts can be properly maintained by Windows Installer (keeping the system from removing the file from the system during uninstall when another application is still using it). For more information about component, package, product, and upgrade codes, visit http://www.installshield.com/aboutus/news/newsletter/0012_50ef-3.asp.

📖 For more information about Windows Installer, read Jeremy Moskowitz and Darwin Sanoy's *The Definitive Guide to Windows Installer Technology for System Administrators* (Realtimepublishers), a link to which is available from http://www.realtimepublishers.com.

Media Tracking

When you receive installation media (CD-ROMs, floppies, DVDs, and so on), it is a good idea to have a way to store and track them in place. In the pile of CD-ROMs that covers your desk, it can be easy to misplace one, and doing so can obviously be a big problem. Many organizations store installation files on a network share. In fact, doing so is often a necessity because a scripted or command-line install will require the source files to do the job. Put a logical directory structure in place that will help you to quickly identify the software you are looking for in the future. Figure 4.1 shows an example.

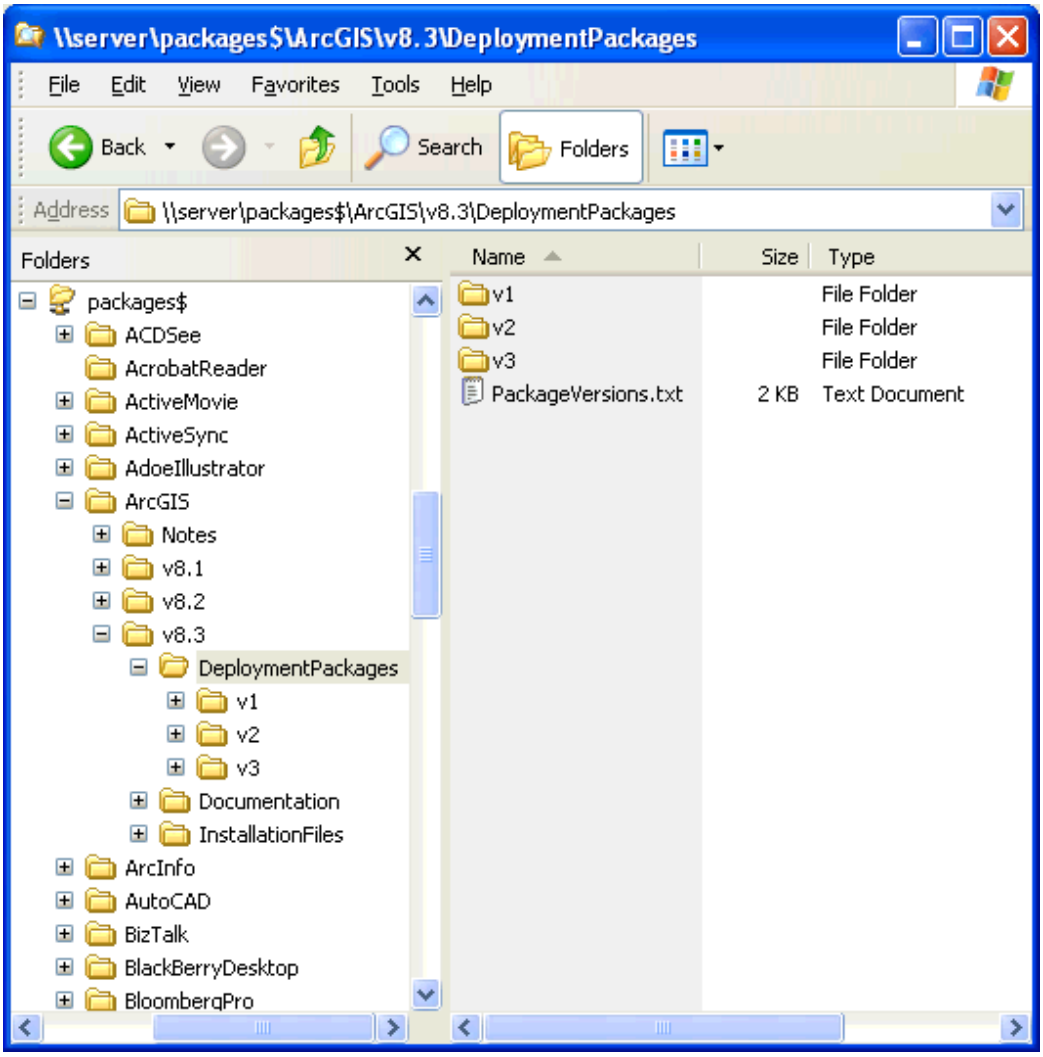

Figure 4.1: An example package directory structure.

> ☞ You can save yourself some trouble by avoiding use of spaces in the names of shares, directories, and packages. When a space is used, you will need to enclose the entire path in quotes when calling it in Windows. When a path to a file might be a command-line parameter, enclosing paths in quotes can make complex command-line syntax even more difficult. Eliminating spaces or substituting them with an underscore character (_) can help you to avoid making long command line calls more difficult than they need to be.

With the installation software on the network where you need it, return the CD-ROM to the owner. Some organizations have a media library or configuration management department that stores installation media and tracks licensing. Returning the media once you have what you need minimizes the time it is in your possession. If you must store the installation media yourself, put together a simple filing system to track its point of contact and the date it was obtained. It is also a very good idea to have a place to record the date, a return date, and initials. Then when someone comes to take the CD-ROM back, have that person initial your file for accountability purposes.

☞ Some setups require that the original CD-ROM be used. Either due to a poor setup process or an intentional restriction to deter pirating, it might be necessary to repackage the application directly from the installation media. You might be able to "trick" setup into thinking it is being installed from a CD-ROM by sharing the folder in which you have stored the media and mapping a drive letter to that location. Doing so will make it appear at root, like it would if you were installing from a CD-ROM. Still other setups check for a volume label as a condition of installation. To satisfy this restriction, you must set the volume label for the drive containing the shared folder. If you are repackaging, just work from the removable media to create your package. If you are working to script the installation, steps like these might be necessary depending upon the limitations accidentally or deliberately imposed by the vendor.

Integration Testing

If you have different versions of Windows on your network or have placed additional security restrictions on your workstations, the software might have difficulty running at all. Before you create and start troubleshooting what might appear to be a problem with your deployment package, first ensure that the application is able to operate within your unique environment when installed manually.

Installation and Usage Documentation

Most software installation wizards provide some number of prompts, selections, and buttons to be addressed during the installation process. If you simply install the software as you think it should be installed, you will inevitably find yourself in situations in which the package does not reflect the desires of the user requesting the software.

As long as integration testing must be done, taking the time to do so with the user requesting the software can be a valuable exercise. Install the software with the user, and document each selection. Although the direction might simply be to "accept all defaults," the user might not be truly aware of what the default settings entail. Some installation dialog boxes might even require that you make a decision that does not provide a default selection—what then? Screen shots of the installation wizard might be overkill, but a simple bulleted list of what was done during installation can let others see the process and might even help you 200 packages later when you have long since forgotten the specifics of the installation.

Another piece of information easy to overlook is the file used to launch the installation process itself. It is not always as simple as a setup.exe on the root of a CD-ROM. There are times when an installation media might have several directories and setup files. Having clear documentation of the one you used can save time otherwise spent hunting around for the proper file. The following list details a very simple example of a software installation document:

- Start install: \bin\eaxsetup.exe
- Accept EULA
- Accept default installation directory
- Begin installation
- OK to reboot

Application Interaction Considerations

What will the impact be on other applications? Many applications depend on components such as the Java Runtime Environment (JRE), Oracle Client, and Microsoft Data Access Components (MDAC). The version (or in some cases the location) of such an installation might significantly affect other applications on a system.

Particularly in situations in which there is an internal development staff or companies are contracted to write custom software for an organization, it is imperative that your baseline configuration be communicated. The version, location, and settings for elements such as MDAC are important to consider early in the development stages. Guiding *custom development* to function in your *custom environment* is an often-overlooked aspect of internally developed or contracted applications. Establish your requirements up front and avoid problems in deployment and compatibility down the road (for more information about how to do so, see the sidebar "Establish Installation Requirements.")

Establish Installation Requirements

When custom software is developed for an organization, it is naturally the functionality of the application that is of paramount concern—so much so that little attention is given to installation, which can cause great expense when you begin to attempt to integrate the application into your environment. Reengineering or repackaging the setup of an application developed especially for your organization is a wasteful process that introduces unnecessary risk. Some points to consider:

- Provide documentation about your build process. If an image is used, provide a copy for use by your internal development staff or contracted companies. If your image only works on certain hardware, give contractors or developers (or have them purchase their own) compatible hardware. It is important that developers understand the environment in which their applications will be deployed in order to avoid assumptions that might cause alteration for the application to comply with reality.
- Provide system and Group Policy information, particularly those that enforce security settings. If areas of your baseline hard drive might not be written to by users, make sure that the development team understands where they *may* store data or configuration files. Often working in an environment in which users function as administrators of their workstations, it might not be apparent that in your environment users may not write to common user areas, particularly the HKEY_LOCAL_MACHINE hive of the registry. It is important that this possibility be communicated to avoid modifications to your baseline security in order to support the application.
- As stated previously, MDAC, JRE, ODBC, and Oracle Client are just some of the components that applications may be dependant upon. For example, if your environment is at MDAC 2.6, it is important that developers not develop their application using functionality not available until MDAC 2.7.
- Demand a Windows Installer installation package. MSI is well documented with several powerful tools available to create installation packages. It is a very "administrator friendly" format that provides you with maximum control and deployment options. Of course, a bad MSI package can cause problems as well—require that Windows Installer validation tools be used to ensure that MSI rules are followed. In particular, I recommend you require MSI validation be performed using the Independency Consistency Evaluator (ICE). This tool is a feature of most MSI tools and is available free in the Windows Installer SDK.

Package Research

Although many organizations will restrict themselves to one or two packaging (automated installation) methods, it's wise to review all of your options before moving forward. Being consistent about creating packages for your software deployments has its benefits—for example, you might document and become very proficient with the method you choose.

Some environments are restricted to the use of proprietary repackaging formats (potentially a limit of your deployment system). For example, although not a proprietary format, environments that rely upon the assignment of Group Policy Objects (GPO) as a method of deploying software are restricted to using the Windows Installer (MSI) package format. In a situation such as this, in which you have a single means of deploying software, you might well need to focus your packaging on one particular method. In the following section, we will cover several common packaging methods.

MSI

Microsoft's Windows Installer (MSI) setup format was first introduced with the release of Microsoft Office 2000. This introduction provided a documented standard for installation with rules, structure, and format. MSI files are databases of installation components and instructions that all follow the same rules regardless of the tools used to generate them. Among quite a few other features, MSI files offers support for a command-line driven, silent installation using the following options:

- /qn—No user interface whatsoever
- /qb—Basic user interface (progress bar only) with no modal dialog boxes
- /qb!—Same as previous listing, but with Cancel button disabled

In addition to these options for silencing installation from the command line, you can use public properties (variables within an installation that can be dictated at the command line) to customize the installation even further. The following list provides several properties that might be helpful in a command-line installation (for more information about these properties, see the Windows Installer SDK):

- REBOOT—Dictates how the system handles reboot requirements by dictating whether Windows Installer should force, suppress, or *reallysuppress* the reboot action. The use of force is clear, but the difference between suppress and reallysupress is that suppress will cause an automated reboot if the installation is being run silently (reallysuppress bypasses the reboot action regardless of how the MSI is being run).
- ALLUSERS—Unlike legacy setups that specify a full path to where shortcuts and configuration information are to be stored, MSI handles this itself. Depending upon whether the installation is being performed for the user or the computer, shortcuts are placed in the user profile or the All Users profile as appropriate. The ALLUSERS property allows control of this behavior from the command line.
- INSTALLDIR—Specifies the directory in which the application should be installed. A default directory will always be specified within the MSI, but you may alter this location using the INSTALLDIR property.
- ROOTDRIVE—By default, the Windows Installer will choose the fixed drive with the largest amount of free space as the destination drive for an installation. To override this behavior, you can specify a drive letter (include a trailing backslash when specifying a value for this property).

- ARP—Several properties are available for controlling how an installation is presented within the Add/Remove Programs applet. For more documentation about properties that control the behavior of the Add/Remove programs listing for an application, see the AppDeploy.com FAQ listing on the subject at http://appdeploy.com/faq/detail.asp?id=58.

> Public properties are always identified in upper case. It is these public properties that you can set at the command line.

Environments that utilize AD's Group Policy as a means of software distribution are limited to this format when assigning software to computers. Because many software vendors have not moved to the MSI setup format, administrators today often repackage installations into this format. Repackaging is covered later in this section.

> For more information about Windows Installer, see Jeremy Moskowitz and Darwin Sanoy's *The Definitive Guide to Windows Installer Technology for System Administrators* (Realtimepublishers) at http://www.realtimepublishers.com.

There are an increasing number of vendors making their software available in the MSI format. The development of MSI packages can be a complex process, which has been the primary reason this format has taken so long to catch on. As a bit of encouragement, Microsoft has made the availability of an MSI setup a Win2K and Windows XP logo requirement. When it comes to the MSI setup format, there are no specific implementations; all kinds of software from games to business software are available as MSI setups.

Is that setup an MSI? Just because you see a setup.exe at the root of your installation CD-ROM does not mean it is not an MSI setup. Because people expect to see a setup.exe, and because not every machine has the Windows Installer service installed, a setup.exe is often provided. The executable first checks to see whether the required version of Windows Installer is present on the machine. If it is not, it gets installed and the MSI setup is initiated. If the desired version is present, it simply calls the MSI setup. If the MSI file itself is stored in a subdirectory rather than the root of the CD-ROM, the situation can be confusing. A giveaway that an MSI setup is initiated is, of course, the display of a Windows Installer dialog box when the setup is launched; you can also be reasonably sure you are dealing with an MSI if you see the Windows Installer setup files in the directory with the setup.exe (see Figure 4.2). There are two such files, one for Windows 95/98/ME systems and one for NT, Win2K, and Windows XP systems: InstMsiA.exe and InstMsiW.exe, respectively.

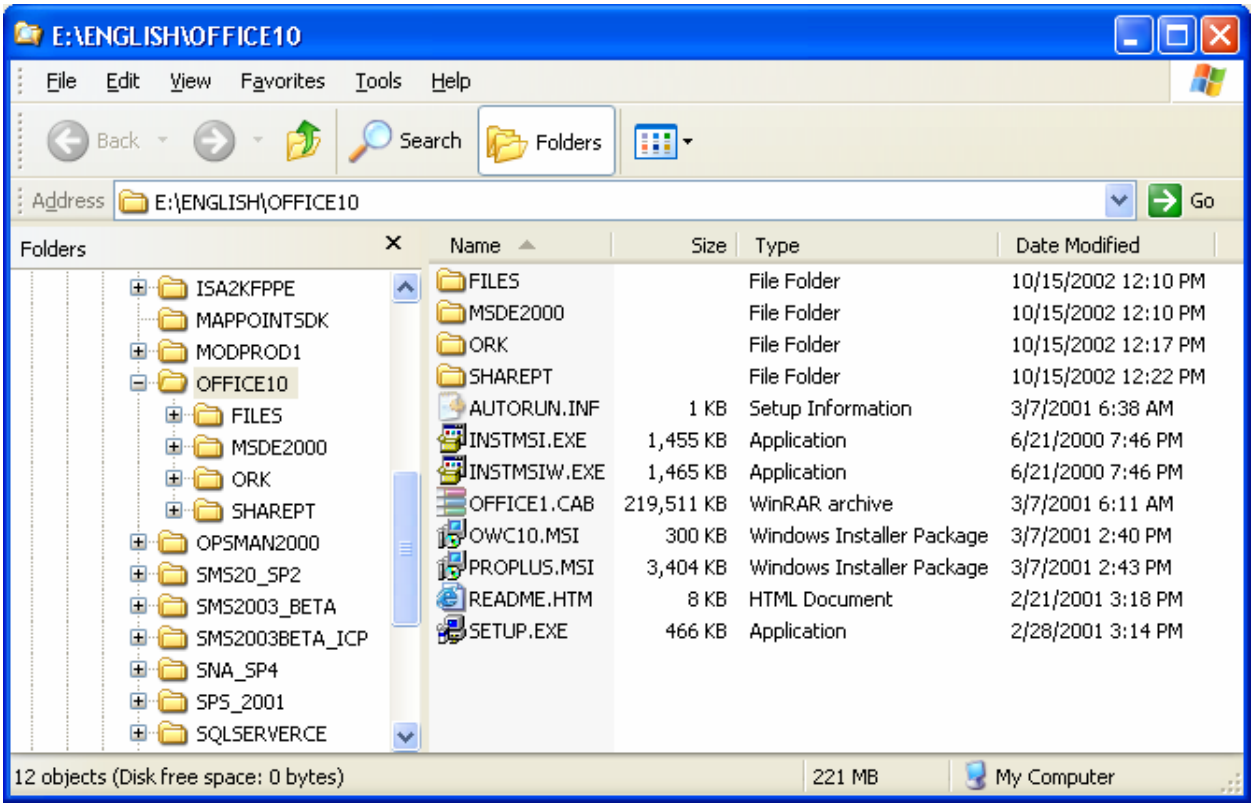

Figure 4.2: Directory of MSI Installation with Setup.exe and Windows Installer support files.

> You should never repackage an MSI setup. You might be able to generate a successful installation by doing so, but upgrading or removing the installation will cause you big trouble down the road. Before repackaging, make certain you are not dealing with an MSI setup. Self-extracting archives that contain MSI setups can trick you into thinking that the setup you are working with is not an MSI. See the sidebar "Accessing Source Files Within Self-Extracting Executables" later in this chapter for more information.

Benefits

Complete control of an MSI setup can be realized without modifying the vendor's MSI directly thorough the use of transform (MST) files. You can use an MST to specify which features to include, move or delete shortcuts, and add files or registry entries. Some software vendors have even provided MST creation wizards to facilitate the customization of your deployment. Microsoft Office provides a Custom Installation Wizard to generate an MST, and many software vendors have licensed InstallShield's Tuner product to provide similar functionality, including IBM's Lotus Notes 6 and NetManage's RUMBA.

Self-healing and install-on-demand are two powerful new features Windows Installer brings to the table. Windows Installer shortcuts check for key files when launched and will attempt to retrieve any missing files automatically. In this way, an application installed with MSI may also be configured not to install certain features until the user attempts to make use of them.

In addition, Windows Installer provides excellent command-line support, including the ability to set public properties. This feature provides you with extensive command-line control of an installation that will translate to every installation provided in this format.

Drawbacks

Windows Installer must be properly administered or it can cause user dissatisfaction in the way of repetitive calls to repair installations or prompting for needed installation files. In a network environment, you can avoid these kinds of problems through conflict checking and ensuring a resilient path to installation files. However, Windows Installer requires an increased level of planning and administration.

Updating an MSI can be technically challenging due to Windows Installer's requirements to recognize and properly perform an update. See the sidebar "Creating Update MSI Files" earlier in this chapter for details.

> A common misconception is that a patch file (MSP) is a way to quickly affect small changes after deployment; however, the creation of an MSP is actually additional (and as an administrator, often needless) work. To generate an MSP, you must first create an entire MSI for the new software update. The MSP is then generated by comparing the old and new MSI packages so that the file's size is reduced to include only the changes between the two. This is primarily done for reducing Internet download sizes or the need to move updates across a WAN link.

Command-Line Installations

Particularly with simple installations that require little or no configuration options, some software vendors provide a means of silently installing software from the command line. Microsoft, for example, provides command-line options for installations of hotfixes and other simplistic installations using a consistent set of command-line switches. The following list provides common Microsoft command-line switches:

- **/A** runs the setup in administrative mode. This option is available only when installing from the original media.
- **/B** specifies the type of setup to be performed. This can be a number, which refers to Typical, Compact, or Custom. The meaning of this number varies across Microsoft products.
- **/C** "" If your product was supplied with a 20-digit license key, use this option to enter the key and bypass the key validation dialog box. You must enter a space before the key and enclose it in double quotation marks.
- **/K** "" If your product was supplied with a 10-digit license key, use this option to enter the key and bypass the key validation dialog box. You must enter a space before the key and enclose it in double quotation marks. The license key can be found on the Product Authorization Certificate or on the CD-ROM casing.
- **/F** performs an installation using only short (8.3) file names.
- **/Q [0 | 1 | T]** runs the setup in silent installation mode. Use /Q1 to suppress the Exit dialog box. Use /QT to suppress all dialog boxes including the background frame window and progress gauge. The /A and /Q options are mutually exclusive.
- **/R** reinstalls the application.
- **/U[A]** Uninstalls the application. If /Q is also specified, the user is not prompted about removing shared components. /UA removes shared components without prompting the user.

Several other common switches can be attempted in an effort to determine support for command-line switches by setup.exe files. Several regularly used switches worth trying out include:

- -/QUIET
- -/UNATTENDED
- -/S
- -/Q
- -/SILENT
- -/SMS
- -/?

> ☞ Some switches are case sensitive and/or require a dash (-) or slash (/) depending upon their unique implementation of command-line switch support. Try several combinations before giving up on this option as a viable method of silent installation.

Microsoft hotfixes and simple utilities often support command-line installation switches where an MSI was not implemented. In addition, this installation method is often supported by applications designed for distribution (often with built-in distribution mechanisms), including Norton and McAfee antivirus software and management software such as the Altiris, BMC Patrol, Lanovation, and Cognet client agents.

Benefits

If your environment allows it, you should always take advantage of the options developed by the vendor. Success will be as likely as with a manual installation, resulting in your best chance at a clean install. Additionally, such an installation method is supported by the vendor, so you will be able to review documentation and make use of other vendor support services should you encounter any problems.

Drawbacks

Command-line installations often provide few options for customization (or none at all). If you want to install to a specific directory or to determine where any potential shortcuts are to be placed, you might need to use a custom script to perform such changes following the command-line install. As a result, many command-line installations take advantage of a configuration file, or answer file, as discussed next.

Answer File

An answer file lets you customize your installation and still enjoy the benefits of a vendor-supported installation process. Answer files are normally in an INI file format, but can also be as simple as a text file with one or two lines (such as where it should be installed).

To satisfy the demand of desktop administrators, many vendors have taken an effort to provide a supported means of silent installation. Such support provides you with the ability to customize a silent installation using the vendor's supported installation process. For example, InstallShield provides a native ability to create an answer file, then use that file to facilitate a silent installation. Many vendors are aware of this functionality, and have documented it as a supported means of installation. However, many other vendors who used InstallShield unknowingly provide this functionality—so it may or may not be supported by the product with which you're working. Still others who were aware of the capability and did not want to provide it, have broken the process so that it cannot be used.

There are several switches supported by InstallShield legacy setup.exe files. In particular, you will want to use the -R switch to create your answer file and -S to perform the installation silently using this answer file. Supported switches are described in the following list:

- **-S** initiates a silent installation using the answer file (.ISS) in the current directory or in the location specified by the F1 switch.
- **-SMS** holds the process until the installation is complete. When scripting or using a management system such as Microsoft SMS that maps a drive only for the duration of the installation, you should use the -SMS switch. Without it, control of the system is immediately relinquished and the only means of monitoring its activity will be via Task Manager's Process tab. This switch is case sensitive.
- **-R** records the dialog boxes and choices made during an installation process. By default, the answer will be named SETUP.ISS and will be stored in the System Root directory (typically C:\windows or C:\winnt). You can use the F1 switch to specify this location and file name.
- **-F1<path to ISS file>** specifies an alternative location and name of the response file (.ISS file).
- **-F2<path to log file>** specifies an alternative location and name of the log file created by the silent InstallShield installation. By default, a setup.log log file is created and stored in the same directory as that of the InstallShield setup.iss file.
- **-m<filename>** causes setup.exe to automatically generate a Management Information Format (.mif) file at the end of the setup (used by Microsoft SMS for reporting inventory and status messages). Do not include a path—the .mif file is always placed in the Windows folder.
- **-uninst** runs the setup as an uninstall.
- **-verbose** provides more detailed information when a setup.exe error occurs.

> ☞ You can pass switches in any order and precede them with the dash (-) or forward slash (/) character. With the exception of the SMS switch, these parameters are not case sensitive. When using long path and file names and those with spaces, enclose the switch in double quotation marks so that they are not treated as command-line delimiters.

The following list walks you through the process step by step:

37. Execute

    "setup.exe -r"

    at the command line. Note that the program will actually be installed on the system on which you perform this action.

38. Answer all dialog boxes through installation. If an option to reboot is presented at the end, you may click OK, but be aware that running the created answer file will reboot the system as well. You might want to specify that the installation wizard not initiate a reboot so that you can control it yourself using a script or utilizing the capabilities of your deployment tools.

39. Go to your system root directory, for example "c:\windows" and locate a file named SETUP.ISS.

40. Copy SETUP.ISS and the installation files (setup.exe and supporting directory structure) to a network share.

41. A simple batch file containing "setup.exe –s –SMS" will initiate a silent installation of the software using the SETUP.ISS file you have stored with setup.exe.

Accessing Source Files Within Self-Extracting Executables

Is your setup.exe several megabytes in size? This file is likely a self-extracting archive, which decompresses itself into the current %TEMP% directory prior to execution. To access these files so that you may attempt this answer file process, simply launch setup.exe (with no switches) and go to the %TEMP% directory to locate the extracted setup files. You can easily identify the %TEMP% directory using the %TEMP% environment variable—on Windows 9x systems, it is typically equal to C:\windows\temp, and on Win2K and later systems, it is typically equal to C:\documents and settings\<user name>\local settings\temp. The files are extracted to a subdirectory created in %TEMP% with a small randomly generated folder name as Figure 4.3 illustrates.

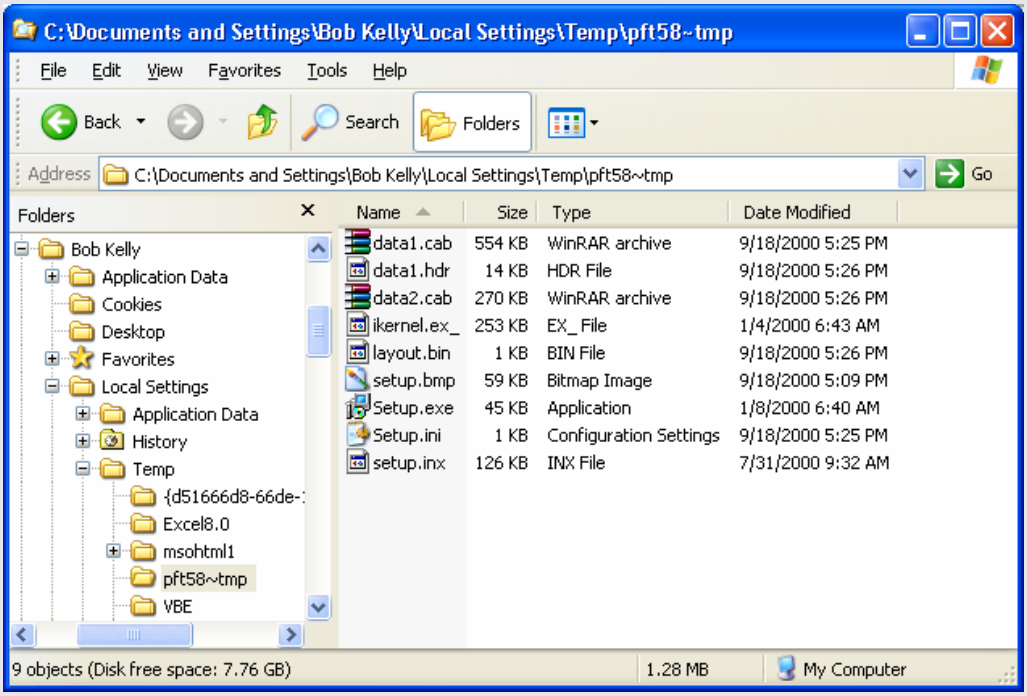

Figure 4.3: Directory structure of temp with an InstallShield temp directory present.

As stated, the vendor might be unaware of this capability or might have intentionally crippled this capability. One field commonly used to hide support is a license string field. Because the vendor does not want their software installed on multiple machines with the same license string, this field may be hidden from capture during the "-r" (record) process. Another common reason for failure is the presence of a dialog box during the silent installation that was not presented when you created your answer file. For example, if Netscape is on your system, you might be asked whether you want to install a plug-in, whereas, without Netscape, the question is never presented. For this reason, you must be sure to test this method of silent installation thoroughly to be confident of success.

Because the installation is performed silently, it is difficult to determine the cause of a problem when one is encountered. The setup.log file that is created contains a section titled ResponseResult. This section includes a ResultCode value that will contain one of the following values when complete—typically your only clue in determining what might have gone wrong when testing your installation. Table 4.1 presents the result codes.

ResultCode Value	Definition
0	The installation was successful
-1	A general error was encountered
-2	An invalid mode was specified
-3	Required data was not found in the ISS file
-4	The system does not have enough memory available
-5	A specified file does not exist
-6	The setup cannot write to the response file
-7	The setup was unable to write to the log file
-8	An invalid path to the ISS file was specified
-9	An invalid list type was specified (string or number)
-10	A specified data type is invalid
-11	An unknown error was encountered during setup
-12	Installation dialog boxes are out of order
-51	The setup cannot create the specified folder
-52	The setup cannot access the specified file or folder
-53	An invalid option was selected

Table 4.1: Result codes and their definitions.

Benefits

As with the command-line installation, answer files offer the advantage of a vendor's support as well as the benefit of using the installation logic developed for this application. In addition, when tested successfully, answer files provide the ability to customize the installation to the degree provided by the vendor in the product's installation wizard.

Drawbacks

When an answer file installation does not work, the installation process simply aborts. This action could be considered positive during deployment, but during development, you are left with no clue as to what went wrong outside the previously discussed result codes. “Dialog Out Of Order” is not very helpful when you are unable to see which dialog box is causing the problem.

Many setup files spawn other setup processes to install prerequisites or required dependencies. Because it is the setup.exe that is making the call to these installations, you are unable to alter this behavior in order to silence those spawned installations. You can try installing them beforehand to see whether the setup program will then bypass the call, but I have found this to work in very few cases and might not be supported by the vendor. If you have a problem, asking the vendor for help is not likely to be an option unless they have documented the process or otherwise acknowledged it as a means of installation. Give it a try, but when no command-line or answer file support is available to you, repackaging is a typical next resort.

Repackaging

The process of repackaging an application is typically performed by comparing the file system and registry of a computer prior to and after installing a program. The delta is packaged into an MSI, EXE, or proprietary format, which, when applied to a similarly configured system, will effectively install the software.

> ✎ I’ve had several people inquire about statements that often exist in EULA documents that state “You may not distribute or repackage the software without written permission.” This statement does not refer to packaging and distribution in terms that a desktop administrator would be familiar with. This statement is referring to bundling the software with software that you plan to sell (or otherwise externally distribute) outside of your organization, including using components of their software within your own software or your installation media.

When no support for a silent installation is made available by the vendor, repackaging is a common means of generating one yourself. If the setup is not an MSI and no command-line support is available, repackaging can provide you with an MSI, executable (EXE), or proprietary formatted installation that can give you the silent installation you desire.

Benefits

When you repackage a setup, you get exactly what you want—a silent installation with all the customization and settings you decide upon. A great number of vendors do not realize or want to address the common need for a remote or automated deployment of their software. As a result, repackaging the vendor-provided setup is often a desirable means of creating such support on your own.

Drawbacks

If you have a problem with a repackaged installation, few vendors will offer support. In addition, it can often be unclear as to what may and may not belong inside your repackaged version of the setup.

Package Development

With the installation media, your installation documentation, and your decided method of creating your software package, it is time to actually create your package. Your deployment system often dictates which methods you can use. Because a command-line installation (using switches or an answer file) requires very little work in this phase, we will focus on repackaging in this section (which requires the most work in this step of the process). For example, we will cover a couple of the most popular software packages for repackaging, focusing on those providing support for the Windows Installer (MSI) setup format desired by most administrators today.

> ☞ For a complete list of repackaging tools, for both MSI and legacy EXE packages, visit the AppDeploy.com tool area at http://appdeploy.com/tools/browse.asp?r=1.

OnDemand Software's WinINSTALL

OnDemand Software's WinINSTALL (see Figure 4.4) has a long history as an administrator tool for both the repackaging and deployment of software packages. Its repackaging capabilities were included in Win2K Server, and though Windows Server 2003 will not ship with a repackaging tool, OnDemand will be making an updated version of its repackaging software available for free in support of Windows Server 2003.

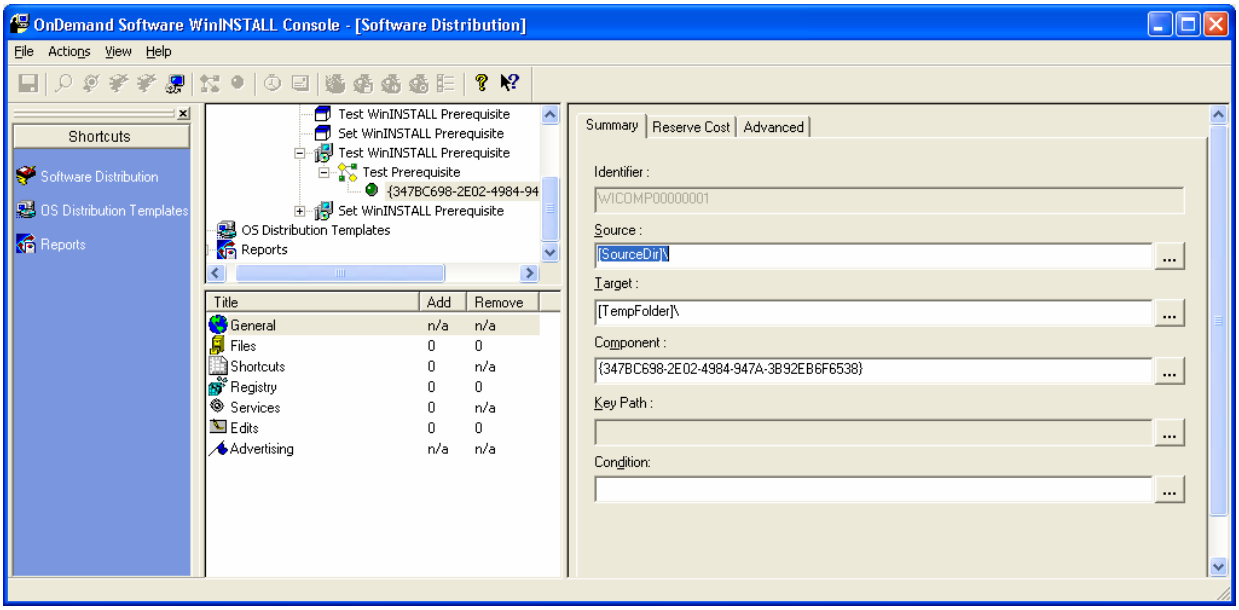

Figure 4.4: WinINSTALL console.

The following list offers some of the features that help WinINSTALL 7.5 stand out among similar tools:

- Provides delivery mechanisms for both push and pull distributions including calendar-based scheduling and deployments via an NT service or network logon script; it also provides an ability to perform an installation locally from a hidden partition
- Automatically synchronizes packages across distributed WinINSTALL servers

- Compares remote desktop and laptop application installations to the original image as a means of troubleshooting
- Automates notification of deployment actions via email, SNMP, or the Windows event logs
- Creates unattended installations of Windows OSs and lets you create and deploy them through the same GUI interface.

Wise Package Studio

Wise Package Studio also has a long history in the administrator community. The product's WiseScript repackaging capability has been around for several years and is familiar to many through Microsoft's licensing of an early release in the form of the SMS Installer. Wise expanded the interface and capabilities of WiseScript while providing a similar interface for the creation of MSI setup packages. Further developed into a full suite of tools for the creation, modification, and management of packages, Wise Package Studio (see Figure 4.5) also provides a powerful conflict checking and resolution tool that reads package components into a database so that they can be compared against selected packages to find potential conflicts.

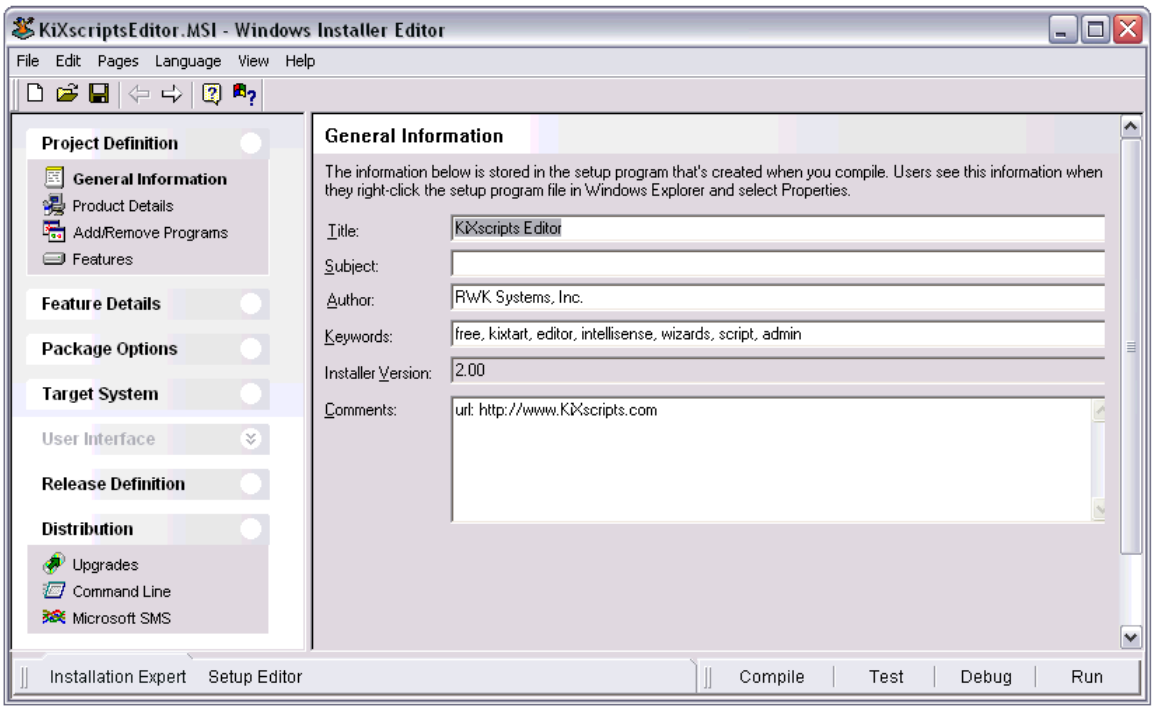

Figure 4.5: Wise Package Studio Windows Installer Editor.

Features provided by Wise Package Studio 4.0 worth mentioning include:

- Pre-built, customizable, "point and click" processes built on best practices promote consistent, reproducible results and reduce errors
- Workbench interface provides a project-based framework that guides you through all phases of a package and release process
- Provides the ability to control access to repackaging projects with user security features

- Workflow features let you assign project tasks to individual team members or add items to a project's "To Do" list
- Application Gateway feature provides a Web-based portal that tracks applications from request to distribution, providing a way of requesting an application and managing the application preparation process
- Virtual Capture feature provides an ability to perform captures on a "non-clean" machine without re-imaging between captures

InstallShield AdminStudio

InstallShield AdminStudio provides much of the same functionality present in Wise Package Studio. InstallShield is known as a leader in setup software and have a long history in its offering of such tools for developers. Although a repackaging tool has been available with the InstallShield Developer (see Figure 4.6) tool (and earlier releases under different names), it has not been until the release of AdminStudio that a directed focus on the administrator community has been prevalent. The company has put significant resources into this tool and has produced a very powerful feature set, offering several new tools and features with each release. In addition to its other functionality, InstallShield AdminStudio provides repackaging, validation, and conflict resolution features.

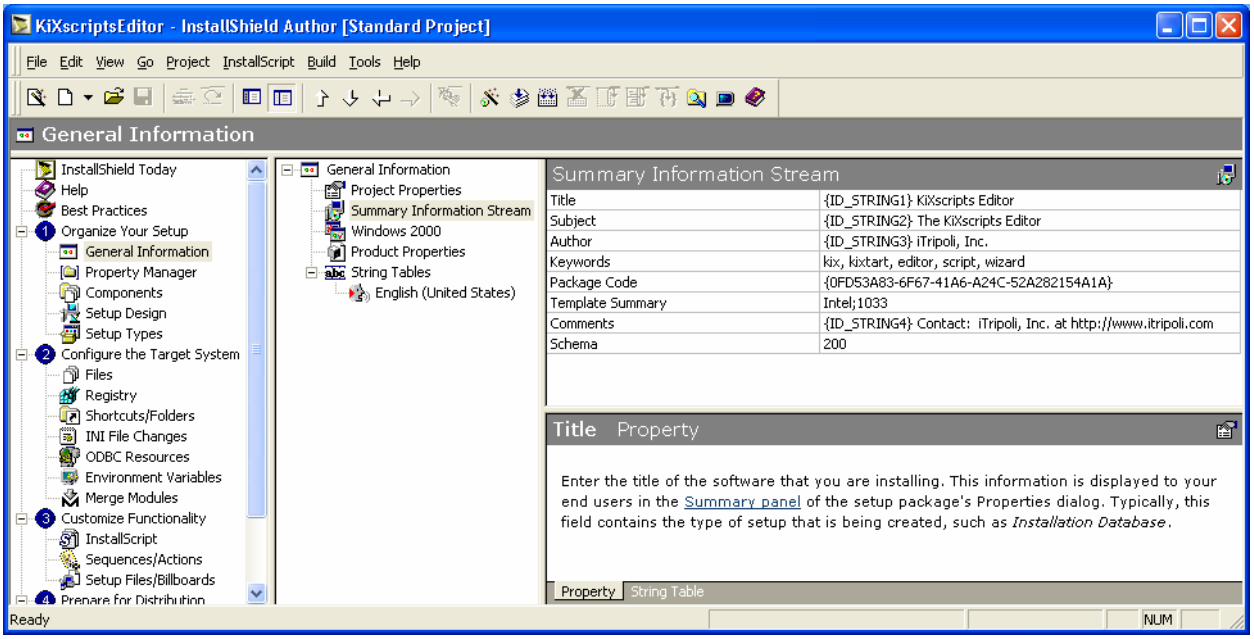

Figure 4.6: InstallShield AdminStudio's package editor, InstallShield Developer 8.

The following list offers some of the features provided by AdminStudio 5.0:

- Allows you to catalog all of your application deployment packages into a single enterprise conflict solver so that you can check your applications for a vast range of potential conflicts before you deploy them into your production environment
- Includes the full capabilities of InstallShield Developer—AdminStudio includes Developer with a selectable interface specifically tailored to systems administrators

- A process-oriented interface allows your team to step through projects, visually tracking what's been done and which tasks remain
- Enables distribution of Windows Installer packages to network and FTP locations, as administrative installations, or using Marimba channel publish technology
- Features its proprietary InstallMonitor technology, which aims to eliminate the need for snapshots by monitoring and recording all changes made during the installation

New Boundary Technologies' Prism Pack

Previously known as Picture Taker, New Boundary Technologies' Prism Pack is a packaging tool geared toward administrators that has also implemented recent updates to include support for the MSI format. Expanding upon this new functionality, a proprietary transform (MST) capability is available as well.

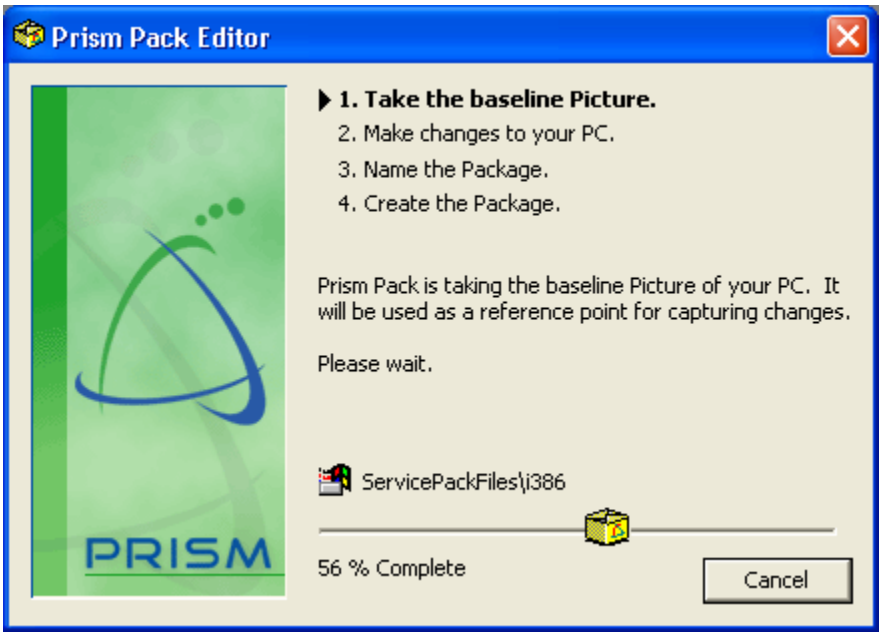

Figure 4.7: Prism Pack snapshot process view.

In addition to the functionality listed earlier, the following features help Prism Pack 5.0 stand out among similar tools:

- Interface is designed to simplify package customization for deployment of a single package for a diverse environment
- Inclusion of InstallShield Developer and Tuner products offer you full editing, customization, and MST support for Windows Installer files
- Smart Variables feature lets you turn a hard-coded value in a package into a variable that resolves itself as you tell it—prompt a user or pull a unique setting from each user's environment or from another variable

- Prism Conflict Checker helps to prevent errors before they occur by evaluating multiple deployment packages against one another for conflicts in the system and program files and the registry prior to deployment
- When it fixes software, it only reinstalls missing or broken files or components making the process faster and more bandwidth efficient than MSI's method of reinstalling at the feature level

Package Testing

It is obviously important to check your package before hitting users with it. Even the most experienced desktop administrators realize there are problems that occur, and it is best to take care of them before anyone has to know about it!

Package Quality Assurance

To ensure that package is ready for deployment, run your package from the command line. When it is complete, reboot if the installation will reboot during distribution but do not reboot if the distribution process will not. Now, log off and log on again as a user with standard access to the system (not an administrator account).

Walk through the procedures for testing the application you drafted when you had the user walking through the installation with you. If possible, have your test user return and validate your package by using the software and verifying any desired optional features are present.

For Windows Installer setups, a good test of your package is to prove that a user can successfully cause MSI to restore missing files and execute properly in the context of an account without administrative access to the computer. The following command line will let you advertise the setup to your workstation while logged on as a regular user:

```
runas /user:seti-1\bkelly "msiexec /j z:\mysetup.msi /qb"
```

Advertising only installs any MSI shortcuts; the files will be seen as missing and installed on first use. Double-click the shortcut to trigger an installation and launch of the application. To account for spaces in a path or file name, you might need to precede the double quotes within the command line with backslashes (\) as in the following example:

```
runas /user:seti-1\bkelly "msiexec /j \"z:\my setup.msi\" /qb"
```

Deployment Testing

Just because you successfully run a package from the command line does not mean that you will have the same results when running it via your deployment system, as Figure 4.8 illustrates. File paths are often the cause for problems in deployment testing. If there are spaces in the path to the setup file (or any other file or path you must specify), make certain that you include quotes around them. When you add the possibility of switches and switches that include paths with quotes, your command line can get a bit ugly and require some playing around to figure out the correct placement.

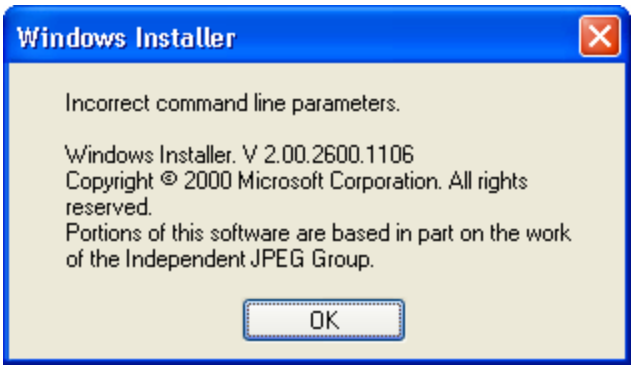

Figure 4.8: When an invalid command line is passed to the Windows Installer, the system presents this dialog box.

Pilot Deployment

Before you send your package to hundreds or thousands of workstations, a period of real-world use is instrumental in a successful rollout. Many organizations make the mistake of allowing their network administrators to serve as pilot users. As they likely have administrative control of their systems, this test does not give you a good picture of how normal users will be affected—the main goal of a pilot deployment. Even if the administrators primarily use accounts with normal user access, the fact that they have administrative accounts might still hinder proper testing. For example, if an installation performs actions at next logon, they might be unsuccessful for a user with normal access. Even if administrators don't use their administrator accounts to test the package, the simple act of logging on as an administrator could hide a potentially big problem.

Volunteers Anyone?

If you ask for volunteers from different business functions within your organization, you will get better coverage of different user habits than if administrators perform this task. The idea of getting new software before everyone else is enticing to some users, who might happily volunteer to take part in your pilot software deployments. Ensure that the users taking part in the pilot testing understand the risks involved. Having users sign an agreement to acknowledge the understanding of these risks is an action taken by many organizations to alleviate potential problems down the road.

Collecting and acting on pilot results is crucial to the package testing process. Often a fixed group of users receive all packages and simply notify desktop administrators if a problem is encountered. Although better than nothing, relying on reports of problems alone is not an efficient or effective method of determining success. Ask specific questions of your pilot users and, to keep it from becoming routine, change your questions regularly.

Normally communicated via email, some kind of tracking method should be in place for future reference. Store responses in a separate mail folder, and for responses of significance, use the Save As option of your email software to store a text version of the email in the package folder structure (see Figure 4.1 earlier in this chapter).

A next step in using email tracking for pilot results is to make use of Outlook forms or similar functionality using other mail systems. More robust tracking systems include making use of databases and Web-based forms and reports. Whatever method you choose, keep retention of collected information in mind.

Full-Scale Deployment

Here we go—deployment time! Roll out the software in groups, over a couple of nights (or weeks if you have a very large network). Doing so will help minimize the impact of any problems you might have missed in your testing thus far.

User Notification

Some board or group of managers for the departments in your organization should be notified of your deployment schedule on a regular basis. Some organizations limit deployments to once per month or quarter or even bi-annually, but many are on an almost constant release schedule. Notify departments and users as appropriate to ensure that they have an opportunity to notify you of critical activities that might demand a modification of your deployment schedule. Some organizations post a Web page or spreadsheet of their deployment schedule for users to view as desired. Most organizations make use of internal email to send more focused announcements. Include in your communications to users information about what will change and which new features will be included in a deployment. If you're installing an update, explain what is new. If you're deploying a new application, provide information about why you're deploying it and how to use it. Make sure that users know what is coming—and providing them with a list of new features and benefits will even have them looking forward to the release.

Track Successes and Failures

Don't wait to count how many complaints there are to determine success. Most deployment suites provide some method of reporting to notify administrators of success or (if an error is encountered) failure. For failures, learn as much as you are able about what went wrong so that you can improve your process or take further parameters into consideration for your next deployment.

Management will want to know what the success rate was for the deployment. You can calculate this rate of success by dividing the number of successful installations by the number of target systems:

```
(Successful Installations / Total Target Systems) × 100 = Success Rate
      740 Successful installations
      932 Target systems
      79.3% Success Rate
```

To help with your success rate, establish what will be counted as a *deployable system*. If a machine is powered off or no longer exists, these machines should be included in the calculation you present to management:

```
[Successful Installations / (Total Target Systems - Powered Off or
Invalid Systems)] × 100 = Success Rate
      740 Successful installations
      932 Target systems
      128 Machines powered down
      4 Machines no longer exist (or are invalid entries)
      92.5% Success Rate
```

Baseline Updates

To maintain the benefits of a consistent baseline, as updates are introduced to your network, a critical task is to ensure that new and rebuilt systems are also up to date. You have a couple of options available to help ensure this capability.

Image Updates

Using your deployment packages to update your baseline image is one reliable method used for ensuring consistency between those systems updated with packages and those with a new image applied. Another common method is to make use of the installation documentation created while developing packages to help ensure that the images contain the same configuration and settings as those systems receiving distribution packages.

Unattended Installation Script Updates

If your installation process is script based, be sure to replace old installation scripts with those of your more recently deployed installation scripts. Because this task requires little effort compared with that required to update an image, you might choose to implement the new installation scripts at the time they are deployed. In this way, you will not have to be concerned with updating new machines via your software deployment system.

> ☞ INOSOFT Garibaldi is a tool to help manage images, installation source files, and scripts. It helps generate and track versions of installation scripts and facilitate adding your scripts to existing systems and to your baseline using an unattended installation process. For more information about Garibaldi, visit http://www.inosoft.de/garibaldi/Default.asp?Sp=en.

Package Availability

Don't keep your packages a secret. Particularly in large environments in which several offices might have a requirement for applications with similar functionality, seeing that a package has already been generated and deployed may be a determining factor in the software they select. Where there is a significant backlog of applications waiting to make their way through your package and deployment process, helping the user to select an application you have already deployed will save the user time waiting for package development and will reduce the number of applications you need to manage.

Package Deployment System

With your package in hand, how do you intend to get it out there? Chances are, unless you are establishing a new network or significantly increasing the size of a small network, you probably have an existing deployment system. Even if you do, this section will still be of interest to help you get an idea of functionality that you might not have and will want to request of your deployment system vendor.

It is beyond the scope of this book to provide sufficient detail about every deployment system available. So rather than list just a few of the most popular deployment systems and only a brief summary of the functionality they provide, Table 4.2 provides a list of many of the solutions available to help those who need one begin the search for the right tool.

What is the right tool? Every organization will have its own set of requirements. It is easy to get lost in the sea of features made available by these systems. Rather than focusing on any one feature, try instead to create a list of features in which you are interested. You can then be more methodical in your approach to finding the best tool for your organization. Besides, there is no sense in paying for features you do not need.

Software Distribution Vendors

A large number of companies have taken a place in the market to provide a tool to address your software deployment needs. Several companies means several different takes on the subject, with each providing its own perspective—resulting in a unique interface and feature set to address a wide range of issues common to many organizations. Table 4.2 describes a few of your options.

Company	URL	Description
Altiris	http://www.altiris.com	Altiris provides several desktop administration tools in its Client Management Suite including Deployment Solution, Application Management Solution, and Software Delivery Solution. Its Web-based interface is modular in nature, providing several capabilities that can be added to satisfy a wide range of issues. Also part of the Client Management Suite: Inventory Solution, Application Metering Solution, and Carbon Copy Solution (remote control).
ASDIS Software	http://www.asdis.com	ADSIS Software provides a solution of the same name, which serves to provide central management of application servers, automate installation of new systems, and manage changes to computers throughout a network. By their own metrics, ASDIS implemented more than 15 million software changes on 36,000 systems in the year 2000.
Baramundi Software	http://www.baramundi.de	Baramundi provides OS-Deploy, App-Deploy, and an enterprise management framework, as well as inventory and other add-ons. As the names imply, the system is capable of remotely deploying unattended installation scripts for Windows OSs as well as automated software deployments.

Boss	http://www.boss-uwin.com	Boss offers U-Win, a complete system for workstation inventory, Windows OS migration, application deployment, and the migration of user data and settings.
Broccoli Software	http://www.broccolisoftware.com	ADS is a management tool for deployment of Windows software applications. ADS restricts the availability of software based on user groups as well as informs users of new and updated applications and patches as they become available. In addition to deploying software, you can also use it to deploy security updates, service packs, and drivers.
CenterRun	http://www.centerrun.com	CenterRun executes all application changes from a central console, simulating changes prior to making them, and maintaining a complete audit history. CenterRun tells you who installs what where. It also compares the actual state of an application with what it should be as recorded in a central repository. This feature captures out-of-band changes to the application environment.
Cognet	http://www.cognet.com	PowerQuest recently purchased Cognet and its Cognet Manager product. Cognet Manager provides a software deployment and configuration solution designed for ease of use. It provides “Plain English Policies” that can affect application configuration to provide each user with a tailored installation.
Computer Associates	http://www.ca.com	Computer Associates’ Unicenter Software Delivery product provides automated software delivery throughout heterogeneous environments with rollback and version control capabilities. It is also designed to support remote users, PDAs, and mobile devices.
Contrado	http://www.contrado.com	Contrado offers SystemBuilder, a solution for the creation, deployment, or migration of Windows systems. It aims to provide a single-step virtual process, eliminating the need for remote staging, hands on touches, and multiple deployment tools.
Executive Software	http://www.execsoft.com	Executive Software provides Sitekeeper as its tool to handle software installation, license tracking, and software and hardware inventory. Supporting all releases of Windows since Windows 95b, Sitekeeper is focused on making system management simple, with a very small learning curve and no training required.
Hewlett-Packard	http://h18013.www1.hp.com/products/servers/management/rdu/description.html	Geared toward server management, the Remote Deployment Utility (RDU) is an application that remotely deploys driver and management agent updates to network attached servers. You can operate RDU from an IT administrator’s workstation, where remote servers are selected for software updates.

IBM	http://www.unitysite.com	IBM's Unity provides a deployment solution for Win2K and Windows XP systems. It can roll out new OS deployments and allows for the creation of MSI deployment packages. It features a replication capability that allows for hierarchical distribution of software packages over a WAN and will automatically discover the fastest link to the package servers.
InfraDirect	http://www.infradirect.com	InfraDirect provides its solution, Xpursuit, to provide application installation and licensing from a single central management console. It features application conflict prevention and a lab environment in which applications may be analyzed before distribution. Client/server communications are MD5 encrypted and authenticated through LDAP standard directory structures for increased security.
Insystek	http://www.insystek.com	Insystek provides a couple of products for desktop administration including NSCM Control Center. This tool lets you inventory, manage services, install, upgrade, or remove applications all from a single console interface. Insystek also provides a similar product (built on the same technology) named SoftDist LE, which is designed for small to midsized Windows environments.
LAN Supervision	http://www.lsvi.com	LAN Supervision (LSVi) provides Change Management Facility (CMF), which offers software and configuration management features. It features self-repair, task scheduling, version control, and built-in rollback and workstation configuration capabilities. It requires no scripting and boasts scalability up to 40,000 clients. LSVi also offers OnDemand Software, a self-service deployment solution.
LANDesk Software	http://www.landesksoftware.com	LANDesk software offers its LANDesk Management Suite with OS deployment. It allows you to build and deploy images with its OS Deployment Wizard and is compatible with leading third-party imaging solutions as well. It also allows for targeted multicasting of software packages with scheduling capabilities. LANDesk also feature a Pre-execution Environment (PXE) server solution that does not require a PXE server for each subnet.
ManageSoft	http://www.managesoft.com	ManageSoft provides a software management solution for desktops, servers, and mobile devices. It features software deployment, asset tracking, license management, Windows migration, and mobile/remote management capabilities.

Marimba	http://www.marimba.com	Marimba provides separate management solutions for desktop and mobile management, server management, and embedded management. The desktop/mobile management solution offers software distribution, inventory management, subscription management, and modules for migration and infrastructure.
Microsoft	http://www.microsoft.com/management	Microsoft provides several solutions for management of Windows systems including Systems Management Server (SMS), IntelliMirror (Group Policy), Remote Installation Services (RIS), and Software Update Services (SUS). Microsoft SMS and IntelliMirror make up a large percentage of the management solutions in use today. Both are powerful and regularly enhanced. As a result of the popularity of these solutions, many companies feature interoperability through sharing of data or by developing add-in products to enhance the features provided out of the box.
Mobile Automation	http://www.mobileautomation.com	Mobile Automation's Mobile Lifecycle Management Suite aims to provide an all-in-one solution for discovering, securing, managing, maintaining, supporting, and migrating all desktops, laptops, servers, and handheld devices from any location via a single management console.
NetSupport Solutions	http://www.netsupport-solutions.com	NetSupport Solutions offers NetInstall, a software packaging, distribution, and installation system that also provides creation, distribution, and execution of configuration scripts. There are two editions of NetInstall, a Standard Edition (SE) and an Enterprise Edition (EE), which includes additional features for use in large and distributed network environments.
New Boundary Technologies	http://www.newboundary.com	New Boundary Technologies provides its Prism Deploy software for the installation and updating of software on local and mobile clients and servers. It provides automatic targeting of computers based on hardware, software, or network configuration without a prior inventory scan.
Novadigm	http://www.novadigm.com	Radia Software Manager is Novadigm's change and configuration management tool that allows policy-based control of deployments, updates, repairs, and removals. Other tools in the Radia family include its Software Manager (Web-based user installations), Inventory Manager (hardware and software), OS Manager (imaging and settings migration), and Desktop Manager (change and configuration management).

Novell	http://www.novell.com/	Novell offers ZENworks for Desktops, which runs from Novell or Windows servers and requires no desktop client for management of Windows 98 and later systems. It provides application deployment, policy management, software and hardware inventory, desktop imaging and remote management.
On Technology	http://www.on.com	ON Command SiteManager is provided by On Technology as a desktop management system optimized for smaller organizations. It provides a single console interface for deploying OSs, installing applications, updating software, and performing disaster recovery and Windows migrations. For larger networks, ON Command CCM offers an open and scalable system providing much the same functionality.
OnDemand Software	http://www.ondemandsoftware.com	Often thought of as only a repackaging tool, OnDemand Software's WinINSTALL features automated software distribution to desktop and mobile clients. It creates packages and allows for deployment of Windows OSs and applications.
Outerbounds Technologies	http://www.outerbounds.com	Outerbounds Technologies offers PC lifecycle management through its Empirum software. It features automated Windows migrations (scripted, not imaging), hardware and software inventory, disaster recovery and backup, and electronic software distribution.
PatchLink	http://www.patchlink.com	PatchLink provides PatchLink Update, a cross-platform patch discovery and distribution utility that provides patch, software, data, and task deployment across the Internet. It automatically detects patch-related security vulnerabilities on client systems and provides the ability to correct them across all platforms and enterprise boundaries.
Quest Software	http://www.quest.com	DeployDirector from Quest Software allows you to deploy, rollback, update, and manage your client-side Java applications, accelerating release cycles and reducing time-to-repair.
ScriptLogic	http://www.scriptlogic.com	A graphical network administration tool, ScriptLogic includes features to support enterprise software deployment. Although it does not include a packaging component, the tool can be used to facilitate the deployment of existing packages (for example, MSI files) such as those containing applications, service packs, and so on.
Snow Software	http://www.snow.no	Snow Software offers its Snow Distribution client management tool for the automatic distribution of applications, updates, and OS changes. It provides centralized Start menu and shortcut management as well as its own scripting language.

St. Bernard Software	http://www.stbernard.com	Focusing on the tracking and deployment of service packs, hotfixes, and other patches, St. Bernard Software's UpdateEXPERT assesses client systems for missing patches based on the company's database of available fixes. You can research available fixes, scan your systems for them, then deploy updates to those machines you wish to update.
Vector Networks	http://www.vector-networks.com	Vector Networks provides centralized desktop management via its PC-Duo Enterprise suite. Its modular architecture gives you the ability to purchase just the modules you require, including Inventory Management, Software Distribution, Software Metering, Help Desk Issue Tracking, Diagnostics and Remote Control.
XcelleNet	http://www.xcellenet.com	XcelleNet Afaria focuses on systems management for mobile devices. It provides the capability to deploy applications and content, automatically back up data, and track hardware and software information.
Xyro B4 Software	http://www.xyro.com	Xyro B4 provides its Active Distribution Console (ADC) for the one-to-many distribution of software or even data files via IP multicasts. It is also able to execute remote commands on several machines simultaneously allowing for its software installation capabilities.

Table 5.2: Software deployment solution vendors.

These and many more products are detailed on AppDeploy.com at http://www.appdeploy.com/tools.

Summary

Each environment must have its own deployment process, and as the desktop administrator, you'll most likely be tasked with the creation of this process. In this chapter, we discussed how determine with application deployment method will work best for your organization as well as explored some of the tools available to help you do so.

In the next chapter, we will turn our focus to security, which has become a hot topic in the past few years. We will discuss security policy, tools, things to watch out for, and the benefits and drawbacks of implementing a locked down environment.

Chapter 5: Desktop Security

In the last chapter, we covered the packaging and deployment of new software and updates. With your baseline established, user environment customized, and applications and updates rolling, we will address the security options available for maintaining the control and configuration of your network. However, it goes without saying that it is best to have your security policy and enforcement methods well established before the first user sits in front of a computer. In this chapter, we will discuss security policy, tools, things to watch out for, and the benefits and drawbacks of implementing a locked down environment.

Security is a topic that has been steadily moving into the spotlight for the past several years. Viruses, hackers, data theft, and piracy are on the minds of most organizations today. Security is also a very broad topic. As such, we will focus on those issues that cause the most concern to you, the desktop administrator. The first of which is the establishment of a security policy.

Written Security Policy

A written security policy is the documentation used to define the rules for the use of computer systems in your organization. Without established, clearly written, and readily available documentation, ignorance will prevail as the reason for all security violations. A written security policy is often overlooked until the damage has been done. To protect your organization from data loss through misuse or from a lawsuit as the result of license violations or administrative action, it is important that a written security policy be defined and enforced.

Keeping in mind that desktop security is only a part of an organization's overall security policy and that the level of restrictions imposed is often based on a user's job, it is unlikely that you should be drafting this policy yourself. There should be several people from management in various departments involved in the creation and review of your organization's security policy.

Defining Your Security Policy

We will discuss the benefits and drawbacks that come with written security restrictions later in this chapter. For now, I'll summarize considerations to keep in mind when establishing this documentation.

- Keep it realistic—In some government networks or other environments in which data is very sensitive, extreme security measures are understandably a necessity. However, too strong a security policy can impede users' effectiveness and might add complexity to desktop administration. This management concept is called *convergence*, when multiple management goals, such as productivity and security begin to conflict. You will need to determine what is important to your business, then use that definition as your middle ground. When it comes to security, it is important to find a middle ground from which you can protect your data and systems without putting an unnecessary burden on those forced to abide by these rules.

> ✎ Are the rules being laid down enforceable? We will discuss many of the tools available to enforce your security policy. However, you must also keep in mind that enforcement and backing by management are vital to a security policy that is of any value. There must be a means of ensuring that the policy is followed and consequences result if they are not.

- Keep it simple—Can those who must read and abide by the security policy understand it? As a technical person, it is easy to assume an unrealistic level of technical knowledge and to use terms that not everyone will understand. Everyone must understand the security policy, and although it is painful to dumb things down to the lowest common denominator, in the case of a document such as this, it is necessary to do so.
- Keep it available—To establish a policy that people do not read defeats the purpose of creating one in the first place. Provide employees with a copy and have them sign a statement that they have read and understand it. Provide an electronic copy of the policy on your network for easy reference—a link from an intranet home page is ideal.

Enforcing Your Security Policy

With your realistic, simple security policy as common knowledge, ensure that all users know how you will be aware of violations and what you will do about them. It is difficult to enforce a rule to which there is no consequence for a failure to comply.

- Employing restrictions—A majority of this chapter will focus on the technologies and tools available to enforce your security policy. If users aren't allowed to introduce media to the network, restrict their ability to do so. If users aren't allowed to install unauthorized software, restrict their ability to do so. And if users are not allowed to save data to their local systems, restrict their ability to do so. See a trend here?
- Monitoring violations—With the right tools and a little persistence, most any restriction imposed can be bypassed. Computer cases can be opened, users might bring in Plug and Play—PnP—storage devices, and users might attempt to guess passwords or access data to which they are not allowed. When you understand the weaknesses in your ability to restrict such behavior, you've reached step two of security policy enforcement—monitoring.
- Imposing penalties—Odds are that you will be the one to point the finger, but not the one to punish offenders. It is, however, important for everyone to understand the consequences of security policy violation. The sensitivity and attention given to security in your organization will dictate the severity of these consequences.

> Get it in writing! Just as if an employee were regularly late for work, if a user has repeatedly violated security policy and a move is made to penalize the user, documented reprimands are imperative. An employee might choose to fight whatever penalty is being imposed, making it difficult to enforce without proper documentation. When a user violates security, the user should be made to sign and date a document clearly stating the violation in order to confirm that the user understands what he or she has done and to establish a trend in the event that repeated violations should arise.

Benefits of a Locked Down Environment

Each release of Windows provides more and more restrictive default security settings. The reason is partly a result of the increasingly common practice of locking down workstations to the greatest degree possible. One major reason for this practice is because, in business, users need to do their work, get their job done, and that is all. Allowing them to make system-level changes increases the likelihood of configuration problems, which leads to increased Help desk calls and user downtime. If users are permitted to install software that might be damaging or violate licensing agreements, the results could be costly. In recent releases of Windows, you can even restrict specified users from running locally installed software (for more information, see the sidebar "Software Restriction Policies"). Finally, in regards to user data, limiting where users can store data will help you to force data storage to a managed area for which fault tolerance and regular backups are available. In this section, we will discuss some of the key benefits to implementing a locked down desktop environment.

Software Restriction Policies

Via Group Policy, Software Restriction Policy (SRP) provides the ability to limit which applications can be launched on a system. It is the intention of SRP to confront the problem of regulating unknown or untrusted code. It does so by allowing administrators to identify and specify which applications are allowed to run. The applications can be identified in policy through a hash rule, a certificate rule, a path rule, or an Internet zone rule. The software can run on two levels: unrestricted and disallowed.

Before we go further, let me point out that the use of SRP is restricted to Windows XP and Windows 2003 systems. This capability is very useful, but its minimum OS requirements will keep this feature in the future for many environments. For those of you lucky enough to be in such an environment (the rest of us can use this information as ammunition to push for an upgrade), we will cover this new feature in more detail.

A rule is used to identify one or more software applications and specify whether the application(s) is allowed to run. There are four rules that can be used to identify software:

- Hash rule—A hash rule is a cryptographic fingerprint that uniquely identifies a file regardless of where it is accessed or what it is named.
- Certificate rule—A certificate rule specifies a code-signing, software publisher certificate.
- Path rule—A path rule can specify a folder or fully qualified path to a program. When a path rule specifies a folder, it matches any program contained in that folder and any programs contained in subfolders. Both local and UNC paths are supported.
- Zone rule—A zone rule can identify software from the Internet Explorer (IE) zone from which it is downloaded. This rule applies only to Windows Installer (MSI) packages. It does not apply to software downloaded in IE.

For more information about SRP, read the Microsoft article "Using Software Restriction Policies to Protect Against Unauthorized Software."

Provide Desktop Damage Control

In most environments, allowing users full access to their systems is a practice that results in an increased number of Help desk calls, license violations, improper use, and data loss. These results might be accidental or deliberate, but in a locked down environment, users are restricted from either case:

- Accidental—Some people just don't know better—or even worse, they *think* they do but they don't. It is these users, that know just enough to do some real damage, that should be your biggest concern. They may mean well—after all, they have a computer at home they tinker with all the time (to which you end up answering endless questions about as well). Restricting users from doing accidental damage to their systems is one of the most common reasons for implementing desktop security restrictions.
- Deliberate—You might puzzle as to why someone would deliberately do damage to his or her system. Hopefully, this damage can be largely attributed to "trying things out" without much care for the consequences. Damage that might be considered deliberate includes conscious policy violations such as introducing personal disks to the system.

Prevent Installation of Personal Software

It is natural for users to desire use of software they are familiar with; this often means personal software from home or from another network on which they do business. If the software they would like to have is not available to them on your network, they might attempt to use their own copy. Restricting permissions that result in an inability to install software and restricting access to removable media devices will go a long way to ensuring that this kind of license violation does not occur on your network.

Eliminate Unauthorized Installation of Company Software

Most organizations store copies of software on the network for ease of installation and to ensure a resilient source for Windows Installer setups. If this share is visible and accessible to unauthorized users, they might ignorantly think that it is okay to install. At the very least, they do not have to obtain the software or get it on the network—it is right there calling to them. Keeping installation media locked away in hidden shares or in locations restricted to those with access will greatly reduce the violations that come with unknown installations of software to which you have a limited number of licenses.

> Particularly when users have direct access to the Internet, restricting their ability to install software is a critical first step in maintaining control of your systems.

Stop Software Theft

Does your organization have a network share that contains setup files? Is there a software cabinet to which users have access? Although most users are not likely to steal company-owned software, a much greater number do not have much problem borrowing software to install at home. They might install it on their personal computer and return the software or (even worse) make a copy of the software to bring home and possibly share with others. Restricting access to installation media and removable storage will make such actions more difficult. With any luck, the added effort required to "borrow" software will further remind users that they should not be doing so in the first place.

Enforce Proper Usage

An organization's computers and networks are business assets available to conduct business. Playing games and conducting other personal business on these assets and on company time is usually more than discouraged—it might even be illegal:

Playing Games "On the Clock"

It might seem obvious that if your boss catches you playing Solitaire, it would probably be frowned upon. However, if it was installed as part of the workstation baseline, frowned upon may be as far as it would go. However, if you get caught playing a first person shooter game you bought at the store last week, you can probably expect to be in a bit more trouble. Again, this all comes back to your security policy, but disallowing use of unauthorized software is among the most obvious such policies. Again, the most direct method of restricting such activity is limitations on system access and the use of removable media. Taking it a step further, software inventory or metering software can also play a role in identifying and taking action on such violations.

Conducting Personal Business

Most organizations do not look kindly on employees making fliers for next weekend's cookout, writing long letters to Grandma, or bidding on Star Trek memorabilia on eBay. Although restrictions on such behavior are commonly enforced through casual observation, restricting access to certain Web sites (including Web mail services) is not uncommon.

Conducting Illegal Activities

When you think of illegal activities, you're probably thinking pornography—and you are right. Unfortunately, you hear about this kind of thing way too often and almost everyone knows a story of some kind relating to someone downloading porn at work. Although it is almost a given that you would be deleting the users account soon after getting caught, the organization itself might be liable as well, making this matter very serious. In addition, illegal MP3 files and online gambling services are possible causes for concern in the workplace. Software metering, software inventory, and periodic system scans are common practices to identify such problems.

Decrease Proliferation of Viruses

Though most environments employ a written policy restricting the use of removable media (floppy disks, CD-ROMs, DVDs, zip disks, and other magnetic and optical formats) it is one of the most casually violated security policies. In addition to the options available in restricting access to such devices, a locked down system will limit this kind of proliferation (and in some cases the damage) that a virus will be able to accomplish. Of course, this problem does not end here—you must also remember to address the Internet and email attachments as common methods for viruses to infiltrate your network.

Principal of Least Privilege

As an administrator (or a user with permissions), a virus has an easier time damaging systems in the user context in which it is executed, both locally and on remote networked systems. An administrator who uses their admin account to process email and download software provides a much bigger risk to the network than that of someone with non-admin access. For this reason, it is common for administrators to have two accounts—the admin account should be used strictly for administration, while their non-admin account can be used for day-to-day work such as reading and responding to email and browsing the Web.

Protect Data

When users store data on their local systems, it is often not managed and is therefore more likely to be lost. Network shares may be hosted on fault-tolerant drive arrays (protecting it from hardware failure) and backed up on a scheduled basis (protecting it from corruption or deletion). In addition to ensuring a desirable default location for applications to save their data, restricting a users' ability to save data locally will further direct users to a managed location.

Drawbacks of a Locked Down Environment

As with so many things, although there are many benefits to be gained from a locked down environment, there are also drawbacks. Locking down a desktop environment can result in an environment in which a user cannot operate needed software or perform business-critical tasks. Locked down environments make software deployment and automation more difficult and in the end requires additional (and more thorough) testing to be performed. In this section, we will discuss some of the drawbacks that weigh in the decision to implement a locked down desktop environment.

Application Failures

As a result of the increased security restrictions that come with Win2K and Windows XP right out of the box, software developers are more conscious of performing actions without regard for user access. Applications that write to the HKEY_LOCAL_MACHINE area of a system (outside the actual installation of the application) are one example of a problem that you are far less likely to encounter today. However, there are still many applications that attempt to create and modify files outside of the user profile, where users may have restricted access. An application dependant upon writing to a user restricted area will often result in some sort of error or unexpected behavior, as Figure 5.1 shows.

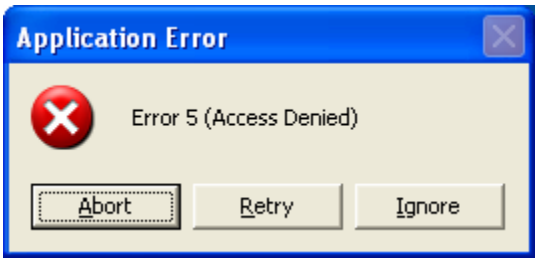

Figure 5.1: A typical error dialog box seen as a result of restrictive access control list settings.

Failure to Save Settings

Some applications attempt to save settings made by a user to a data file instead of the preferred HKEY_CURRENT_USER hive of the registry. Particularly common in older applications, such a practice may result in an error when changing preferences or exiting the application (or at whatever point the application attempts to record these settings for future use). Quite often there will be no visible error—the settings will simply fail to be saved.

> ☞ I have encountered this failure to save settings several times, but I particularly recall this problem when I was deploying Microsoft FrontPage 98 in a locked down environment. The installation went fine and, not being a user of the application, I did not realize that the application was failing to save its settings. The silent failure to do so was the result of the users' inability to generate two INI files used to store the settings. The files were to be created by the application on first use, and were then to be updated with specified settings as required. Of course, I was following the advice of this book's previous chapter, and quickly discovered this problem in the application testing phase of the package development. The fix was to simply add these two files to the installation package and allow users Change access to the files as part of the deployment.

Failures When Saving Data

A surprising number of applications actually require a fixed local path to create or modify data files. A directory created right off the root of the drive or a subfolder of the program's installation directory are not uncommon "working" directories for many applications. The application may launch and function normally, but when actually working with it to modify or create data, failures may occur. Obviously if you are able to specify a working directory, you may specify an area to which users have necessary access. If the problem is a hard-coded default directory, the simple inability for users to save their work may force them to save to a proper location or it may cause them to loose their work. This situation again shows why actual application use (in the security context of a user and not as an administrator) is a critical step in the package development and deployment process. When a hard-coded working or data directory is required by an application, the decision to allow access to the directory may be one forced on you as a deployment requirement.

> 🕮 We will discuss methods for modifying access to files and directories later in this chapter.

First Launch

Although you might not know it until you see it fail, some applications perform actions that are restricted to users when the applications are launched for the first time. Some applications attempt to write data to the HKEY_LOCAL_MACHINE area of the registry. Others attempt to register DLL files when first launched or when their functionality is first needed. This should be performed as part of the installation process, but might be easily overlooked. Ensure that you log on and perform the initial execution of an application as a user immediately following deployment to determine whether this issue is one you will have to deal with.

Complex Deployments

Although restricting the environment in which users operate may mean fewer Help desk calls as a result of users' own actions, your own job of testing and deploying software packages becomes far more difficult as a result of a locked down environment:

- Restriction of actions automated for users—Out of the box, the logon script is executed in the security context of the user logging onto the computer. Although you might want to automate printer installation, software configurations, or other restricted actions, such actions may be just that—restricted. Microsoft and third-party utilities and command-line tools exist to trick or even replace this functionality; most environments are limited in their ability to perform actions during user logon as a result of the fact that systems must operate in the security context of the user.
- Bad package or bad application? The failure of an application to run may be the result of a bad deployment package or the application's inability to operate in a locked down environment. The additional step of installing the application and operating it as a user with regular user access to the computer thus becomes a necessity. I have often seen desktop administrators spend hours trying to determine why their script or repackaged installation is failing to operate correctly only to find that even when installed manually, the same problem occurs.

Political Considerations

Many companies have gone down the locked-down path and found that it decreases productivity, impacts morale, and causes increased difficulties in desktop administration. The decision to lock down your environment (and to what extend) must be measured against your corporate culture and the sensitivity of the systems you are trying to protect. Simply approaching desktop administration from a technical perspective will often result in backlash when the cultural implications are not also taken into account.

Modifying File and Registry Access and User Rights

With a read-only registry and file system, you may exercise the strictest level of security. However, doing so will most certainly result in application and system failures aplenty. Still, the most methodical and thorough approach to the limitation of file and registry access is to lock everything away from the user, then open only what is deemed necessary on a case-by-case basis. As situations that require relaxed security of specific folders and registry keys increase over time, your once locked down system could be come a very open one. It is therefore important that the additional engineering and testing time be taken to identify, document, and adequately justify each security modification determined to be necessary. In this section, we will discuss some of the tools available for modifying registry security, file security, and user rights.

Registry Security

The registry holds settings for both the user (HKCU) and the computer (HKLM). These computer settings affect everyone that utilizes the system and are typically restricted to manipulation by those with normal "user" access to the system. The installation of most all software requires keys and values to be created in the HKLM hive of the registry. Thus, software installation can be fairly easily restricted simply by ensuring that users are not members of groups that grant them access (such as the Administrators or Power Users groups).

> ☞ In Win2K and later, only Administrators and Backup Operators have default network access to the registry. To restrict network access to the registry, create the HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurePipeServers\winreg key in the registry, and create a value named Registry Server with a type of REG_SZ. The Security permissions set on this key define which users or groups can connect to the system for remote registry access. For more details, see the Microsoft article "How to Restrict Access to the Registry from a Remote Computer."

Because some applications insist on writing to the HKLM area of the registry, you might need to modify permission to allow users write access to certain keys. In other cases, you might not want users to have the ability to make changes to keys that contain certain values. There are several ways to go about modifying registry security; we will cover some of these tools in the following sections.

Group Policy

In an AD environment in which client systems are running Win2K or later, the preferred way to go about enforcing access control list (ACL) changes to either the registry or the file system is to use Group Policy. You can use the following steps to create a Group Policy Object (GPO) that will modify the permissions of a registry subkey:

42. In a new or existing GPO, navigate to the Windows Settings/Security Settings/Registry folder under Computer Configuration in the tree view on the right (see Figure 5.2).
43. Right-click the Registry folder (or after selecting this folder, right-click in the pane at the right of the display), and choose Add Key.
44. A browse dialog box will appear entitled Select Registry Key. Browse to the key you want to modify or enter the key manually, and click OK to continue.
45. A standard security dialog box will appear in which you can add groups or individuals and choose their desired level of access. Click OK when your desired modifications are complete.
46. A dialog box will appear that lets you choose whether you would like to propagate (edit) the permissions to all subfolders and files or you would prefer to entirely replace the existing permissions. You may also choose not to affect the key changes whatsoever. Click OK to confirm your choice.

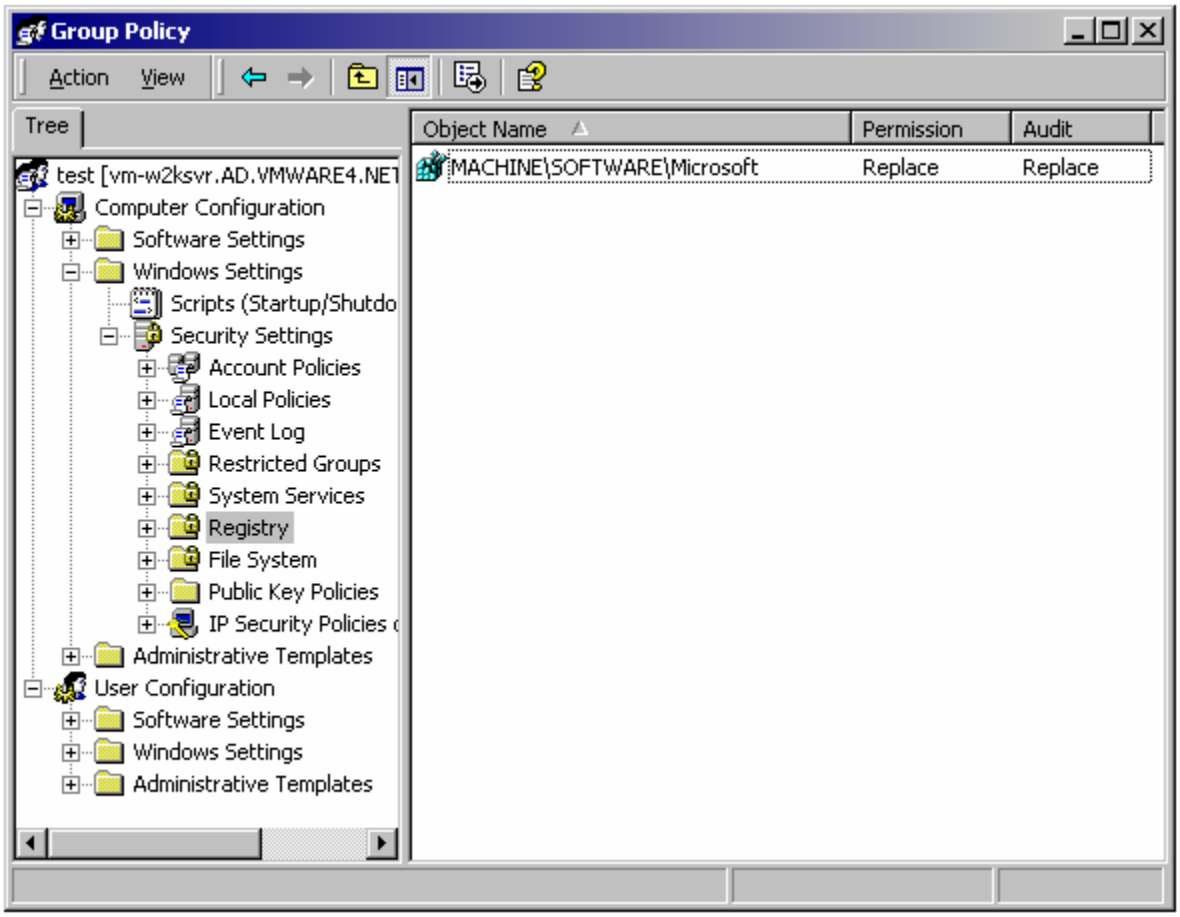

Figure 5.2: GPO with registry security modification.

Once a GPO such as this has been created, you can link it to any group of computers or organizational unit (OU) containing computers within AD. Many organizations choose to implement a single GPO that contains all ACL changes in their environment. However, where a GPO exists for a specific software distribution, it is common to include the ACL change in the same GPO as the software package. However you choose to implement ACL changes via Group Policy, aim to minimize the number of GPOs required to avoid delays during the startup process.

> ☞ You can use Group Policy to manage only Win2K and later clients in an AD environment. Even with AD services installed on earlier versions of Windows, Group Policy is not an option. You can enforce settings on these earlier versions using System Policy, but System Policy can be used only to enforce settings and not to manipulate registry or file security. For these systems, an alternative means of accomplishing this task must be utilized (typically using a command-line utility such as RegINI.)

RegINI

Available in NT and later resource kits, RegINI provides a means of manipulating registry values and, more importantly, security settings from the command line. The utility specifies an external file that acts as a script for setting and modifying registry entries, and optionally, their security. With its long list of security options and odd formatting, it is not the friendliest of utilities and can prove difficult to work with. Table 5.1 shows many of the permission identifiers that you can use for setting security with this tool.

Permission Identifier	Definition
1	Administrators Full Access
2	Administrators Read Access
3	Administrators Read and Write Access
4	Administrators Read, Write and Delete Access
5	Creator Full Access
6	Creator Read and Write Access
7	World Full Access
8	World Read Access
9	World Read and Write Access
10	World Read, Write and Delete Access
11	Power Users Full Access
12	Power Users Read and Write Access
13	Power Users Read, Write and Delete Access
14	System Operators Full Access
15	System Operators Read and Write Access
16	System Operators Read, Write and Delete Access
17	System Full Access
18	System Read and Write Access
19	System Read Access
20	Administrators Read, Write and Execute Access
21	Interactive User Full Access
22	Interactive User Read and Write Access
23	Interactive User Read, Write and Delete Access

Table 5.1: RegINI permission identifiers.

The following RegINI script file will change the permissions of the RunOnce key to allow everyone Full Access:

```
\Registry\Machine
      SOFTWARE
            Microsoft
                  Windows
                        CurrentVersion
                              RunOnce [7]
```

> Because security settings for a key affect its subkeys as well, you must have permission to change security information for the key and all its subkeys in order to change security information for the key.

RegDmp

RegDmp is a Windows resource kit utility that works in conjunction with RegINI. It dumps the contents of the registry to an output file in a RegINI-compatible format. Using RegDmp, you can create RegINI input scripts based on actual registry values. You can then modify the contents of RegDmp output to make any modifications you need to the registry.

SecAdd

SecAdd is a simple command-line tool that allows registry security modification from the command line. It is much easier to use than RegINI, but is more limited in its functionality. You can use the tool to remove the Everyone group from a specified registry key in the HKEY_LOCAL_MACHINE hive or to add read access to a specific user for a specified registry key.

File Security

There are times when you might want to restrict a user's ability to save files to a certain location or to allow access to a certain file or folder in order for an application to operate. When locking down a system, only the files and folders that are required to be modified by users provide such access.

As we have discussed, with everything locked away from the users, there are likely to be situations in which you need to relax certain permissions in order for applications to operate. There are several tools available for managing file security to help you in this task. In this section, we will discuss some of the methods and tools available for modifying file and folder permissions.

Group Policy

Using the same procedure specified earlier in this section for modifying registry security, you can use Group Policy to modify file security. Once again, this capability is restricted to AD environments in which client OSs are running Win2K or later.

CACLS

The native CACLS command allows ACL modification from the command line. CACLS supports the following command-line switches:

- /T—Changes ACLs of specified files in the current directory and all subdirectories.
- /E—Edits the ACL instead of replacing it.
- /C—Continues when an access denied error is encountered.
- /G user:perm—Grants specified users access rights; perm can be R (read), W (write), C (change), or F (full control).
- /R user—Revokes specified user's access rights (only valid with /E).
- /P user:perm—Replaces the specified user's access rights; perm can be R (read), W (write), C (change), or F (full control).
- /D user—Denies the specified user access.

You can use wildcards to specify more than one file in a command. You may also specify more than one user in a command.

XCACLS

The XCACLS tool provides further support for modifying ACLs from the command line. In addition to the switches provided by CACLS, XACLS also supports:

- /X—The same as /E except /X affects only the ACL, which the specified users own.
- /G user:perm;spec—Grants specified users access rights as CACLS does, but provides more options including the ability to modify the ACL for a directory; perm can be R (read), C (change), F (full control), P (change permissions), O (take ownership), X (execute—special access), E (read—special access), W (write—special access), and D (delete—special access.); spec has the same options as perm and will only be applied to a directory.
- /P user:perm;spec—Replaces specified user access rights; it accepts the same parameters as the /G switch.
- /Y—Replaces user rights without verification.

XCACLS is not a native Windows command, but you can download it at no cost from http://www.microsoft.com/windows2000/techinfo/reskit/tools/existing/xcacls-o.asp.

SuperCACLS

TrustedSystems.com's SuperCACLS is a third-party tool for command-line manipulation of ACL settings. It contains a suite of tools including PRACL (for printing ACLs in a simple format that you can also use as a backup and restore feature), REACL (for replacing ACLs with new ones), MODACL (for modifying, removing, or adding individual ACL entries, leaving other entries unchanged), and TAKEOWN (for taking ownership of items and optionally granting you full control).

📖 To learn more about SuperCACLS at http://www.trustedsystems.com/scacls.htm.

User Rights

User rights dictate which actions a user may perform on an NT and later system. There are several rights restricted by NT and later systems that you can use to limit what a user can do without elevated access to the system. Rights are normally assigned at the group level (as opposed to individual users) and can be assigned from a GUI or command-line interface with various tools made available by Microsoft and other third-party vendors.

From within NT, rights are set using the User Manager for Domains (at the domain level) and User Manager (at the local machine level) tools for defining which groups can perform actions that are restricted by the available user rights. In Win2K and later, there are additional rights that you can specify by using a Group Policy Object (GPO), which may be assigned to an OU or even to the entire domain. Finally, there are tools provided to allow for the command-line granting and revoking of user rights.

Group Policy

You can specify user rights within any GPO or using the Domain Security Policy snap-in that modifies the Default Domain Policy, which is a GPO assigned at the domain level. It is customary to place domain-wide security settings in this single policy, including the assignment of user rights.

NT User Manager

In NT, user rights are assigned using the User Manager and User Manager for Domains applets. To access user rights for a domain, launch User Manager for Domains. From the toolbar, click Policies, and choose User Rights. From the drop-down box, choose the right you want to grant or restrict (see Figure 5.3).

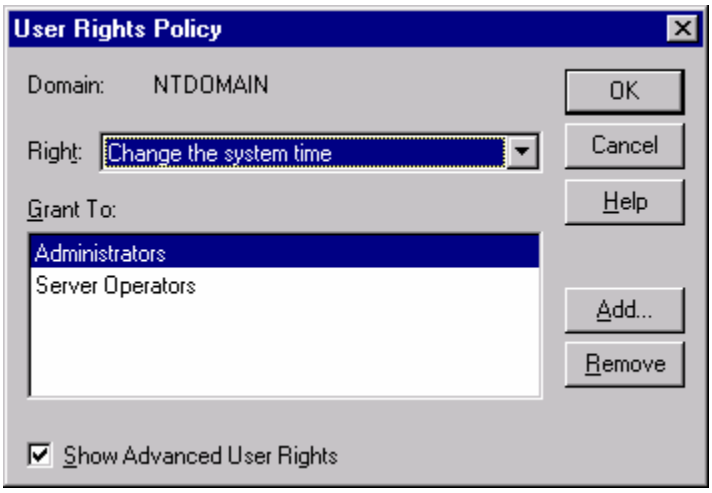

Figure 5.3: User rights manipulation in NT.

NTRights

You can use the NTRights utility to deny or grant user rights to users and groups from the command line. Table 5.2 defines these rights.

Right	Definition
SeAssignPrimaryTokenPrivilege	Replace a process-level token
SeAuditPrivilege	Generate security audits
SeBackupPrivilege	Back up files and directories
SeBatchLogonRight	Log on as a batch job
SeChangeNotifyPrivilege	Bypass traverse checking
SeCreatePagefilePrivilege	Create a pagefile
SeCreatePermanentPrivilege	Create permanent shared objects
SeCreateTokenPrivilege	Create a token object
SeDebugPrivilege	Debug programs
SeIncreaseBasePriorityPrivilege	Increase scheduling priority
SeIncreaseQuotaPrivilege	Increase quotas
SeInteractiveLogonRight	Log on locally
SeLoadDriverPrivilege	Load and unload device drivers
SeLockMemoryPrivilege	Lock pages in memory
SeMachineAccountPrivilege	Add workstations to domain
SeNetworkLogonRight	Access this computer from the network
SeProfileSingleProcessPrivilege	Profile single process
SeRemoteShutdownPrivilege	Force shutdown from a remote system
SeRestorePrivilege	Restore files and directories
SeSecurityPrivilege	Manage auditing and security log
SeServiceLogonRight	Log on as a service
SeShutdownPrivilege	Shut down the system
SeSystemEnvironmentPrivilege	Modify firmware environment values
SeSystemProfilePrivilege	Profile system performance
SeSystemtimePrivilege	Change the system time
SeTakeOwnershipPrivilege	Take ownership of files or other objects
SeTcbPrivilege	Act as part of the OS
SeUnsolicitedInputPrivilege	Read unsolicited input from a terminal device

Table 5.2: NTRights command-line rights.

☞ The NTRights.exe utility is included in the NT Server 4.0 Resource Kit, Supplement 3.

Changing Security Context

Although an administrative or service account may have desired rights and ACL accesses, you might not want to provide them to users. To make use of such access and rights for a specific task, you may choose to temporarily change to a user context that does.

Changing the security context of a user to that of one with greater access can be a security risk. Some tools require that a name and password be provided in plain text, which you will need to store in your script or batch file in order to automate such a task. Win2K introduces the RunAs utility that allows you to execute a process using alternative security credentials by specifying a name and password. RunAs does not accept a password as a parameter, which eliminates silent automation of this utility. There are, however, other tools available to help you accomplish the same results, as we will discuss.

The Switch User Utility (SU.EXE)

Earlier versions of SU required that the user have several rights that are not standard for regular users, including *Act as part of the operating system*, *Increase Quotas*, *Replace a process level token*, and *Restore files and directories*. In addition to having to supply a username and password in clear text, this utility was often considered to be an undesirable option. The current release utilizes a client-side service that eliminates the need for additional rights by users.

You can install the client-side service using the following command:

```
suss.exe -install
```

There are several optional command-line parameters that you can specify to dictate the user name, user account domain, and, of course, the command to be executed.

> 🕮 The SU utility is available in NT and later resource kits. For more information about the SU utility, see SU.TXT included with the SU utility in the Windows resource kit.

TqcRunas

Quimeras Software's TqcRunas is a third-party utility that accepts passwords directly in the command line. You can use this utility by creating executable modules or by using the scriptable COM objects provided by the TqcRunas.dll.

> 🕮 To learn more about TqcRunas, visit http://www.quimeras.com/TqcRunAs/tqcrunas.htm.

> ✎ In Win2K, Microsoft introduced the RunAs command, which allows a specific command to run in a specified user context by providing the name and password of the account to be used. Unfortunately, the command was specifically designed for manual use and not to be used in a script; the name and password cannot be passed as parameters.

Controlling Removable Media and Restricting Device Access

When Internet access is limited or unavailable, managing the security of peripheral devices can put up an effective roadblock against the introduction of unauthorized files. You can greatly minimize the introduction of viruses and unauthorized software by preventing their introduction to the network.

Restricting access to floppy and CD-ROM drives is a common practice to help prevent both the introduction and removal of data and software to and from the network. Depending upon how serious you are about this restriction, you may limit access using software controls or even go so far as removing the devise entirely.

Windows does offer policies that control access to floppies and CD-ROM drives over the network, but it does not yet provide a built-in ability to restrict access to removable media by the locally logged on user. However, there are resource kit and third-party utilities available to provide restricted access.

- Floplock—Floplock restricts access to floppy drives, allowing only those that are members of the Power Users or Administrators groups to utilize the drive. It is available in both the NT and Win2K resource kits.

> 🕮 To learn more about the Floplock utility, see the Microsoft article "How to Restrict Floppy Disk Drive Access Using Floplock Service."
>
> 🕮 Floplock is also available as part of Microsoft's Zero Administration Kit (ZAK) http://www.microsoft.com/ntworkstation/downloads/Recommended/Featured/NTZAK.asp.
>
> 🕮 Full source code for Floplock is available on the Microsoft Win32 Software Development Kit (SDK) compact disk, in the \q_a\sd_floppy directory.

- DeviceLock—SmartLine's DeviceLock provides control over which users can access specified devices (floppies, serial and parallel ports, magnetic and optical disks, CD-ROMs, USB drives, zip drives, and so on) on a local computer. To learn more about DeviceLock, visit http://www.protect-me.com/dl/.

Some manufacturers (Dell being the first major one) have already begun the move to cease shipping computers with floppy drives. However, CD-ROM burners are very common and provide a similar risk in the removal of software from a network.

Disable PnP and Other Devices

With floppies and CD-ROMs removed or restricted, do not overlook Windows PnP support. If a user sticks a USB memory device into an available USB port, the native support for such a device may even eliminate your ability to simply restrict driver installation.

Devices such as the mouse and keyboard are set to start at startup by default; however, you can edit the registry to modify this behavior. You can set the floppy and CD-ROM not to start in Windows by setting the Start values to 4 in both the Ftdisk and Cdrom keys (under HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services). With these devices set to not start, the floppy and CD-ROM drives will not appear to the user when in Windows (before Windows loads, they are still accessible).

> 🕮 Scripting will be covered with more detail in Chapter 7

The *Load and unload device drivers* Right

The *Load and unload device drivers* right (see Figure 5.4) allows users to install and uninstall PnP device drivers. This right does not affect the ability to install drivers that are not PnP—those may only be installed by Administrators.

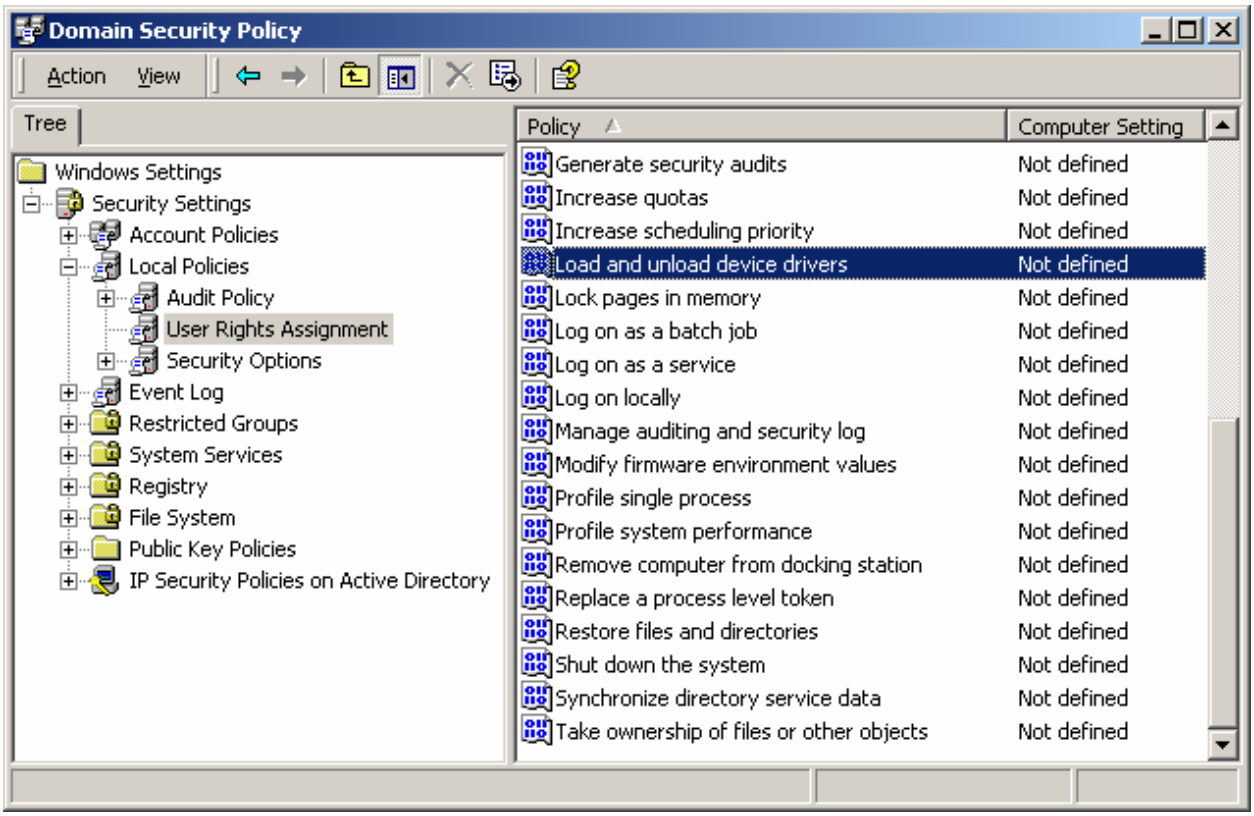

Figure 5.4: User right for loading and unloading drivers.

> Many PnP devices do not require administrative privileges to be installed.

SecureNT

SecureWave SecureNT provides the ability to control end-user access to I/O devices such as the floppy drive, memory sticks, PDAs, USB external storage, CD-ROM, serial and parallel ports, as well as many other PnP devices.

> To learn more about SecureNT, visit http://www.securewave.com/products/securent.

General Security Recommendations

Certain security settings will be more or less important to one organization than another. However, there are several security practices that make sense for all environments, as we'll explore in this section.

Rename Local Administrator Account

By renaming the local administrator account, you make it twice as hard for unauthorized individuals to utilize the account. With a user name and password being the key to full system access, having an administrator account named Administrator provides unauthorized users half of what they need to gain access.

Enforce Complex Passwords

NT account policies provide the ability to enforce passwords of a certain length, to enforce how often they must be changed, and to limit reuse of passwords through its User Manager tool (see Figure 5.5).

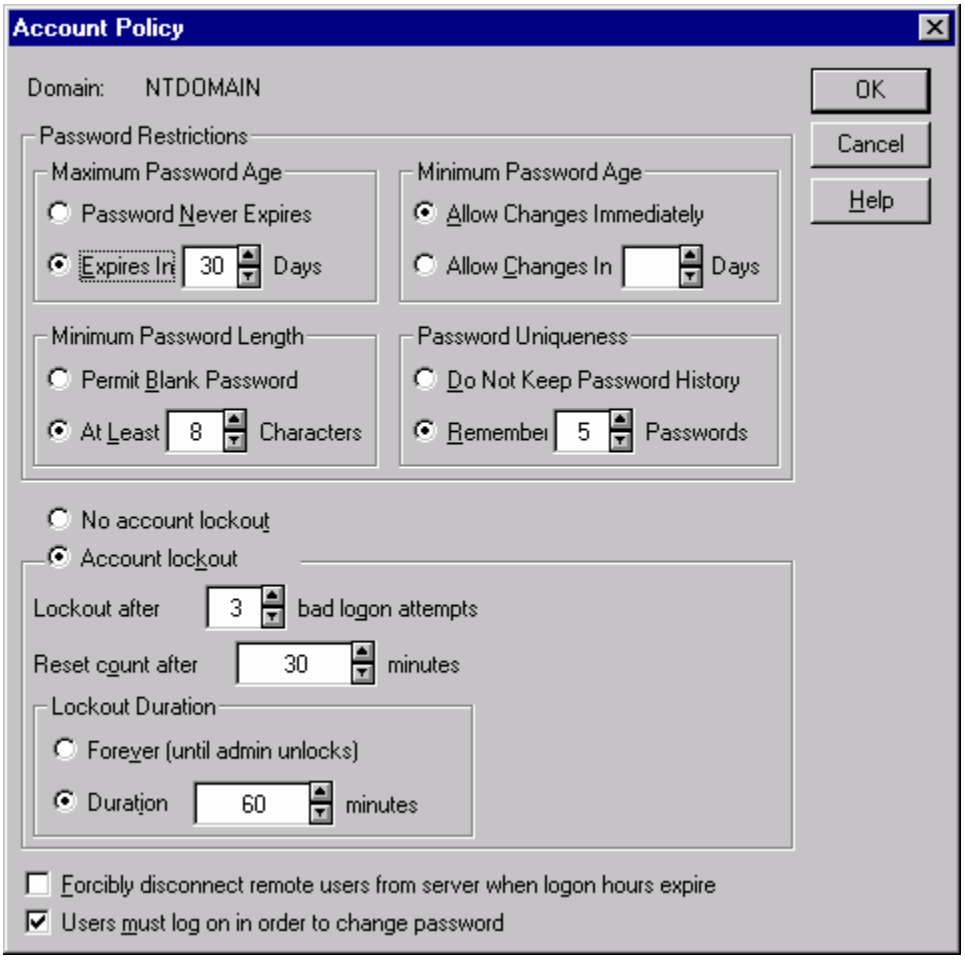

Figure 5.5: NT Account Policy dialog box.

Further, you can enforce complex passwords that must contain at least three of four character types: upper case, lower case, numbers, and special characters (symbols). In NT SP2, you can configure this requirement by performing the following steps:

47. Copy PASSFILT.DLL from SP2 to %WINDIR%\SYSTEM32.
48. Launch a registry editor, and create (or edit) the HKLM\SYSTEM\CurrentControlSet\Control\Lsa key.
49. Add a REG_MULTI_SZ value called Notification Packages with a value of PASSFILT (if the value FPNWCLNT already exists, then edit the value, and add PASSFILT under FPNWCLNT).
50. Restart the server.

For Win2K and later servers, this information may be set within a GPO, typically the Domain Security Policy GPO, which Figure 5.6 shows.

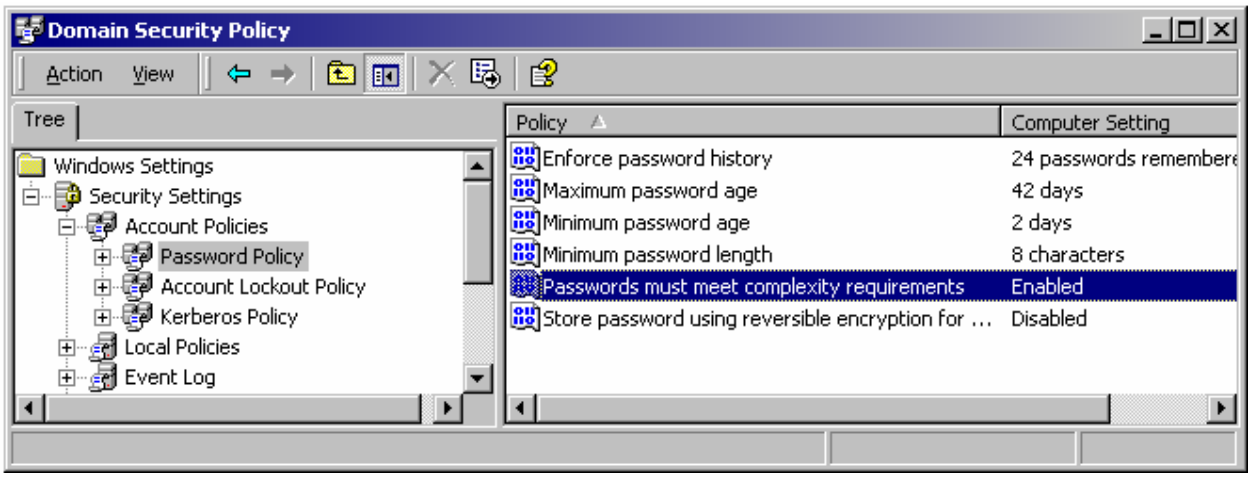

Figure 5.6: Domain Security Policy's password policy settings.

Disable Guest Account

On Win2K and later, the guest account is disabled by default. However, before rolling an image out across your network, it is wise to give this setting a quick check to ensure that the Guest account is disabled. With the guest account enabled, users may utilize the guest account to access resources without providing credentials of their own.

Replace the Everyone Group with Authenticated Users on File Shares

Authenticated Users was added with the release of NT SP3 and is included by default within Win2K. The Everyone group includes even unauthenticated users, whereas the Authenticated Users group includes only users who have supplied proper credentials. Wherever possible, remove permissions from the Everyone group and replace it with permissions for Authenticated Users.

Employ Logon Warning Messages

Security banners are presented when Ctrl+Alt+Delete is pressed, before the logon box is presented for entering a user name and password. It is customary to make use of this feature in order to provide a legal warning or notice of how the system may be used.

Figure 5.7: Example logon security banner.

Hide Last Logged On User Name

Without a policy in place to prevent it, the name of the last user who logged onto the computer is already typed into the Windows logon dialog box and only the password need be entered. Although this configuration is more convenient, it gives someone attempting to break into the system half of what he or she needs to gain access. You can disable this setting through the registry or using NT System Policy or Win2K Group Policy.

Password Protect the Screensaver

By enforcing a screensaver timeout period and specifying that the system should lock when the screensaver activates, you can minimize the possibility of someone sitting down at a machine and performing actions with another user's credentials. You can configure this setting through either System Policy or Group Policy, depending upon your environment.

Enable Auditing

The most basic form of intrusion detection for NT and later is to enable auditing. Doing so will alert you to changes in account policies, attempted password hacks, unauthorized file access, and other configurable items. Consider auditing the events that Table 5.3 shows.

Event	Setting
Account logon events	Success, failure
Account management	Success, failure
Logon events	Success, failure
Object access	Success
Policy change	Success, failure
Privilege use	Failure
System events	Success, failure

Table 5.3: Suggested auditing configuration settings.

Use RunAs

Using the Win2K and later RunAs command will allow you to perform the functions that require administrative access while logged on as a user with normal access. One reason to avoid being logged on as an administrator unnecessarily is that any viruses or malicious code you should encounter will run in the security context in which you are logged on. With administrative permissions, such a situation has the potential to be far more damaging not only to your system but to others on the network.

Examining Your Security

Once you've implemented your desired security settings on your systems, it's time to check whether you missed anything. After systems have been in use for a few months, do they still comply with your desired settings? You've implemented different security policies to several OUs via Group Policy, do they conflict? Always perform periodic checks and be aware of the tools you can use to troubleshoot conflicts. In this section, we will discuss some of the tools available to help you do just that.

Microsoft Baseline Security Analyzer

The Microsoft Baseline Security Analyzer checks computers running NT 4.0, Win2K, and Windows XP for common security issues. The tool provides recommendations and is useful for identifying misconfigurations—it focuses on items that are appropriately set by default. IN addition, it lets you scan remote systems (provided you have access); thus, in a large environment in which systems are not well managed, this tool can be very helpful in identifying problems you might have otherwise assumed were not an issue for your network. It provides a HTML view as well as links to what it scanned and how to correct any problems (see Figure 5.8).

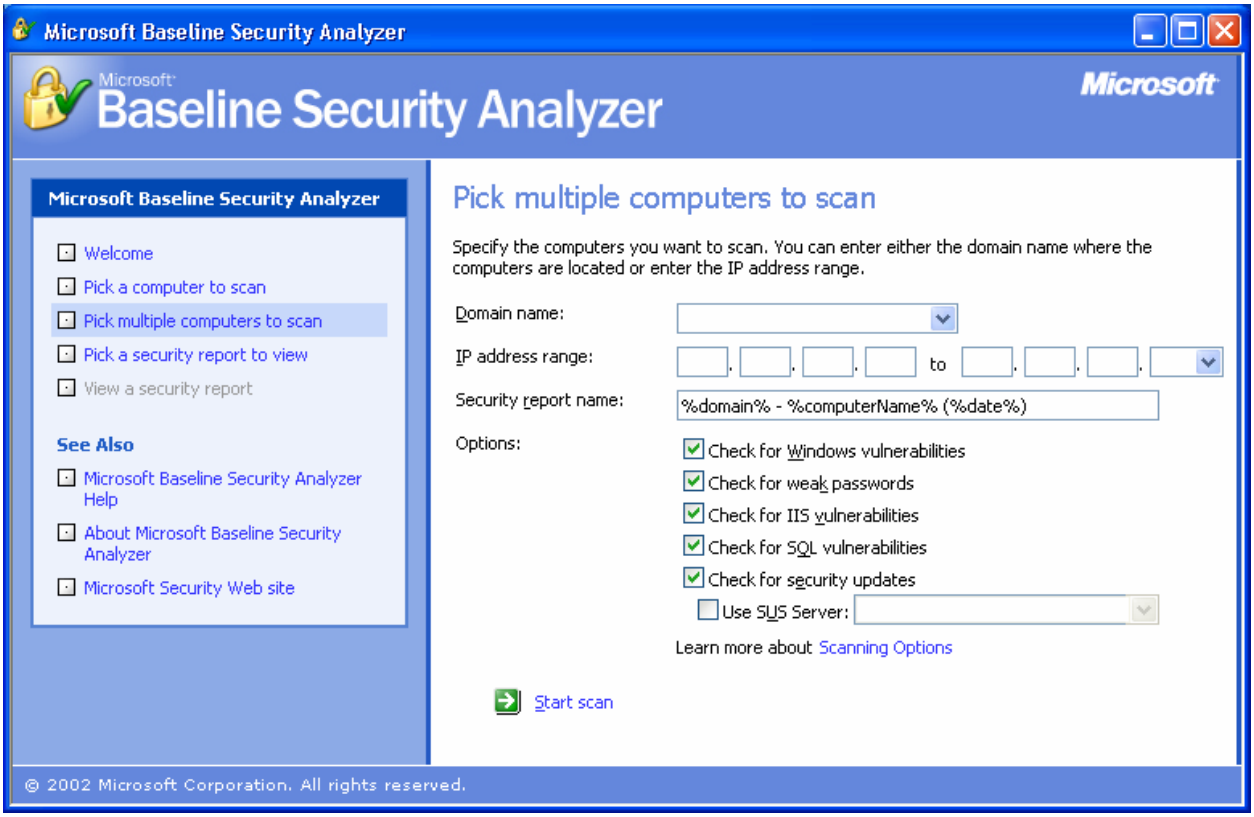

Figure 5.8: The Microsoft Baseline Security Analyzer.

The tool scans and reports on several security-related configuration settings for the Windows OS as well as some specific Microsoft applications such as Outlook and Office. The items processed as part of a scan of Windows include:

- Checking for missing security updates and service packs
- Checking for account password expirations
- Checking for which file system is in use for local hard drives
- Checking to see whether the autologon feature is enabled
- Checking to ensure that the guest account is not enabled
- Checking the RestrictAnonymous registry key settings (dictates access for anonymous users)
- Checking the number of local Administrator accounts (should be only one or two)
- Checking for blank or simple local user account passwords
- Checking to ensure that there no unnecessary services are running
- Listing the shares present on the local computer
- Checking to see whether auditing is enabled

> The Microsoft Baseline Security Analyzer is available for download at http://www.microsoft.com/technet//security/tools/tools/MBSAHome.ASP.

Security Configuration and Analysis

You can use the Security Configuration and Analysis Microsoft Management Console (MMC) snap-in to compare (or apply) a system with a security template:

51. Select Run from the Start menu, and type

    ```
    MMC
    ```

    into the Run text box, then click OK.

52. Select Add/Remove Snap-in from the Console menu on the toolbar, then select Security Configuration and Analysis, and click Add.

53. Click Close, then click OK.

54. Right-click the Security and Configuration Analysis node, and select Open Database. Here you may type in any name (the security database is just temporary storage for this program). You may also make a copy of the Secedit.sdb file from the Winnt\Security\Database folder from a domain server and specify this file to compare local settings with that of Group Policy settings.

55. Choose a template to import. You can also make your own template using the Security Templates MMC snap-in. The template is a database of recommended security settings. The following are some of the predefined templates that are available:

    - basicdc.inf contains basic security settings for an AD domain controller
    - basicsv.inf contains basic security settings for a Win2K server
    - basicwk.inf contains basic security settings for a Win2K Professional machine
    - compatws.inf contains low security settings for a Win2K Pro machine in a mixed mode domain (NT compatible)
    - hisecdc.inf contains high security settings for an AD domain controller
    - hisecwk contains high security for a Win2K server

    Choose the security template that is appropriate for your situation, and click Open.

56. Right-click the Security Analysis and Configuration node in the left pane, and select Analyze Computer Now from the resulting menu.

The computer's settings are compared with those that are recommended by the template, and when complete, you can navigate through all the items in the left pane. A red X will appear in the right pane for any settings that conflict with the selected template, as Figure 5.9 illustrates.

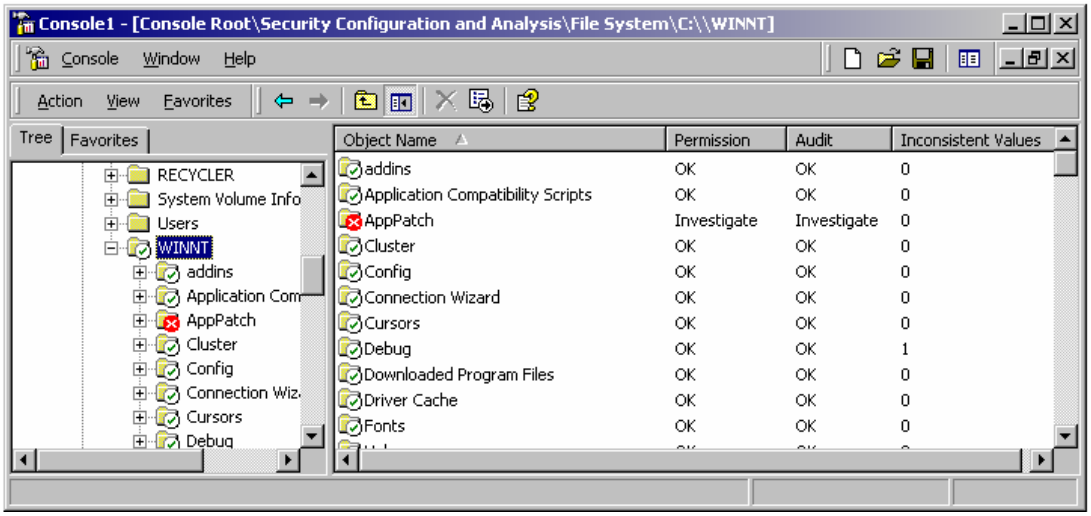

Figure 5.9: The security analysis results view.

> The National Institute of Standards and Technology (NIST) and the National Security Agency (NSA) have both published Win2K security guidelines as well as security templates that are available online at http://csrc.nist.gov/itsec/guidance_W2Kpro.html and http://www.nsa.gov/snac/win2k/index.html, respectively.

SecEdit

SecEdit performs the same functions as the Security Configuration and Analysis snap-in, but does so from the command line. It configures and analyzes system security by comparing a system's current configuration with a specified security template. SecEdit supports several switches. You can use the following switches to analyze security:

- /db—Specifies the path and file name of a database that contains the stored configuration against which the analysis will be performed
- /cfg—Specifies the path and file name for the security template that will be imported into the database for analysis
- /log—Specifies the path and file name of the log file for the process
- /quiet—Suppresses screen and log output, but the results can be viewed with the Security Configuration and Analysis snap-in described earlier.

> You can use Secedit.exe with the /REFRESHPOLICY switch to impose Group Policy object settings upon a target workstation. To immediately enforce GPO settings located within the machine node of relevant GPOs, type
>
> ```
> SECEDIT /REFRESHPOLICY MACHINE_POLICY /ENFORCE
> ```
>
> To do the same for the user node of the relevant GPOs, type
>
> ```
> SECEDIT /REFRESHPOLICY USER_POLICY /ENFORCE
> ```
>
> For Windows XP and later, this functionality has been replaced with the GPUPDATE command.

Event Log Management

Event logs are a good troubleshooting tool, and with auditing in place, there is often a requirement to maintain, back up, and archive log files for security reasons. You can configure and back up event logs in multiple ways, a few of which are explored in this section.

Configuring Event Logs' Retention

Locally, or via policy, you can configure the Application, Security, and Systems logs independently. You can specify how large they may get and how long you would like to keep them (see Figure 5.10).

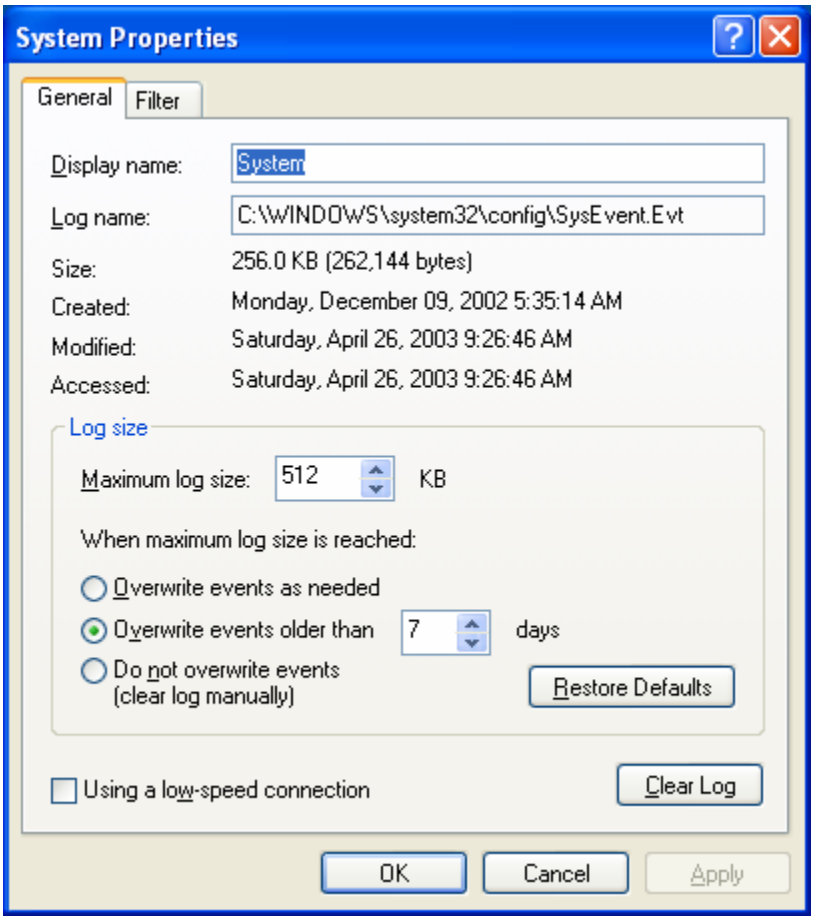

Figure 5.10: System event log default retention settings.

You can also configure event properties using Group Policy (see Figure 5.11) and assign those preferences to an entire domain or specific OU.

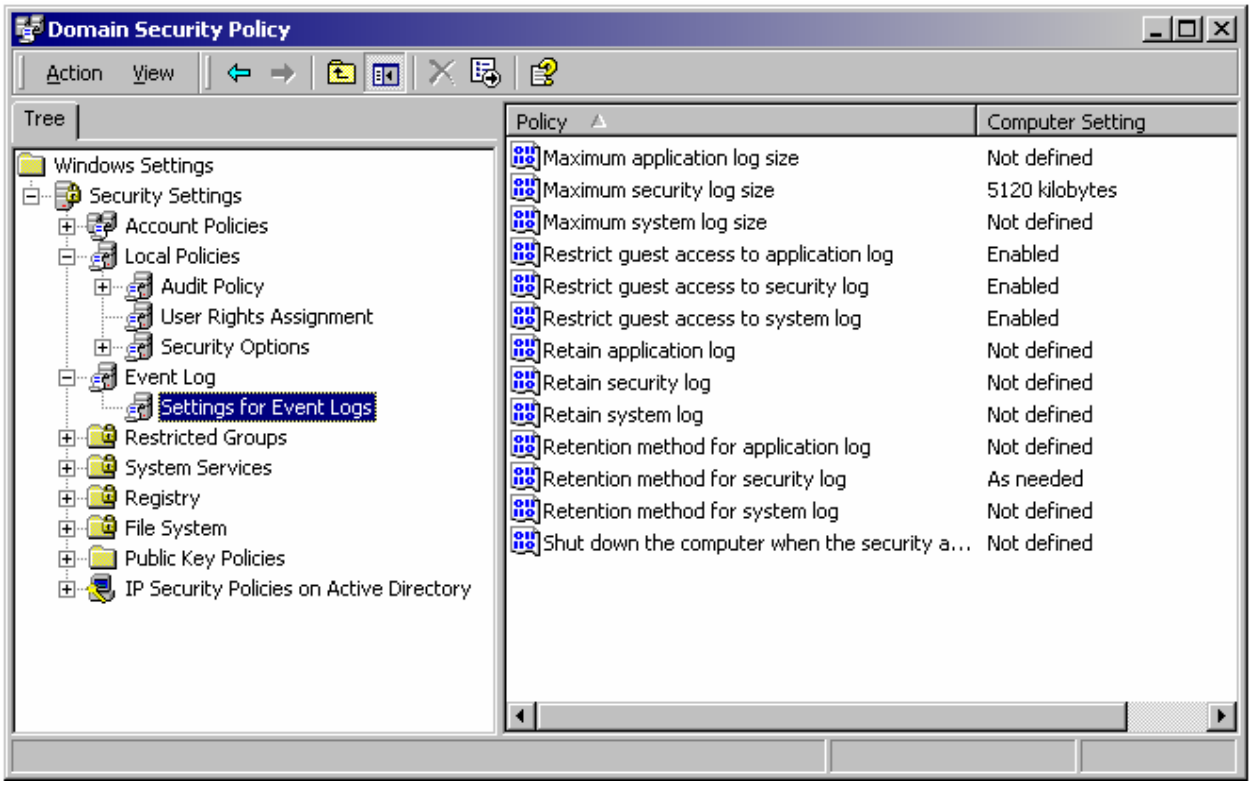

Figure 5.11: Domain security policy event log settings.

Archiving Event Logs

An audit trail can contain information about changes that are made to computers on the network. If intruders gain Administrator rights and permissions, or if administrators abuse their rights and permissions, they can erase event logs' entries to conceal their actions. If you regularly back up and archive security log entries across your organization, even if entries are cleared, you have an increased chance of being able to trace the actions of intruders or administrators using your archived copies:

- DumpEL—Dump Event Log is a command-line tool that dumps an event log for a local or remote system into a tab-separated text file. You can also use this tool to filter for or filter out certain event types. Although you can export the data in a tab-delimited or comma-delimited format, the tool does not dump the files in the native .evt file format that the Event Viewer uses.

> 🕮 DumpEL is available in the Win2K resource kit or online, at http://www.microsoft.com/windows2000/techinfo/reskit/tools/existing/dumpel-o.asp.

- EventSave—This freeware utility saves event logs in their native .evt file format to a specified location. It creates one file for each month named year_month_computer_xxx.evt, where xxx is the name of the log (Application, Security, or System).

For more information about EventSave, visit http://www.heysoft.de/Frames/f_sw_es_en.htm.

More robust solutions exist that include event log collection and reporting, such as Microsoft Operations Manager (MOM) those from NetIQ. Such systems can provide a consolidated view of an entire organization's event logs.

Summary

In this chapter, we have explored how to generate a security policy for your environment and the benefits and drawbacks of implementing a locked down environment. We also discussed some general security recommendations and methods that you can use to examine your own security configuration. In the next chapter, we will discuss desktop support, exploring methods and tools, including remote control.

Chapter 6: Desktop Support

In the last chapter, we explored implementing security in an effort to help users from inadvertently damaging their systems, to enforce the saving of data to network shares, and to prevent the installation of unauthorized software. However, even with these restrictions in place, you will certainly have desktop problems to contend with. Some of these issues might simply involve a lack of user knowledge or understanding, other problems will be legitimate issues that will require investigation and troubleshooting.

Adopting and Implementing Best Practices

Adopting industry best practices can help you to avoid learning certain aspects of desktop support the hard way. Guidelines provided by published best practices—which we'll explore in this chapter—can help you to identify issues that may not be clear to you in the early stages of implementing your desktop support processes. Investigating these best practices and how you can apply them to your environment can help you to avoid mistakes.

The most recognized source of best practices for IT can be found in the IT Infrastructure Library (ITIL), which is published by the United Kingdom's Office of Government Commerce (OGC). The ITIL defines overall service management as 10 core processes separated into two groups: IT service delivery and IT service support (see Table 6.1).

Process	Category
Capacity management	IT Service Delivery
Financial management	IT Service Delivery
Availability management	IT Service Delivery
Service level management	IT Service Delivery
Continuity management	IT Service Delivery
Service desk	IT Service Support
Incident management	IT Service Support
Problem management	IT Service Support
Change management	IT Service Support
Configuration management	IT Service Support

Table 6.1: ITIL service management processes.

When leveraging how others have done something (for example, implementing a best practice), you might feel that doing it exactly their way is a requirement. However, every situation is different and should be adjusted as necessary. Don't think of best practices as hard and fast rules; rather, consider them a way that others have found to solve a problem that works well in typical environments. One of the reasons that ITIL has been so widely implemented is that it does not provide specific standards and processes, but a flexible framework that you can adjust as needed.

📖 To learn more about the ITIL service management processes, visit http://www.ogc.gov.uk.

Regardless of which processes you implement, it is important that you use them consistently. You might have heard of International Organization of Standardization (ISO) 9001, which helps companies establish and consistently use processes. Even if your intention is not to become certified, the practice of establishing and consistently using a process is no less valuable. However, processes can (and should) be enhanced and updated as you discover worthwhile changes. (For information about a method of reviewing and improving processes, see the sidebar "The Continuous Process Improvement Model.") But to constantly improve processes, they must be defined in the first place. Regularly deviating from a set process can undermine the benefits of establishing such a process.

The Continuous Process Improvement Model

Continuous Process Improvement (CPI) is a systematic approach for improving performance and customer satisfaction through the enhancement of an organization's processes. The model is designed to help document processes, assess their current status and functionality, and utilize a variety of tools to increase process efficiency and effectiveness. Figure 6.1 illustrates the CPI method model:

- Design—Creating a process and determining which metrics are important to establish the success and performance of the process
- Execute—Automating, deploying, and training staff on the process
- Measure—Determining how the process is performing and what can be learned from this data to further improve the process
- Manage—Modifying the process to improve it as well as considering changes that have affected the process

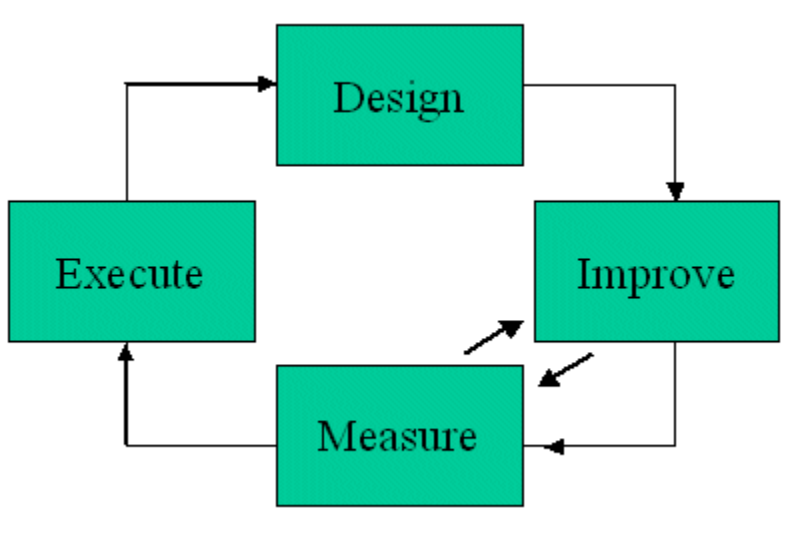

Figure 6.1: The CPI model.

Desktop Support Best Practices

One area in which implemented best practices and established processes can greatly improve the life of a desktop administrator is desktop support. Hopefully, you have done so much testing and research during the development of your baseline and in the deployment of updates and applications that the number of Help desk calls you receive is minimal. However, it is the nature of desktop support that there will always be at least a few users who will require your assistance.

When a user calls in a problem, it is very important that the issue be seen through to the end, which is very difficult without a tracking system in place. Particularly if several support staff are involved in solving a single problem, the need for a tracking system becomes clear.

Tracking Trouble Tickets

Tracking tickets through completion provides many benefits: they allow you to easily identify trends and widespread problems and to ensure that problems are followed through to resolution. When your time is consumed by many tasks, as is always the case for desktop support staff, documenting your work can become a challenge. To ease this burden, tickets can be generated from several sources—as we'll explore in the following sections—all of which should end up in your tracking database.

Web

Most Help desk software vendors such as Remedy, Altiris, and WebGroup Media provide a browser-based interface for tickets. The notion of running a client application to collect such information is quickly becoming an antiquated concept. A browser-based interface for recording a trouble ticket (and even managing them) is increasingly popular, as it requires no deployment and generates a ticket automatically in your database.

> 🕮 For a list of browser-based Help desk tools, check out http://dmoz.org/Computers/Software/Help_Desk/Browser_Based.

Telephone

A telephone call is another common way to receive trouble calls. This sort of ticket initiation is typically handled by having the caller provide the necessary information to support personnel for manual entry. This method of reporting is obviously less efficient than having the user enter data via a Web page. However, given the choice, many users are more comfortable speaking with someone so that they are better assured that someone is aware of their problems (rather than sitting in a queue that nobody has looked at yet).

Many organizations affix stickers to monitors to ensure that users know the right person to contact when they have a problem. This practice is a good idea—when users are already frustrated with problems, you do not want them to have to track down information to obtain assistance.

Email

A problem reported by email is one step behind that of a telephone-initiated ticket. Odds are against you that all the necessary information will be contained within the email, which means that you will have to track down the user by telephone to manually complete a ticket. However, if implemented properly, email can be as effective as a Web interface for the collection of information. To create an efficient solution, an email form is used and sent to an automated process for populating the necessary data into your support system database.

See Outlook Forms later in this chapter for additional discussion about email support requests.

Walk Ups

Walk-up requests and complaints are the worst of the ways to initiate a trouble ticket. Unfortunately, to the user, this method is often perceived as the best way. After all, they told Joe about the problem, and Joe always helps them out. Chances are pretty good that these problems will be handled without a ticket being created—or might end up forgotten altogether.

Without best practices in place, telephone, email, and especially walk-up support requests might go unrecorded in your ticket-tracking database. This lack of record would be an injustice, as you will not get credit for the problem you have solved. Most Help desk teams record data about the number of tickets they receive and how fast they are resolved as an important metric in judging the effectiveness and success of the team. Without a tracking system in place, you can easily answer questions and solve support problems throughout the day, but it will look to management like you did nothing.

Using a Knowledge Base

Knowledge sharing is essential to providing an increased level of support and, when made available to users, a decreased number of support calls. A knowledge base is a repository for recording solutions to problems identified on your systems, specific to your organization.

There are several benefits of a mature knowledge base. Containing the past experience of those that are either more knowledgeable or have spent several hours or days discovering a solution, a knowledge base can help you avoid having to determine a solution for the same problem twice, and might even help to reduce the volume of trouble calls:

- Faster resolution times—When a user calls in with a problem, a quick search of your knowledge base might provide the fastest answer. Most enterprise support departments utilize a knowledge base as an early step in the process of managing reported problems. In addition to speeding answers to users, this resource benefits less-experienced personnel, giving them the assistance of knowledge shared by those more knowledgeable in dealing with a specific problem.

- Problem call reduction—Making a knowledge base available to users via an intranet Web site will benefit some organizations more than others; in the end, it should reduce the number of trouble tickets by some degree. A more technology-based organization is more likely to have users that would prefer to help themselves. Many problems may indeed require administrative privileges to solve, and some users will always pick up the phone before trying anything themselves. But an available knowledge base lets you take advantage of the population of users that will do whatever they can before they pick up the phone.

> 🕮 Properly naming and categorizing a knowledge base entry can be critical to the useful of this resource. Failure to do so effectively can result in situations in which support staff and users cannot find what they are looking for even when the answer they seek has been provided. For some guidance in this area, you can obtain a draft of the Distributed Management Task Force (DMTF) paper "Using CIM for Support Activities" at http://www.dmtf.org/standards/cim_schema_v25.php as + CIM_Suport25.

A knowledge base can certainly be an excellent asset, but if it is not managed and maintained, it will be worthless. Ensure that recording resolutions to problems in the knowledge base is an integral part of your process. Further, it is important that those entries be reviewed before becoming available to the rest of the support staff and user population:

- Entering solutions—If the answer to your question is in the knowledge base already, use the available information to close your ticket. If it is not or you need to perform additional actions or checks not present in the solution article, it is important that the knowledge base information be updated. If the solution to the problem you are addressing is not contained within the knowledge base at all, you need to create a new entry. Either way, new and modified entries should go through an approval process of some kind.
- Entry approvals—A manager with a good background in troubleshooting (writing skills would be a definite plus) should be positioned to approve all knowledge base entries and modifications before they are made available. A poor practice or potentially damaging resolution is certainly not something you want in print for future encounters.

Employing Remote Control

Remote control tools are among the most widely used desktop troubleshooting tools. Whether the system your are supporting is down the hall or across the world, remote control is a direct and convenient way to help a user through a problem or perform small-scale checks or changes.

If you don't mind a lack of support and a reduced set of features, you can find a freeware solution to just about any software requirement—remote control is no exception. Microsoft has begun including built-in support for such functionality and offers NetMeeting free of charge. Additional solutions include the very popular Virtual Network Computing (VNC) remote control software from AT&T. However, if you're looking for a more complete feature set, consider a third-party tool such as ScriptLogic's Web-based remote control tool called Desktop Authority or Symantec's pcAnywhere.

Microsoft

Microsoft provides several remote control solutions. Remote Desktop and Remote Assistance are available only for Windows XP and later systems. NetMeeting is available for a much wider range of Windows systems and, although not free, Systems Management Server (SMS) provides a commonly used remote control solution as well.

Remote Desktop is designed to provide remote desktop access to a Windows XP system. It is not a remote control tool, but I mention it in this context because it is a very helpful tool in desktop administration. Use of Remote Desktop will log off any user on the remote system. Further, someone sitting at the remote system will not see the desktop environment in which you are working. Thus, it is a useful tool for performing remote operations; however, for actually working with users on a problem interactively, Microsoft now offers its Remote Assistance tool (which I'll discuss in a moment).

To configure a system as a valid host for Remote Desktop, select the *Allow users to connect remotely to this computer* check box on the Remote tab of the System Properties applet (see Figure 6.2). By clicking Select Remote Users, you can specify groups or individual accounts that are authorized to connect to the computer.

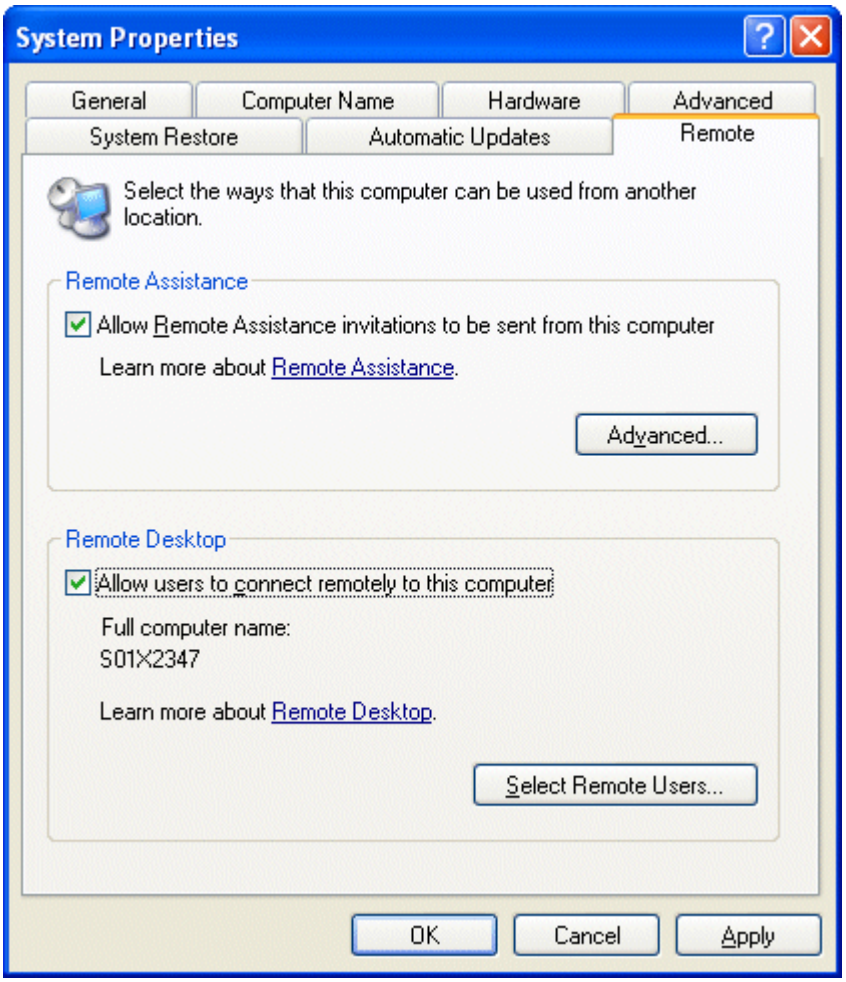

Figure 6.2: Use the System applet to control Remote Assistance and Remote Desktop settings.

This setting enables the system to function as a host (to be remotely controlled). To actually perform remote control operations, you will need to utilize the Remote Desktop Connection client (a Terminal Services client software component). On Windows XP systems, this program is available in the Communications program group under Accessories in the Start menu.

For Windows 95, Windows 98, Windows ME, Windows NT, and Win2K systems, you can install the client software so that you can connect to any properly configured Windows XP client. To install the Remote Desktop Connection client, insert the Windows XP CD-ROM, and click *Perform additional tasks* from the Welcome page. You can also download the client setup from Microsoft's Web site at http://www.microsoft.com/windowsxp/pro/downloads/rdclientdl.asp.

> Accounts used for remote administration must not have blank passwords. Hopefully, this requirement will not be an issue in your production environment; in your lab, this issue is one to be aware of.

Not to be confused with Remote Desktop, which only allows one active session at a time per computer, Remote Assistance allows both the owner and the remote user to control the computer at the same time. It also provides additional support features such as keyboard and microphone chat.

Because security is a concern with this feature, establishing a connection is not a one-step process (giving the remote computer plenty of opportunities to deny the session). The following steps walk you though the process of initiating a Remote Assistance session:

User:

57. Click Start, Help and Support to open the Office XP Help and Support Center.
58. Click the link in the upper left *Invite a friend to connect to your computer with Remote Assistance*.
59. Select *Invite someone to help you* from the main Remote Assistance screen.
60. At the bottom of the screen select *Save invitation as a file (Advanced)*. MSN Messenger or any MAPI mail account can also be used to initiate the session at this step in the process.
61. Specify the name that you want to appear on the invitation, and set a time period that the invitation should be set to expire in hours, minutes, or days.
62. Next specify a password, which the remote user will need to enter to make use of the invitation being created.
63. Click Save Invitation, and you will be prompted for a location to save the file, which will have a file extension of .msrcincident.
64. Email (or use any preferred method) to get the file to the administrator.

Administrator:

The administrator must open the file received from the user. When he or she does so, as Figure 6.3 shows, a prompt will appear to connect and supply a password (if specified).

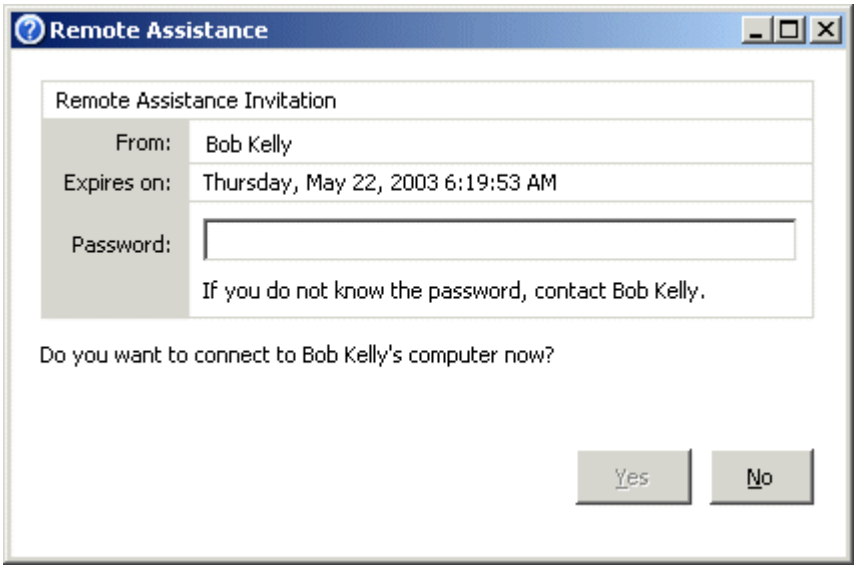

Figure 6.3: An example Remote Assistance invitation.

The Remote Assistance application will launch and wait for a response from the user.

User:

> Remote Assistance prompts the user with a window advising that the administrator has accepted the invitation and is ready to connect to the user's computer. The user must accept this prompt by clicking OK for the session to begin.

Once the session is established, a Remote Assistance window is presented with chat, file transfer, and control options.

Administrator:

> When first initiated, the remote system can be viewed but not manipulated by the administrator. To request control, the administrator must click Take Control at the top left of the screen.

User:

> The user is prompted to allow or disallow the administrator from being able to share control of the user's desktop (mouse and keyboard movements).

If the local or remote computer is behind a firewall, the firewall will need to be configured to allow Remote Assistance traffic via outbound TCP port 3398.

Remote Desktop Web Connection is a Win32-based ActiveX control (COM object) that you can use to run Remote Desktop sessions from within Internet Explorer (IE). This version is the latest of what was previously referred to as the Terminal Services Advanced Client (TSAC). You can download this control from http://www.microsoft.com/windowsxp/pro/downloads/rdwebconn.asp.

Another free Microsoft remote control tool, NetMeeting is often used for chatting and video conferencing. However, there is also a desktop sharing option that allows users to share specific applications and allow or disallow control of the application to the remote user. As Figure 6.4 shows, a PowerPoint presentation is being shared and a dialog box that is not shared has appeared on the screen of the host system. Because the contents of the dialog box are not being shared, this remote view hides the dialog box from the remote system. The dialog box covers a portion of the work area, and NetMeeting presents a box pattern in its place, effectively hiding the dialog box.

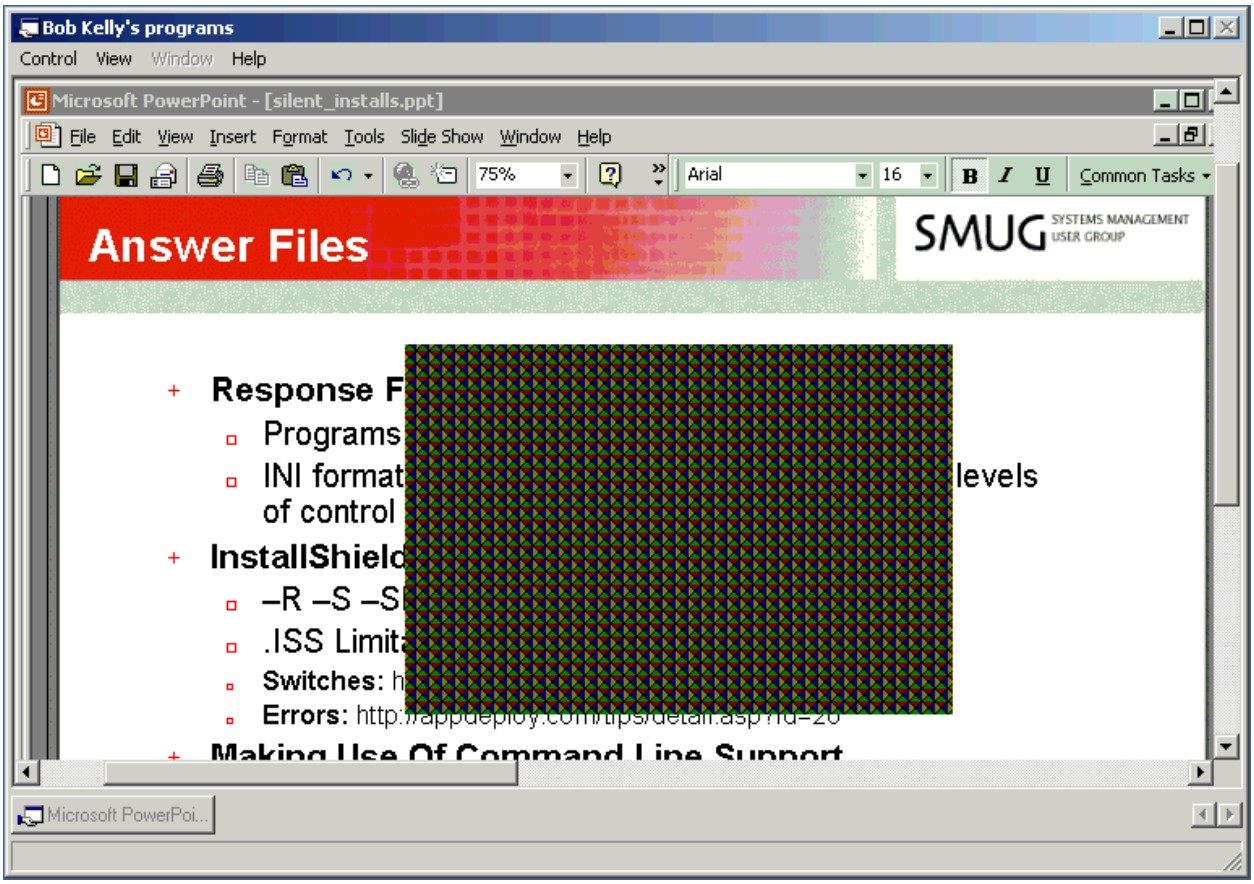

Figure 6.4: NetMeeting desktop sharing allows for the sharing of specific applications; others, such as the unavailable dialog box that appears here, cannot be seen by the remote user.

Although not free, Microsoft's SMS tool it is worth mentioning here as its remote control tool is among the most widely used in large organizations. (SMS may be considered "free" to those that own BackOffice or have its licensing covered by an Enterprise License Agreement). Although very useful, this tool is a complicated and robust program that offers much more than remote control features. Thus, SMS can be very dangerous in the wrong hands. If you are using, or plan to make use of, SMS as your remote control solution, be sure to restrict access to those features not necessary for desktop support. For details about how to use SMS to remotely support users, see the Microsoft article "SMS: How to Set Up a Help Desk Administrator" at http://support.microsoft.com/default.aspx?scid=kb;en-us;252674.

SMS 2003 integrates three remote control technologies: SMS remote control, Terminal Services, and Remote Assistance. You control access to remote control through the SMS admin console.

> In many environments, Microsoft SMS is used almost exclusively as a remote control tool despite its many additional features (and their inherent complexities). In fact, such is the case to such an extent that about 80 percent of the resumes that I have seen that boast SMS experience turn out to be simply the use of its remote control functionality. Used as a verb, I have often been told "I SMS computers all day helping users."

AT&T

With a stable reputation and the fact that it is free, AT&T's VNC is a very popular remote control solution. Though its security features are lacking, it is very easy to operate and supports many OSs.

VNC utilizes its VNC protocol to redirect the display of a system. This tool's design is based on the concept of a remote frame buffer (RFB). The protocol simply allows a server to update the frame buffer displayed on a viewer (as Figure 6.5 shows). Because it works at this low level, you can use VNC on any OS. Currently there are client applications (viewers) for Linux, Solaris, Windows, Macintosh, DEC Alpha, and Windows CE. The protocol will operate over any reliable transport, such as TCP/IP.

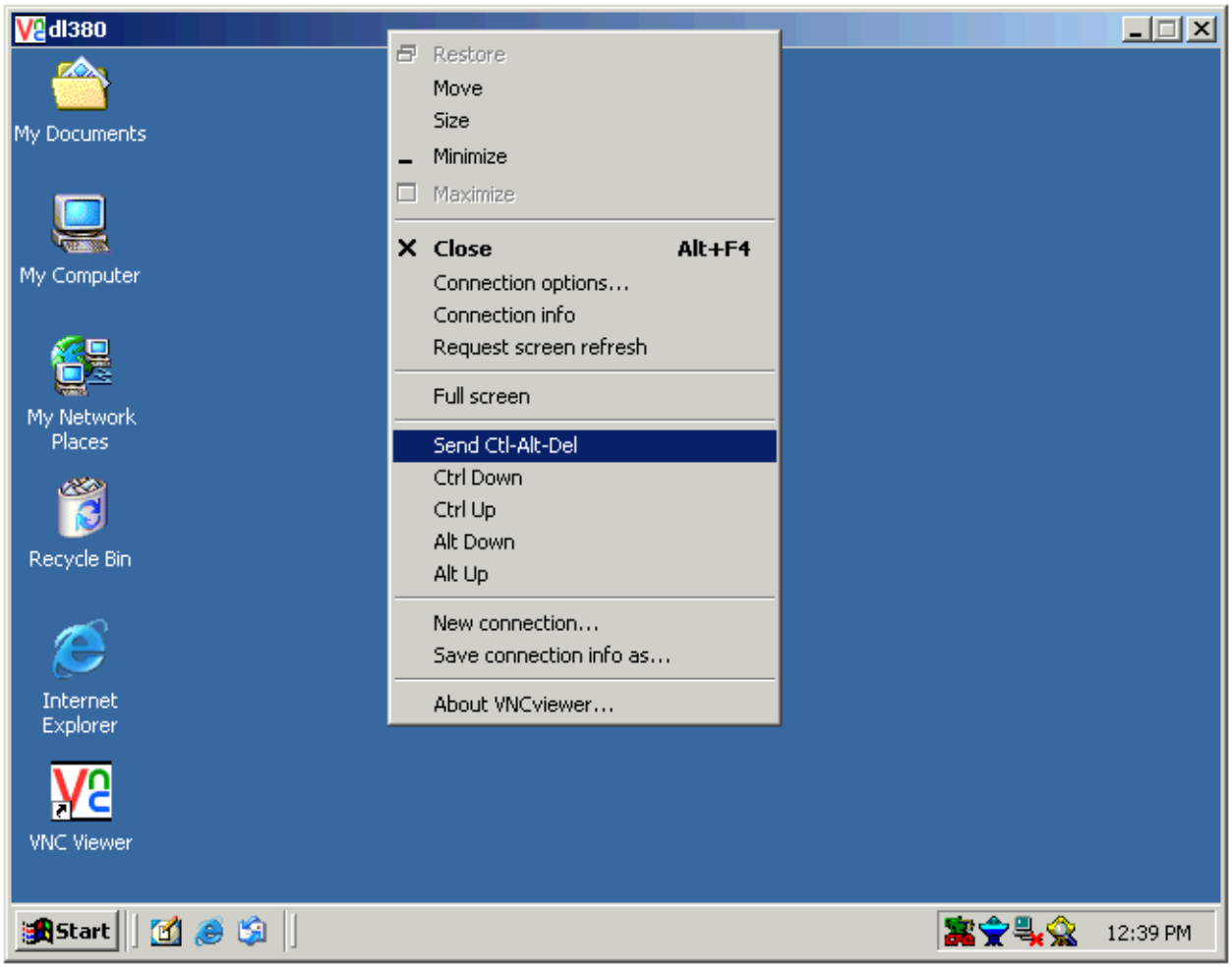

Figure 6.5: A VNC viewer session.

To use this tool, install the VNC Server on the machine that you want to remotely control. Then, using the VNC Viewer software, you can connect to any system running this VNC Server component. If you are having trouble connecting to a server, try specifying the IP address of the VNC Server system.

VNC is available from AT&T at http://www.uk.research.att.com/vnc/.

Another freeware remote control product that is starting to build a following is Specrem Pro. This tool is an open source project by Farid Bengrid and Antti Kirjavainen of Finland. In addition to the features that you would expect from a remote control product, Specrem also lets you control multiple computers at the same time. To learn more about Specrem Pro, visit http://specrem.sourceforge.net/.

Paying for the Good Stuff

In addition to these free (with the exception of SMS) remote control tools, there are several feature-rich third-party solutions available on the market for your consideration. Many of these products integrate with desktop management solutions and provide benefits not found in many of their free counterparts, including:

- Increased speed
- Increased security
- Integrated deployment
- Support for mobile devices
- No client software
- Cross platform support

Remote Control Vendors

The following list offers remote control tool vendors to get you started in your search:

ScriptLogic Desktop Authority at http://www.scriptlogic.com

Symantec pcAnywhere at http://www.symantec.com/pcanywhere/

NetSupport Manager http://www.netsupport-inc.com/NSM/

3am Labs RemotelyAnywhere at http://www.remotelyanywhere.com/

Netopia Timbuktu at http://www.netopia.com/en-us/software/products/tb2/

You can find feature summaries and links to downloads for a wide range of remote control solutions at http://appdeploy.com/downloads.

Working through Problems

Although it is beyond the scope of any book to tell you how to solve all your problems (wouldn't you love that book?), there are several typical obstacles and troubleshooting methods you can apply to most problems. In this section, I'll start out by discussing a few basic problems that are common to most networks, then we'll explore troubleshooting logic as it applies to desktop administration.

What is Your System's Computer Name?

When a user calls with a problem, do you know the computer name of the user's system? If you have chosen an effective naming convention for your systems, you might be able to identify the user's machine quickly. However, particularly in larger environments, the computer name is often an unknown fact that must be investigated. Such being the case, the ability to quickly resolve the user's problem begins with quick identification of the system about which the user is reporting a problem. The following list provides common best practices for doing just that:

- Asset management database—A well-maintained asset management database often identifies the primary user of a system. This database is the perfect look-up tool for accessing a wealth of information about the system and the user. Some support desks have even gone so far as to incorporate an automatic presentation of information based on the phone number from which the user is calling. Not having to ask the user for information other than what the problem is adds in a great level of efficiency to the support process.
- My Computer icon—When asked for the computer name of their systems, users often ask "Where should I look?" Having them bring up the System applet, then select the Computer Name tab is certainly not the most efficient way to spend the first minutes of a support call. One popular way to make this information easily available is to change the text identifying the My Computer icon to that of the computer name. To do so on Windows XP systems, manipulate the HKEY_CLASSES_ROOT\CLSID\{20D04FE0-3AEA-1069-A2D8-08002B30309D} registry key by setting the value of LocalizedString of type REG_EXPAND_SZ to %COMPUTERNAME%.
- Stickers—A sticker or other manually attached identification is yet another common way to make systems easy to identify. One primary benefit to this method is easy identification when the system is powered off. If the machine will not start, you are going to have a very difficult time getting a look at that My Computer icon.

Where is that Computer?

Once you have identified a computer on the network, often you don't know where it is located or to whom it belongs. If this situation sounds absurd to you, you are doing something right. A well-planned naming convention and asset-management system will make such information readily available. However, there are many organizations that need to improve things. To that end, the following list offers ways to identify the location of a computer on the network:

- AD—In AD, a location field is associated within the properties of each system (as Figure 6.6 shows). However, this value is only useful if it has been configured; the default value is not populated automatically. Setting this value should be a part of the process used when creating any new computer account on the network.

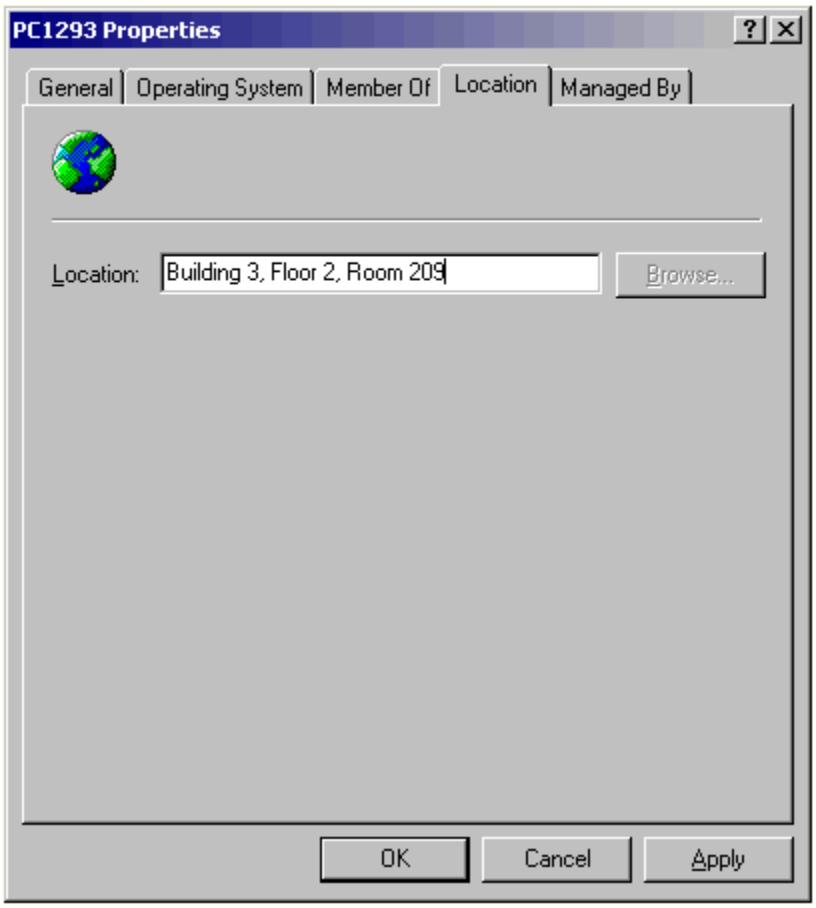

Figure 6.6: The AD Location tab in Computer Properties.

- Asset management database—As discussed previously, an asset management database is a common place to track the primary user of a system, among many other things—such as the location of a computer.

> 🕮 I cover asset management in more detail in Chapter 8.

- Locally stored data—When establishing your build procedures, it is a good idea to identify the system's location on the system as well as in a database. Creating a custom registry key and storing various identifying information there can be helpful down the road in a scripted or automated process that relies on such information. You would normally use computer groups (or organizational units—OUs) to identify collections of machines, but what if all the machines installed by one of the people on your team have a problem? Having the ability to base the correction for such a problem on a value that identifies the person who set up the computer can be very helpful (see Figure 6.7).

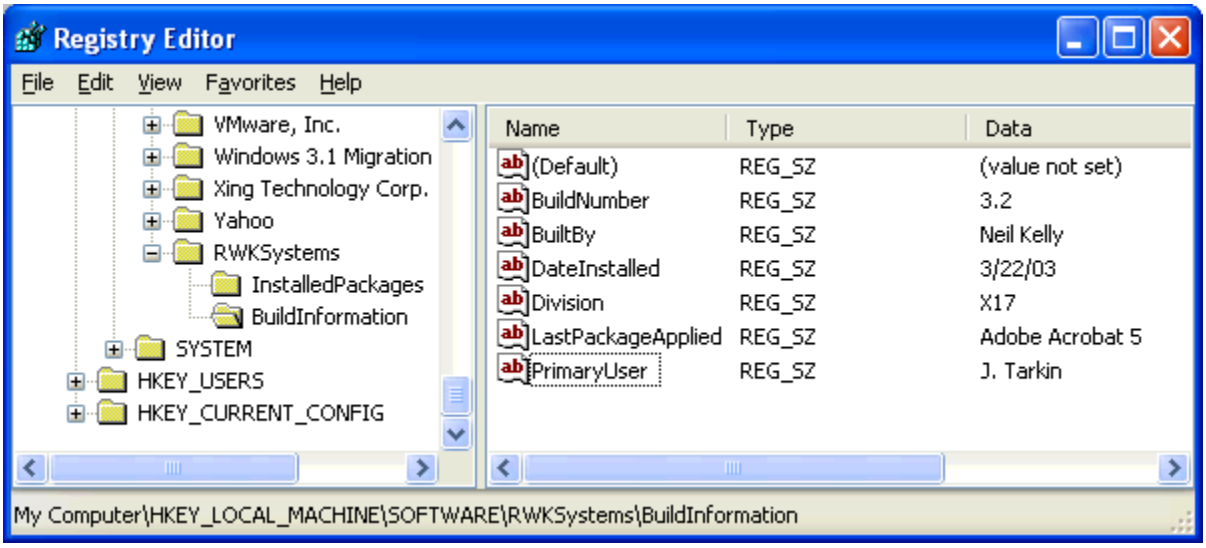

Figure 6.7: Locally stored build information for use by automation scripts or inventory.

- Net send—If you don't have a good naming convention or tracking database, give good old NET SEND a try:

  ```
  NET SEND <computername> Please Call The Helpdesk at x111.
  ```

  Then, sit back and hope for a phone call!

Troubleshooting Tips

Like I said earlier, there's not one answer to any IT problem, but there are a few things you can keep in mind when troubleshooting a problem. Particularly when dealing with Windows desktops, user profiles can often be a cause of grief.

> 🕮 For more information about user profiles, flip back to Chapter 3.

Profile Problems

Roaming or not, it is likely that the user has spent some time configuring his or her desktop and choosing system settings. Deleting this profile will cause the user to regenerate this environment, and in some cases, doing so could cost hours of time that the user might have spent being more productive. If the problem that the user is experiencing is also present for another person that logs onto the machine, it is unlikely to be the user profile. If the person has a roaming profile and is able to perform the same action without error after logging into another system, it is also unlikely to be a problem with the user profile.

If deleting the user profile should become a necessary troubleshooting step, try renaming the user's NTUSER.DAT file. Doing so will cause the user's files and application data to remain, but the HKEY_CURRENT_USER settings will be regenerated from the default profile. If the problem persists, try looking for and removing any related data from the user's Application Data directory that corresponds to the application or vendor that is causing problems. Finally, you can move or delete the user profile in an attempt to eliminate it as a problem altogether. Just be sure that anything you do can be undone if the user profile turns out not to be the problem's source.

Works On One Computer, But Not On Another

The obvious question to ask when something works on one system and not another is, what is different? Unfortunately, the answer is not so easy to expose.

"Just do a 'Diff'." If you have heard this response as often as I have, it probably makes the hair on the back of your neck stand up. Aggravating as it is to explain, you cannot simply compare two systems—there is too much to consider. Even if you were to dump the security settings, ACL settings, policy settings, file versions, registry settings, and data files to a comparable format, there would be so many inherent differences between two systems that the results would be very difficult to examine. Still, if you can narrow the problem to a reasonable scope, a comparison between a subset of a functioning and non-functioning systems may be helpful (see Figure 6.8).

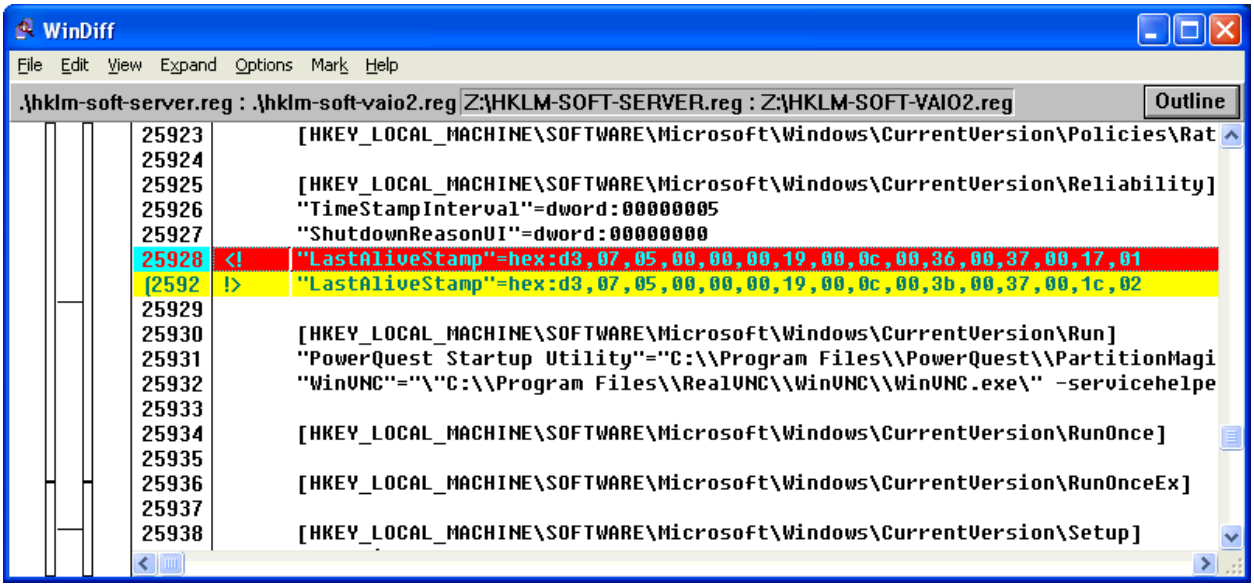

Figure 6.8: WinDiff file comparison tool view.

There are several tools available to perform file comparisons, and WinDiff is one such tool. It is available from many places, including the Support Tools folder on the Win2K and Windows XP CD-ROM as well as in the Windows 98 and NT resource kits, the Microsoft Windows OS Driver Development Kits, and with Visual C++. The tool shows all file contents by line number. Lines that are different show the first file entry in red and the line from the second file in yellow.

> 🕮 For step-by-step instructions for how to use WinDiff, see the Microsoft article "How to Use WinDiff to Compare Registry Files" at http://support.microsoft.com/default.aspx?scid=kb;en-us;Q171780.

Discovering what is installed on a machine is usually best performed by examining software inventory data. However, if you do not have such reports available to you, examining the registry can be the next best thing (when you know where to look). All software installed in Windows that offers the ability to be uninstalled is listed in the registry's Uninstall key. This key is located at HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall (see Figure 6.9). It accounts for most Windows software installed on a system (with very few exceptions—those that exist are usually made up of simple command-line utilities or 16-bit applications).

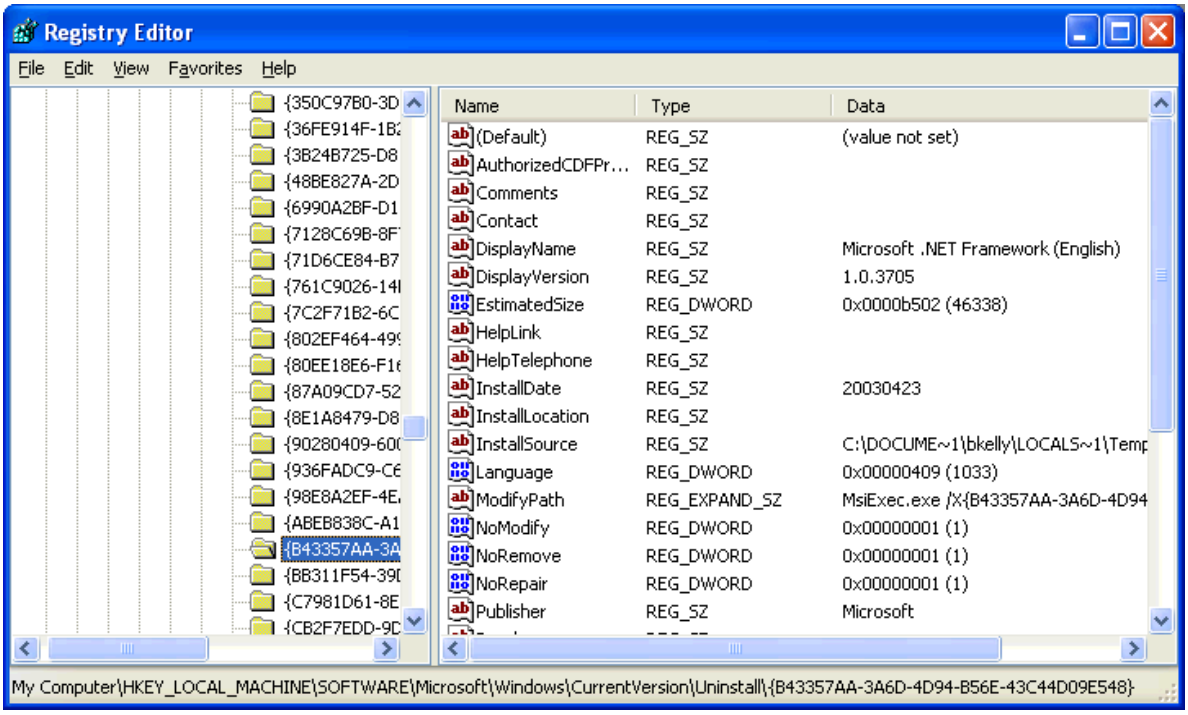

Figure 6.9: The Uninstall key lists Windows software installed with uninstall support.

A GUID identifies newer software keys, whereas older software will store uninstall details in a key with the application name as an identifier. Although the older way is better, the benefits of the new way far outweigh this simple issue. Before Windows Installer (the new way), software developers included whatever information they felt necessary, normally UninstallString and DisplayName; sometimes there would be more, sometimes less. The new way provides guidance about the inclusion of a great deal more data.

> 🕮 In the next chapter we will explore how to use this information and the KiXtart scripting language to generate an inventory.

The Web Is Your Friend

It is hard to imagine having to crack open a manual or call a support line every time you had a problem to investigate. The Internet provides an excellent administrative resource that cannot be understated. Chances are that when you encounter a problem, someone has encountered it before—possibly quite a number of people. Depending upon that number, the solution may be easy to find (or if not a solution, a tip to put you on the right track).

An increasing number of vendors now provide knowledge bases, answers to frequently asked questions (FAQs), technical manuals, and white papers on their Web sites. These resources are of enormous value and should always be investigated when a problem arises.

Newsgroups and user forums (message boards) can be another great resource, but use them with caution. Some such discussion areas are moderated, meaning that there is someone taking responsibility for content through a screening process of some kind. Most discussion areas are not moderated, providing no such filter for quality. In fact, even when moderated, the answers given may not be “official” answers and should be pursued with scrutiny.

Is it Worth the Time?

Before spending too much time on a problem, you should ask yourself "How long will it take to rebuild this workstation?" and "What will be lost if I rebuild this workstation?"

If the answer to a problem is escaping you, consider rebuilding the machine. Assuming you have a strong baseline and have done proper testing on your deployment packages and changes, any problems should be the result of non-standard configurations or file corruption—both of which can usually be quickly corrected by rebuilding the workstation. If you have automated your Windows deployment, take advantage of that work and use it to reset a computer to a known good state before spending much time and effort identifying a problem that cannot be easily replicated.

If a computer can be re-imaged in a few minutes, and you have spent the last 2 hours trying to figure out some odd inconsistency, you might be itching to make use of that image and be done with it. However, similar to deleting a user's profile, wiping out an entire computer can be devastating to users in some environments. The key is whether users' store data locally. Further, if the user has significant changes that fall outside your automated process, the user must be considered when weighing the speed of starting over again.

Keeping Users Informed

Email is the primary means of communication with users for most organizations. However, you cannot send an email to let users know that email is unavailable. And if there is an emergency of some kind, do you want to rely on users to check their email? It is important to have an alternative means of communicating with everyone available to you *before* it is needed.

> ☞ Was there a power outage over the weekend? Was a massive deployment initiated last night? Will your application servers be offline this afternoon? If you have to hunt down answers to these questions, time has already been wasted. Users aren't the only ones that you need to communicate with—it is vital that everyone on the support staff is aware of network-wide events and issues to avoid spending time troubleshooting the cause of a problem that may have an obvious.

Some phone systems provide a means of performing a one-way broadcast so that you can ring all phones and make your announcement as people pick up. It is important that this communication method is exercised; otherwise, users might pick up, hear a recording, and hang up.

As most users are sitting at their desks, a dialog box or visual warning is often implemented. Particularly when announcing impending server outages, this option is common.

Remember the first class you took on Windows, in which everyone discovered the NET SEND command? It lost its charm quickly then, but if you need to get a message out to a large number of people at one time, this method is effective.

> The messenger service accepts and displays only six messages. If more than six messages are received and not acknowledged, all subsequent messages will not be displayed.

To send to all computers within your subnet, specify an asterisk as the target system. To send a message to all computers within a domain, specify the domain name as the target using the /DOMAIN: <domain> parameter, as the following example shows:

```
NET SEND * Mail Server Outage In 5 Minutes
```

The message is received by the client workstation's Messenger Service, which presents the specified message along with the source system, target system, date, and time (see Figure 6.10).

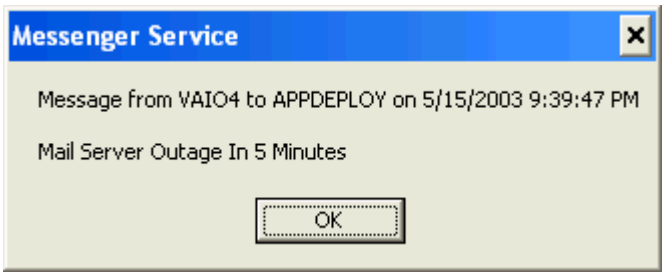

Figure 6.10: Sample message displayed by the NET SEND command.

> Unless your routers are configured to forward broadcasts (not normally the case) only computers on the local subnet will receive messages using the NET SEND * and NET SEND /DOMAIN options. You can send messages to specific computer across subnets, but not broadcasts such as this. It is therefore possible to send messages to a list of computers using a script; we will explore this option in the next chapter.

Shutting down a server? You can send a message to all connected users by using NT Server Manager's Send Message option. For Win2K servers, you can right-click the Shared Folders item in the Computer Management MMC snap-in, and select Send Console Message under All Tasks.

> Windows 95 and Windows 98 systems do not support the NET SEND command because it relies on server-side named pipes (functionality not present in Windows 9x). To receive such messages, the target system must be running the WinPopup utility. Because Windows 9x cannot run services, the utility must be added to the registry's RUN key or the Startup program group.

Monitoring the End-User Experience

You might think that the desktops in your environment are operating at a very high reliability rate, but people using these systems might think otherwise. If your users are happy with their desktop experience, you may not know until you ask. After all, things can seem pretty negative when you only hear complaints and problem calls. Know for sure by generating workload and success metrics and surveying your users.

Survey Users

How do you know what the users are experiencing if you do not ask them? Measuring success solely on the number of complaints you receive is a poor method resulting in an inaccurate view of the way your desktops are actually perceived. With the exception of those users that are a regular voice in your head (letting you know about everything they experience), most users will live with a problem unless it is affecting their ability to perform their jobs. It is not unusual for users to answer an error message every time they logon for months, until you happen to see it and ask them about it.

Email

Email is widely the most heavily used communication medium in business today. It seems only natural to make use of it in an attempt to learn more from your users as well. Although a bulleted list of questions provides some feedback, more robust means of utilizing email for this purpose are readily available.

- Outlook forms—Microsoft Outlook is easily the most commonly used email software on Windows networks, and it provides an Outlook Form Designer that you can use to provide a customized survey. All forms you create must be based on an existing Outlook form. First, select *Design a form* from the Forms submenu under Tools in the Outlook menu bar. When you select a form ,such as the Message form used to send email, you will be able to perform tasks such as add fields, insert images, and add controls. You can also specify field validation parameters to help ensure that questions are answered as you want them to be. It is a fairly simple interface (similar to that used for creating Microsoft Access database forms) and should serve to stimulate your ideas for how to gather information from your users. If you start to get carried away, you can use multiple pages in a single form. Figure 6.11 shows an example survey.

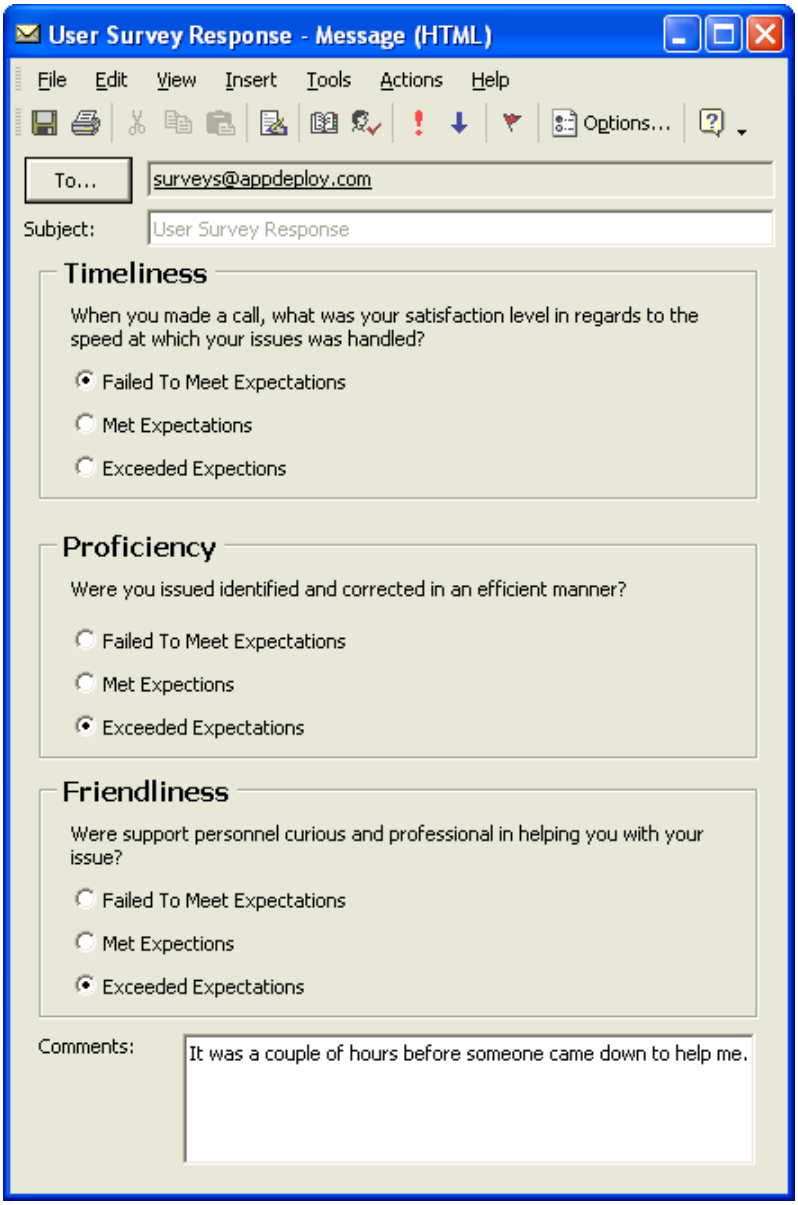

Figure 6.11: Example of a survey form created with MS Outlook.

> ☞ When working on a form, disable Microsoft Word as your email editor; otherwise, the forms design environment won't be available to you.

- IBM Lotus Notes databases—The IBM Lotus Notes email system provides an environment that lends itself to implementing a survey. Lotus Notes provides an interface that is made up of a series of database forms, making it a particularly suitable environment for such a requirement. Although it might take a bit more knowledge on the development side, most environments that utilize Notes as their mail system have some level of competence with Notes Development and are capable of generating a suitable database interface for collecting information from users.

- Other solutions—Web-based forms are also a logical choice for performing surveys, and several solutions are available as both products and online services to help you design, deploy, and analyze surveys. If you are looking for such a product or service, check out the following vendors:
    - Survey Said at http://www.surveysaid.com
    - Infopoll Business Intelligence Suite at http://www.infopoll.com
    - Zoomerang at http://www.zoomerang.com
    - ViewsFlash at http://www.cogix.com
    - SelectSurveyASP at http://www.classapps.com

Documenting Your Workload

Few departments are able to account for and prove their workload; however, the nature of desktop administration tasks makes them easy to measure and closely monitor. For example, a count of how many tickets were addressed and the average time to close those tickets is a metric often recorded for IT support departments. This information can be used to benefit the IT department:

- Due appreciation—It might seem inconsequential, but if you have a group of people working hard week after week handling users and solving problems, it is nice to show them what they have accomplished. During a migration or heavy software deployment period, the level of effort required of an organization's support staff will increase significantly.
- Acceptable delays—Along the same lines, if you are experiencing an increased number of support calls and it is taking longer to address them all as a result, user knowledge of this fact can go a long way. If informed of the situation, users will be more tolerant of perceived delays in support—some might determine that their problem isn't worth your trouble and eliminate the request altogether.
- Justification for staff increases—If your support team has a constant backlog of trouble tickets and can show that a healthy number are indeed being addressed each day, it is very easy to show a need for more personnel.

Summary

In this chapter, we looked at desktop support, including best practices, the benefits of a ticket-tracking system, and the importance of a knowledge base and remote control tools. We ended with a discussion of common obstacles and troubleshooting tips. In the next chapter, I'll cover scripting and how you can automate tasks and simplify administration through scripts.

Chapter 7: Scripting Custom Solutions

Even the most robust, all-encompassing off-the-shelf solution cannot provide an answer for every problem you are likely to face. The more complex the requirements of your organization, the quicker you will encounter a need to develop your own solutions—perhaps by extending the capabilities of a management system or utility that you already own. Acknowledging this need, many management solution vendors include in their tools an ability to facilitate custom scripting by providing integration through Common Object Model (COM) automation, application programming interfaces (APIs), and even their own scripting languages.

Scripting and automation are synonymous. If you need to perform the same actions every day, or hundreds of times at once, the need for a utility or script becomes clear. Many command-line tools offer functions designed for use in scripts. For example, suppose you want to reboot several computers; you can use the shutdown.exe utility that has been around for years, but it has a limitation—it accepts only a single computer name as a parameter. Although you no longer have to visit hundreds of machines to reboot them, you will still have to type each individual computer name hundreds of times as a parameter for this utility. However, you can write a script that can read your list of computers and execute the command line for each one. Doing so lets you spend your time writing a script instead of writing out a batch file or typing the shutdown command over and over again:

```
Shutdown.exe -r -m \\PC7463
Shutdown.exe -r -m \\PC3772
Shutdown.exe -r -m \\PC9283
Shutdown.exe -r -m \\PC0139
```

> 🕮 Later in this chapter, we will cover a script that you can use to restart multiple machines.

Should You Script It?

If it is faster to script a task than it is to perform the task manually, the decision of whether to script the task seems like an obvious decision. However, if you will need a week to script a task or a week to perform the task manually, what then? Consider the following benefits of scripting:

- Reuse—To use the previously mentioned shutdown script as an example, the decision to script the task might be questionable if only 50 machines need to be restarted. However, you must consider the fact that you will have this script available the next time those machines need to be restarted. If restarting computers is something you do often (as is usually the case for systems administrators who use Group Policy for software distribution), a script that does the job can be quite valuable.
- Customization—The ability to customize every aspect of your solution is a major benefit of scripting. However, this benefit comes with a drawback: Once you start to produce custom solutions, you will soon be asked for further customization and scripting can quickly become a large part of your day. (Tasks management might have originally seen as technically unfeasible will suddenly start to seem within their grasp—they can think up all kinds of ridiculous things for you to script!)

- Easily modified—Depending on how you write your script, you can easily make a script capable of change (such as running on an alternative network). In addition, when a change is requested, a well-organized script can make the process painless.

> ☞ One way you can make your script easy to modify is to use environment variables and to set your own variables for anything that could change down the road. Use of comments and in-line documentation can make your script easy to follow. Additionally, when you segment your scripts by functionality, you can more easily determine where newly requested actions could best be inserted.

You could spend weeks or months perfecting a script to make it look nice and take every situation into account. This effort is well worthwhile at times and sometimes it is getting carried away. Establish a set of goals for your script and try your best to stick to accomplishing them. Additionally, documentation, support, and the potential to do a great deal of damage must be taken into consideration as potential drawbacks of scripting:

- Documentation—Administrators are not typically fond of documentation. However, an undocumented script can lead to problems when the individual that wrote the script is no longer around to answer questions. I have seen months of effort scrapped simply because it is easier to start over than to figure out what the existing script is doing (and why).

> ☞ Every scripting language available offers some form of support for comments or remarks. These are lines in the script that are not processed when executed and that you can use for in-line documentation. Regardless of whether you plan to start generating text or Help files to explain and instruct others about your scripts, in-line comments will help others to quickly understand what is happening and will also help you when you come back to a script you wrote long ago.

- Widespread damage—The potential to do damage is as much a problem for the experienced scripter that makes a simple mistake as it is for the newbie scripter that wants to try out a new script function. Particularly when dealing with scripts that reach out across the network to many machines, it is very important that operations be tested in a lab or offline environment before letting them loose on your production network.

  Many environments restrict the use of development tools in production to prevent damaging mistakes but allow scripting to take place on the production network. This type of policy doesn't make sense—you can do just as much damage with a VBScript as you can with a Visual Basic application (see the sidebar "Script vs. Application: What is the Difference?").

Script vs. Application: What is the Difference?

There is a fine line between script and application. Some organizations define applications as executable files with graphical user interfaces (GUIs) that are executed by users. A script can certainly meet these criteria. For example, a KiXtart script could be bundled into an executable with the KiXscripts Editor (http://www.kixscripts.com), utilize COM to provide a graphical interface (http://www.kixforms.com), and, with a simple shortcut, be made available for users to execute. Figure 7.1 shows a KiXtart script that uses KiXforms to provide users with a means of browsing Windows Management Instrumentation (WMI) and its values.

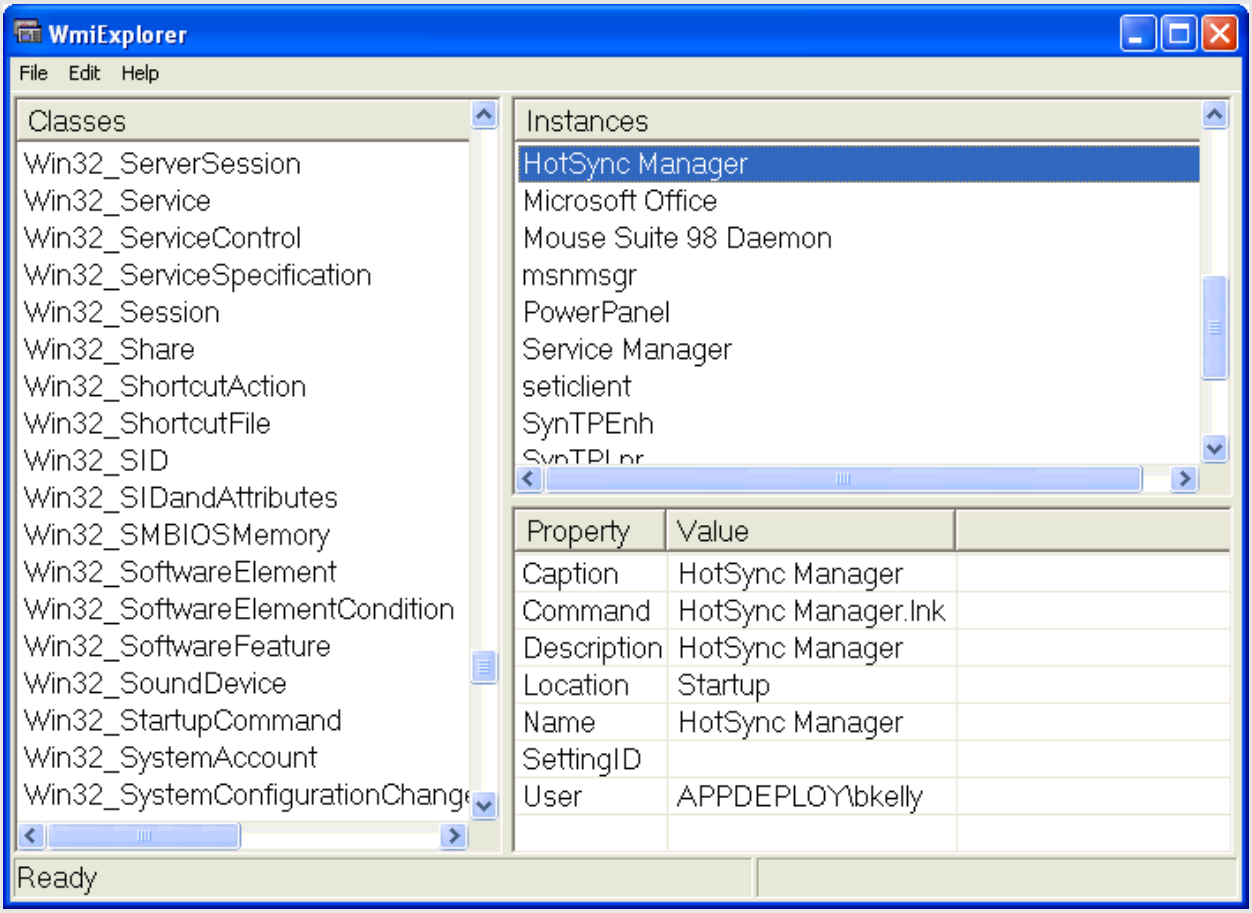

Figure 7.1: WMI Explorer script using KiXforms.

☞ An offline test or lab environment is ideal for script development. If you need access to a database or server-side component, consider replicating a minimal facsimile of what you require for your testing before using your script on a mission-critical server. If you don't have an entire lab at your disposal, consider products such as VirtualPC or VMware to emulate an entire network on your computer.

- Support—Using a third-party solution to perform a scriptable action ensures a certain level of support. If a change to the Windows OS or its security stops a custom script from functioning, who can you call?
- Upgrades and enhancements—Chances are, when you develop a custom script, it is with a very narrow view. You know exactly what you want the script to do, and you target the script to accomplish that goal. A third-party tool is typically enhanced and upgraded to remain competitive with similar tools. This competitive nature will likely result in the inclusion of features you would not have thought about nor had the time to develop.

Choosing a Scripting Language

There are many scripting languages available, and everyone has a preference. Like those that love Linux, Macintosh, or Windows, people generally have a strong opinion regarding the language they favor. Typically those administrators that learn a scripting language stick with it, and most administrators can prove that with enough effort, most any language can be made to accomplish much the same tasks. Although one language might be easier to learn or work with than another, completing a scripting task is really a matter of familiarity and creativity with whichever language you choose. For each of the languages presented in this chapter, I provide a sample for how to go about writing a value to the registry in an effort to help illustrate the complexity of each.

VBScript

VBScript is among the most popular scripting languages. It is still heavily used in Web sites that utilize Active Server Pages (ASP). With the introduction of the Windows Script Host (WSH), VBScript gained considerable exposure in the world of desktop administration. In addition, Microsoft routinely provides script examples using VBScript. VBScript makes heavy use of COM (which may be challenging to beginners), but there are several books and examples to help administrators become familiar with its use. The following example shows how to write a value to the registry using VBScript:

```
Set WSHShell = WScript.CreateObject("WScript.Shell")
WSHShell.RegWrite "HKLM\Software\BuildInfo\BuildNumber", "1.3A"
```

📖 You can download VBScript from http://msdn.microsoft.com/downloads/list/webdev.asp.

KiXtart

KiXtart was originally developed as a logon script processor. However, with each release, it offers more robust support, making it useful in most any situation. In addition to full COM support, KiXtart provides a number of built-in functions and variables that greatly simplify the scripting process when compared with performing those same actions using COM. Additionally, only the KiXtart executable is required (locally or on a network share) to execute KiXtart scripts in NT and later environments. KiXtart provides full support for Windows 9x as well (it requires a service to be run on one or more network servers in order to provide certain functionality). Its power, ease of use, and support for all versions of Windows make it very popular among administrators. The following example shows how to write a value to the registry using KiXtart:

```
$rc = WriteValue ("HKLM\Software\BuildInfo", "BuildNumber",
"1.3A", "REG_SZ")
```

> 🕮 KiXtart is available for download from several KiXtart-dedicated sites, including http://scriptlogic.com/kixtart, http://www.kixscripts.com, and http://www.kixtart.org.

KiXtart Script Examples

As a result of its ease of use and robust capabilities, I have chosen KiXtart as the language to be used in script examples for this book. I did so for those not familiar with scripting, as KiXtart is an easy language to read and will help everyone more easily follow the examples provided. When it comes to administrative scripting, KiXtart is simple—especially when compared with most other languages. For those familiar with COM scripting, KiXtart provides a (COM) support—which will be familiar to those that know VBScript—with its CreateObject and GetObject functions.

KiXtart scripts are simple text files that you can write using any simple text or script editor. To run a KiXtart script, simply use your script file as a parameter when calling the KiXtart script processor (KIX32.EXE or wKIX32.exe) as the following line shows:

```
KIX32.EXE MyScript.kix
```

The KiXtart script processor may be called from a local or network location. Although the extension used is of no consequence, the .kix extension is typically used to identify KiXtart scripts. On Windows 9x clients, certain functionality is available only by installing the KiXtart RPC service on a network server. For more information, see the documentation included with KiXtart.

JScript

JScript (Java Script) provides similar capabilities to that of VBScript but with different formatting rules and built-in commands. JScript is Microsoft's implementation of JavaScript and is available as part of the Microsoft Windows Script. JScript relies heavily on COM. The following example shows how to write a value to the registry using Jscript:

```
sh = new ActiveXObject("WScript.Shell");
sh.RegWrite ("HKLM\\Software\\BuildInfo\\BuildNumber", "1.4A");
```

> 🕮 JScript is available as part of the Microsoft Windows Script at http://msdn.microsoft.com/downloads/list/webdev.asp.

WinBatch

Like KiXtart, WinBatch is a script language that gets a lot of attention for its ease of use. It is a powerful and fairly easy-to-learn language that has a dedicated following. WinBatch is not free, but a 30-day evaluation copy is available for download. Also available for WinBatch is a compiler that allows you to create freely distributable EXE files. The following example shows how to write a value to the registry using WinBatch:

```
RegSetValue (@REGMACHINE, "Software\BuildInfo\[BuildNumber]",
"1.3A")
```

> 🕮 WinBatch is available for download at http://www.winbatch.com.

PerlScript

PerlScript (available within the free ActivePerl) is typically a language used by those already very familiar with Perl. It is a fairly complex scripting language and is not recommended for those that do not already have a strong background in Perl. Although not necessarily a benefit when it comes to administrative scripting, ActivePerl is provided for Linux and Solaris as well as Windows. Although the development tools provided are not free, the ActivePerl distribution is. The following example shows you how to write a value to the registry using PerlScript:

```
use Win32::Registry;
if ($HKEY_LOCAL_MACHINE->Create("SOFTWARE\\BuildInfo",$Key))
{
      $Key->SetValueEx("BuildNumber", 0, REG_SZ,"1.3A");
      $Key->Close();
}
```

> 🕮 ActivePerl for Windows is available for download at http://www.activestate.com/Products/ActivePerl.

Shell Scripting

Most network administrators are familiar with shell scripting, commonly known as batch files. This widely used scripting language is often combined with other scripting languages. Although shell scripting has the most limited command set of the scripting languages, when used with other command-line tools (such as those from the Windows resource kits), shell scripting can be quite effective. To use shell scripting to write a value to the registry, you must call an external command-line utility.

COM Automation

Although the scripting language you choose will provide varying degrees of control over the functions you desire, most everything else can be handled via COM. COM provides the ability to share reusable code between vendors' applications as well as to expose certain functionality to other applications and scripts. Without getting too detailed about how COM works and because access to COM objects differs slightly from language to language, we will briefly cover a few COM objects that are commonly used in scripting.

> For more information about COM, visit http://msdn.microsoft.com/library/default.asp?url=/library/en-us/com/htm/com_757w.asp.

WMI

WMI is Microsoft's implementation of Web-Based Enterprise Management (WBEM). WBEM establishes an architecture for supporting network enterprise management. WMI provides the ability to gather a great deal of information about a computer including installed software, peripherals, and other system details. WMI does not (at least very well) access information stored in AD.

> If you like to learn by example, check out Microsoft's clever Scriptomatic. This tool generates VBScript samples for a chosen class of WMI data. The Scriptomatic tool is available at http://www.microsoft.com/technet/scriptcenter/WMImatic.asp. The PrimalScript script editor available at http://www.sapien.com provides a similar capability through its WMI script wizard feature.

> Check out the following sites for more information about WMI:
>
> WMI Tutorial at http://www.microsoft.com/downloads/release.asp?releaseid=12570
>
> WMI Scripting Primer at http://msdn.microsoft.com/library/en-us/dnclinic/html/scripting06112002.asp
>
> Microsoft WMI Scripting home at http://msdn.microsoft.com/library/en-us/dnwmi/html/wmiscript.asp

Active Directory Service Interfaces

Active Directory Service Interfaces (ADSI) is a set of ActiveX controls (COM objects) that present the capabilities of directory services from different network providers as a single set of directory service interfaces. You can use ADSI to manage network resources in a directory service regardless of which network environment contains the resource.

WSH

Many people perceive WSH as a scripting language. WSH actually allows Windows systems to execute VBScript and JScript scripts. Additionally, WSH provides a set of components that expose functions helpful to scripting. You can take advantage of WSH using any language that supports COM automation.

WSH is built-in to Microsoft Windows 98, Windows ME, and Win2K and later editions. For support for Windows 95 and NT 4.0 or to update to the latest version, visit http://msdn.microsoft.com/scripting.

ActiveX Data Objects

ActiveX Data Objects (ADO) is a collection of objects that provide an API to query data. Often used with Microsoft Access databases, ADO can also be used with different drivers and database engines to help you extract information from MS SQL, FoxPro, Oracle, and other sources, including Microsoft Word documents. The KiXtart script that Listing 7.1 shows uses ADO to record to an Access database the amount of free drive space on the local machine.

```
$CNstring = "provider=microsoft.jet.oledb.4.0;data source=" + $DBpath +
";persist security info=false"
$CMDtxt = "select * from COMPUTERS where computername = '" + @WKSTA +
"'"
$cn = CreateObject ("ADODB.Connection")
$cmd = CreateObject ("ADODB.Command")
$rs = CreateObject ("ADODB.RecordSet")
$cn.connectionstring = $CNstring
$cn.Open
$cmd.activeconnection = $cn
$rs.cursortype = 3 ; a snapshot of the recordset, not dynamic
$rs.locktype = 3 ; optimistic locking, assumes it is the only source
locking the record.
$rs.activecommand = $cmd
$cmd.commandtext = $CMDtxt $rs.Open ($cmd)

$DriveSpace = GetDiskSpace( "C:\" )

If $rs.eof = -1   $rs.addnew EndIf

$rs.fields.item("computername").value = @WKSTA
$rs.fields.item("domain").value = @DOMAIN
$rs.fields.item("drivespace").value = $DriveSpace
$rs.update

$rs.Close
$cn.Close
```

Listing 7.1: An example KiXtart script that uses ADO to record to an Access database the amount of free drive space on the local machine.

First, a connection string is established ($CNstring) that identifies the type, source, and security settings for the connection to the database. The next variable, $CMDtxt, is a SQL query string used to look for an entry in the database for the local workstation (@WKSTA is a KiXtart macro variable that stores this value). Next, we create the necessary objects and open the connection to the database. The connection is then set as the active connection so that we make use of it. The cursor type is set to three, which indicates a snapshot of the recordset (it is not dynamically updated), and the lock type is set to three, which indicates optimistic locking (assumes it is the only source trying to lock the record). The SQL query is then executed, and the KiXtart GetDiskSpace function is used to obtain the amount of free space for the C drive. If the SQL query fails to find a record that matches this computer, $rs.eof (the end of file for the recordset) will be set to negative one. If such is the case, we use the addnew method to identify this record as a new database record before updating the values in the database. Each value is written to the database, and the update method is used to apply the changes to the database. Finally, the recordset and database connection is closed.

> 🕮 One of the most common problems with the use of databases in scripts is the creation of a valid connection string. For details about the formatting of connection strings, see the Microsoft article "HOW TO: Create an ADO Connection from a Data Link File in Data Access Components" at http://support.microsoft.com/default.aspx?scid=kb;en-us;300261.

Internet Explorer

Internet Explorer (IE) exposes many functions to COM and many environments take advantage of the capabilities IE provides. The following two examples show how you can both automate actions and use IE to display information about the execution of your script.

Automate Actions

The following KiXtart script uses IE to navigate to the AppDeploy.com Web site:

```
$ie = CreateObject ("InternetExplorer.Application")
$ie.Visible = 1
$ie.navigate ("http://www.appdeploy.com")
```

First, the CreateObject function is used to create a session of the IE application in memory. As with many such objects, it is not visible to the user by default. If you were watching the Windows task list, you could observe a new instance of IE appearing as a process. This invisibility to the user is useful for automating tasks that users don't need to see; however, in this case, we want to see the action, thus the visible property for the IE object is set to true (a Boolean value where one is true and zero is false) so that the application is visible. Finally, the navigate method is used to instruct IE to navigate to the specified location—in this case AppDeploy.com.

Display Information

When most people think of a script, a DOS console window with text scrolling by comes to mind. Although there are certainly ways to make the information in the console window more attractive, you can utilize another method of displaying information that may be more "attractive" for your users. As the following script shows, IE can do the trick quite well (see Listing 7.2).

```
$ie = CreateObject ("InternetExplorer.Application")
$ie.AddressBar = 0
$ie.MenuBar = 0
$ie.StatusBar = 0
$ie.ToolBar = 0
$ie.Resizable = 0
$ie.Height = 450
$ie.Width = 400
$ie.Visible = 1
$ie.Navigate("about:blank")
While $ie.busy <> 0 AND @error = 0 Loop
$Doc = $ie.Document
$rc = $Doc.Open
$Doc.Write("<TITLE>Logon Window</TITLE>")
$Doc.Write("<BODY>")
$Doc.Write("Mapping Network Drives...<P>")
Sleep 2
$Doc.Write("Checking For Software Updates...<P>")
Sleep 2
$Doc.Write("Done!<P>")
Sleep 2
$rc = $Doc.Close
$ie.Quit
$ie = 0
```

Listing 7.2: Displaying information through IE.

In Listing 7.2, an object handle to the IE application is established so that we may begin addressing it. To keep IE from looking so much like IE, we hide the address bar, menu bar, status bar, and toolbar by setting them to false (zero). So that the display cannot be resized, we set resizable to zero as well, and establish the dimensions of the display with height and width values of 450 and 400, respectively. With this done, we make IE visible to the user by setting this property to one (true). We then navigate to a blank page, which is identified by the title about:blank. Because we do not want to start trying to write to IE before it has established the blank page for us, we loop until IE is no longer busy. We then open a new document and begin writing HTML to it using the write method. To keep the example short, we write a status message to IE, then set the script to sleep (pausing) a couple of seconds before displaying the next message. When done, we close the IE document and quit IE to destroy the IE object. Failure to do so will result in the window staying open.

Failing to close an object when complete can result in a new instance of the application being loaded into memory each time (as Figure 7.2 shows). Using a close or quit method is a common means of properly closing a document or application. In most cases, you can close objects by setting them to zero.

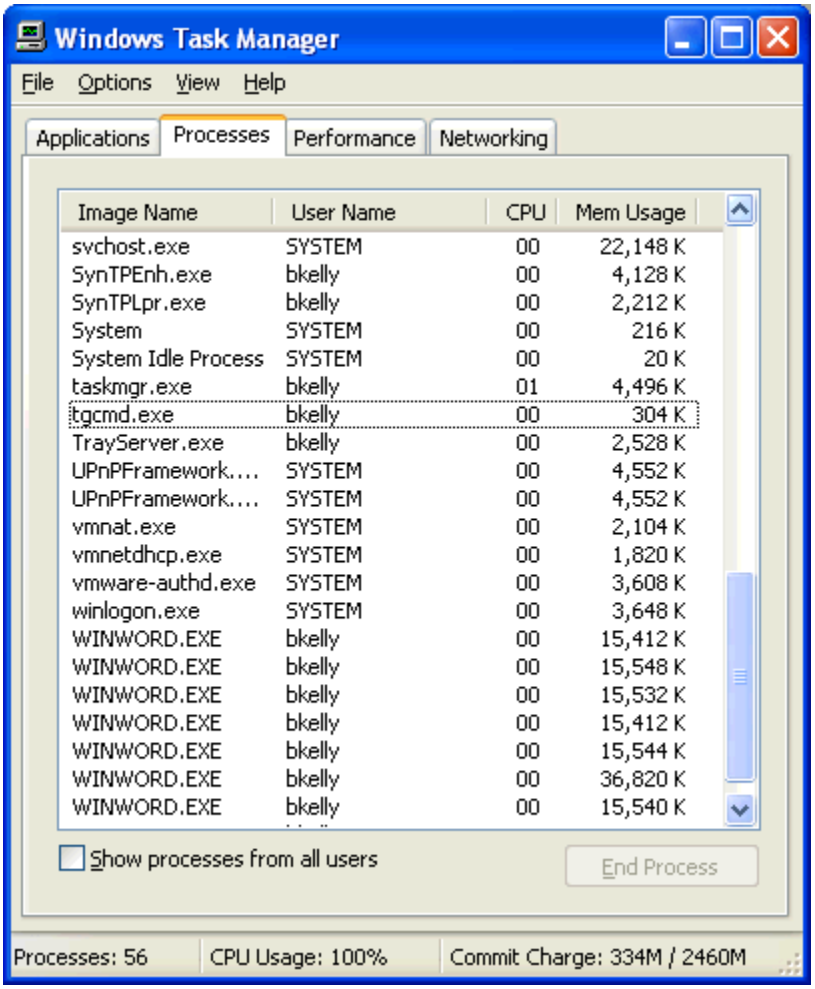

Figure 7.2: Task Manager shows multiple instances of Microsoft Word because it was not properly closed.

Running Your Scripts

Once you have created your script, how will you go about launching it? The answer is largely dependant upon the task you are trying to accomplish. If your script targets a user, a logon script, startup shortcut, or RunOnce registry entry is typically used. If you are targeting workstations, Group Policy, the scheduler service, and manually executed scripts are common choices. Let's explore each of these options.

Logon Scripts

Logon scripts are executed at logon before RunOnce registry entries and startup shortcuts. Unless third-party tools or utilities are being used, logon scripts always run in the security context of the user logging onto the system.

Logon scripts may be configured in the user profile or assigned to users by using Group Policy. Logon scripts are typically placed in the replicated NETLOGON share on a Windows network (%WINDIR%\system32\Repl\Export on NT and %SystemRoot%\Sysvol\Sysvol\<domain_name>\Scripts. on Win2K and later).

> If you don't specify a path, KiXtart searches files in the following order: NETLOGON, KiXtart Startup Directory, current directory.

For a Win2K client, the logon script runs minimized by default. To change this behavior, you must instruct the system to run logon scripts synchronously. Doing so will also cause Windows Explorer to not start until the logon script has finished running. Aside from making the logon script visible, this setting ensures that logon script processing is complete before the user starts working, but it can delay the appearance of the desktop.

If the Win2K client is a member of a Win2K domain, you can specify logon scripts should run synchronously by configuring the following setting in Group Policy:

```
User Configuration\Administrative Templates\System\Logon/Logoff
```

If the Win2K client is a member of an NT 4.0 domain, you can use System Policy to configure the following setting:

```
Computer Configuration\Administrative Templates\System\Logon
"Run logon scripts synchronously"
```

For Windows NT 4.0 or to make this change manually, set the value RunLogonScriptSync to a REG_DWORD value of 1 in the HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon hive. Alternatively, you can use a script to modify the HideLogonScripts registry value of type REG_DWORD and set the value to 0 in the HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\System hive.

Logon scripts are most often used to perform dynamic "per user" actions when a user logs onto the network. For example, Listing 7.3 shows a logon script that sets drive and printer mappings.

```
If InGroup("HR")
      Use P: "\\fileserver\hr_files"
EndIf
If InGroup("Managers")
      Use M: "\\fileserver\managers"
EndIf

$Floor = Left(@WKSTA,2)
Select
      Case $Floor = "01"
            $rc = AddPrinterConnection("\\printserver\floor1_prnt")
      Case $Floor = "02"
            $rc = AddPrinterConnection("\\printserver\floor2_prnt")
      Case $Floor = "03"
            $rc = AddPrinterConnection("\\printserver\floor3_prnt")
EndSelect
```

Listing 7.3: An example logon script that sets drive and printer mappings.

This script maps the P drive and M drive for those in the HR and Managers user groups, respectively. If a user is not in either group, neither of these drives will be mapped for them. If a user is in both groups, both drives will be mapped by this script. The script assumes that the two leftmost characters in the computer name identify the floor where the computer resides. A different printer is mapped depending upon which floor the computer is located.

RunOnce and Run

The RunOnce and Run keys may exist for both the machine (HKEY_LOCAL_MACHINE) and the current user (HKEY_CURRENT_USER). Any machine RunOnce values execute prior to the loading of the desktop. Then after the desktop loads, any values specified in the Run keys for the machine and the user are then executed (respectively), and finally RunOnce for the current user is executed.

The specified commands will be executed in the context of the user logging on to the system. Although Run values remain on the system and execute during each logon, RunOnce values execute and then remove themselves from the RunOnce key so that they do not run again:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce
```

To specify an entry, simply create a value in the appropriate key, and give the value any name you like (the value name does not matter, but should be something to help you identify its action). Then, as the data for the new value, enter the command line or script path that you want to execute.

> Never call a script from RunOnce that inserts an entry into RunOnce or you will encounter an endless loop. RunOnce does not continue the logon process until all entries are executed and the RunOnce key in the registry is free of entries.

Startup Shortcuts

Startup shortcuts are executed after the logon script and any calls made by Run and RunOnce values. If there are multiple items in the startup folders for the current user, those items are triggered alphabetically starting with those assigned to all users. The following list shows the order of startup execution:

65. Network logon script
66. HKEY_LOCAL_MACHINE RunOnce entries
67. HKEY_LOCAL_MACHINE Run entries
68. HKEY_CURRENT_USER Run entries
69. HKEY_CURRENT_USER RunOnce entries
70. All users' startup folder shortcuts
71. Current user startup folder shortcuts

> For a useful article about startup sequences, including additional actions and entry points into the logon process, see the ScriptLogic article T1056 at http://www.scriptlogic.com/support/kb/displayarticle.asp?UID=66.

Schedulers

You can use the following schedulers to configure events to occur at specified times:

- At.exe—At.exe is a command-line scheduler that Microsoft introduced with NT. This scheduler provides the ability to schedule events to occur on either a local or remote system—a very valuable desktop administration tool. For example, the command

  ```
  AT 20:00/every:m,t,th,f "C:\Chk4Updt.bat"
  ```

  causes C:\Chk4Updt.bat to be run each Monday, Tuesday, Thursday, and Friday at 8:00 PM.

- Soon.exe—Soon.exe is referred to as a near-future command scheduler. It utilizes the at.exe scheduler to set the specified event to occur a couple of seconds in the future. With at.exe, it is tricky to schedule an event to occur right away—if you are off by just a few seconds, the task is set to run the next day. On top of this problem, you cannot assume that your system clock and the remote system clock have the same time configured. Soon.exe takes care of this problem. For Windows XP and later, the SchTasks.exe utility supports a parameter to identify a task that should be run immediately.

- SchTasks.exe—Windows XP and Windows Server 2003 include SchTasks.exe, which replaces the NT and Win2K command-line scheduler at.exe. SchTasks.exe allows an administrator to create, delete, query, change, run, and end scheduled tasks on a local or remote system. The following command-line parameters control the SchTasks.exe command-line utility:

  ```
  /Create to create a new scheduled task
  /Delete to delete a scheduled task
  /Query to display scheduled tasks
  /Change to change the properties of a scheduled task
  /Run to run a scheduled task immediately
  /End to stop the currently running scheduled task
  ```

- WMI—You can use WMI to manipulate the Windows Task Scheduler, though the format for identifying dates and times is rather complex. The time to run the job must be specified in the form of YYYYMMDDHHMMSS.MMMMMM(+-)OOO. The date (YYYMMDD) must be replaced by ******** because the scheduling service only allows jobs to be configured to run one time or run on a day of the month or week (and cannot specify a specific date). In the following KiXtart sample code, a utility named DiskCheck is configured to run every Wednesday at 4:25 AM:

```
$Computer = "."
$WMIService = GetObject("winmgmts:" +
"{impersonationLevel=impersonate}!\\" + $Computer +
"\root\cimv2")
$NewJob = $WMIService.Get("Win32_ScheduledJob")
$rc = $NewJob.Create ("Notepad.exe", "********012500.000000-420",
True , 4, , True, JobID)
```

For more information about using COM to schedule jobs (including how to decipher these time and date values) visit http://msdn.microsoft.com/library/en-us/wmisdk/wmi/win32_scheduledjob.asp.

Manual Execution

To configure your script to perform actions on remote systems, simply running the script from your workstation may be all you need to do. Any security restrictions for remote connections and rights will need to be taken into consideration. In most cases, executing an administrative script while logged on as a member of the Domain Admins group is more than sufficient to perform most all actions you want to automate. However, it is always best to utilize an account that has just enough permissions to get the job done.

Always be aware of the potential for scripts to result in widespread damage. When it comes to scripting, it is very important that operations be tested in a lab or offline environment before letting them loose on your production network. Such is especially the case when dealing with scripts that reach out across the network to many machines.

To access a file remotely on NT and later systems, you can take advantage of administrative shares. For example, to copy a file locally, you might use something like this:

```
Copy "\\server\share\file.exe" "c:\winnt\system32"
```

To copy the file to a remote machine, you can take advantage of administrative shares by changing this statement to the following:

```
Copy "\\server\share\file.exe" "\\computername\admin$\system32"
```

Depending upon System Policy or Group Policy settings in your environment, these administrative shares ($Admin, $C, $D, etc.) may be disabled.

In KiXtart, you can access remote registries in much the same way by preceding with the universal naming convention (UNC) path to the target system when identifying the subkey you want to work with (the computer name preceded by two backslashes). Of the following examples, the first example shows the reading of a local registry value, and the second illustrates the same action against a remote system by using its UNC name.

```
$EditorVersion = ReadValue
("HKLM\SOFTWARE\iTripoli\KixscriptsEditor", "Version")

$EditorVersion = ReadValue
("\\computername\HKLM\SOFTWARE\iTripoli\KixscriptsEditor",
"Version")
```

Group Policy

Group Policy can target machines or users at startup (before the logon prompt is presented) or during logon. Additionally, you can target scripts to specific domains, organizational units (OUs), or groups from within AD:

- Logon—A logon script runs in the user context as a user logs on to a workstation.
- Logoff—A logoff script runs in the user context as a user logs off of a workstation.
- Startup—A startup script runs in the context of the local system account when a computer is started, before the opportunity to log on is presented.
- Shutdown—A shutdown script runs in the context of the local system account when a computer is shutdown.

Assigning Scripts

On networks that have implemented AD, the preferred method of configuring logon scripts is to assign them via Group Policy. To do so:

72. Open the Group Policy snap-in (in Administrative Tools, Active Directory Users and Computers).
73. Right-click the OU (in the console tree) to which you want to assign the logon script. For all members of the domain, right-click the domain.
74. Select Properties, and select the Group Policy tab.
75. Create a new Group Policy Object (GPO) by clicking New or choose an existing GPO, and click Edit.
76. Double-click Logon in the right pane under User Configuration/Windows Settings/Scripts.
77. Click Add. Notice that a browse dialog box appears for selection of a file from within the GPO directory.
78. Copy and paste your script into this browse dialog box to make it available to the GPO, if you have not already done so. If you are using a scripting language that is not installed and requires a support file (such as KiXtart's kix32.exe), include this as well.

79. Select the script to be run as your script name. If you're using a language that uses an executable file to process the script, set that as the script name and use the parameters text box to enter the name of the script file. If you assign multiple scripts, the scripts are processed according to the order that you specify. To move a script in the list, click the script, then click either Up or Down.

Avoiding Repeated Executions

Because scripts assigned to a user will always execute, it is important to add a condition for execution when dealing with scripts that you do not want to run again and again. One method is to look for the result of the script you are trying to run. For example, if a logon script is to map a drive to a network share, the script can base this action on whether the drive is already mapped to the desired share. Another method is to drop a *footprint* that indicates an action has taken place. A footprint is a term used to indicate something left behind in this way (such as a file or registry entry).

Using a registry entry, the following KiXtart script checks to see whether the current date has been updated by this script today. If it has not, the script records the current date, then executes the script. When the script is executed again on the same day, the date will match and the script will not be called.

```
If ReadValue("HKCU\Software\Scripts","TimeCardReminder.kix") <> @DATE
      $rc =
WriteValue("HKCU\Software\Scripts","TimeCardReminder.kix","@DATE","REG
_SZ")
      Call "TimeCardReminder.kix"
EndIf
```

Using a registry key, the script will check to see whether the script has been executed before. If the word Executed is not found in a registry value being used to identify a script that should only be run one time, the script is executed and the registry is updated to reflect it. When run in the future, the value will exist, and the script call will always be skipped:

```
If ReadValue("HKCU\Software\Scripts","Acknowledgement.kix") <>
"Executed"
      $rc =
WriteValue("HKCU\Software\Scripts","Acknowledgement.kix","Executed","R
EG_SZ")
      Call "Acknowledgement.kix"
EndIf
```

Building Lists

Tackling a project is the best way to get started in scripting, so let's get right into it with a common first step in many administrative scripts, building a list. To run an operation against several client workstations, it is common to base such actions against a list of machines. Although the same script can create the list and perform the desired action, it is common to dynamically perform actions against the list gathered without actually writing the machines to a list file. However, to keep the examples in this book shorter and more useful, I will separate the task of creating a list of machines with that of the scripts that follow, which will utilize the lists.

📖 For a comprehensive look at KiXtart, see *The Start to Finish Guide to Scripting with KiXtart* at http://www.kixscripts.com/book.

Machine List

Depending upon your environment, you can go about creating a machine list in a number of ways. In an AD environment, you would probably utilize your OU structure to target systems. You may also make use of the Windows network browse list to retrieve workstation names. I'll discuss and show you examples of these options in the following sections.

AD

For organizations that have taken on the effort of creating an AD domain, there is typically extensive planning involved. Therefore, making use of that planned structure is often a worthwhile approach to targeting systems with scripts. Aside from utilizing Group Policy to assign scripts to desired systems, you can use your AD structure to build a desired list of machines, which you can then use to perform actions on computers remotely from a manually executed script. For example, in the script that Listing 7.4 shows, KiXtart is used to write the machines found in AD to a file named machines.txt in the Windows Temp folder.

```
$rc = Open (1, "%TEMP%\Machines.txt", 5)
  $Container = GetObject ("LDAP://cn=computers, dc=intranet,
dc=appdeploy, dc=com")
  For Each $Computer In $Container
      If $Computer.class = "Computer"
            $ComputerName =
SubStr($Computer.Name,4,Len($Computer.Name))
            $ = WriteLine (1, $ComputerName + @CRLF)
      EndIf
  Next
  $rc = Close (1)
```

Listing 7.4: Using KiXtart to write machines found in AD to the machines.txt file in the Windows Temp folder.

In the open statement, the number one is a handle used to identify this file later in the script. The number five tells KiXtart to create the file if it does not already exist and to open the file for writing. An object is created to the computer container, as specified in the LDAP string (for information about how to create an LDAP query string, see the sidebar "Constructing an LDAP Query String"). The class property identifies whether a computer is a computer, group, and so on. Because the computer name is returned as CN=MyComputer, we use the SubStr function to create a new string ($ComputerName) to reflect all characters from the fourth character to the end of the string (thereby stripping the CN= portion of the name from the string). Next, we write this to the file opened earlier as 1, and include the macro variable @CRLF, which identifies a carriage return and line feed (otherwise all computer names would appear on one long line). Once each item in the $Container object is processed, the machines.txt file is closed.

Constructing an LDAP Query String

For beginners, the most difficult part about this script is getting the LDAP query string format correct. This task is not so difficult once you do it a few times.

`LDAP://`

specifies the provider type.

`cn=computers`

refers to the default container in AD. I chose this container because it is where all computer objects are placed by default. If you want to use actual OUs, then the cn= must change to ou=. The cn= is only for the default containers. You can distinguish the default containers, as they don't have the OU icon on them (unlike the ones you create). If the container you want to specify is more than one level deep, you must always start with the most specific container first. For example, ou=subOU, ou=higherlevelOU.

`dc=`

specifies the DNS domain name. As with OUs, each level must start with dc=, and each level must be comma separated. For example, intranet.appdeploy.com would be entered as

`dc=intranet,dc=appdeploy,dc=com`

Server Manager

For NT domains (as well as Win2K and Windows Server 2003 servers), you can query servers for computer names using a method similar to that of AD by using the WinNT provider. Although it does not provide many of the details about objects that LDAP does, this method is easier to use. Its syntax does not require you to deal with the hierarchical structure of AD, and it can be used to access domain-based SAM databases (residing on NT 4.0 domain controllers), local-based SAM databases (for NT 4.0 and Win2K member servers and workstations), and Win2K domain controllers. In the example that Listing 7.5 shows, KiXtart is used to create a list of the machines found in the current domain's SAM database (in KiXtart, @DOMAIN is used to represent the current domain name).

```
$rc = Open (1, "%TEMP%\Machines.txt", 5)
 $Domain = GetObject ('WinNT://' + @DOMAIN + ',domain')

 For Each $Computer In $Domain
     If $Computer.class = "Computer"
        $rc = WriteLine (1, $Computer.name + @CRLF)
     EndIf
 Next
 $rc = Close (1)
```

Listing 7.5: Using KiXtart to create a list of machines found in the current domain's SAM database.

In the open statement, the number one is a handle used to identify this file later in the script. The number five tells KiXtart to create the file if it does not already exist and to open the file for writing. If the class property identifies the object as a computer, it is written to the file opened earlier as 1, and includes the macro variable @CRLF, which identifies a carriage return and line feed. Once each item in the $Domain object is processed, the machines.txt file is closed.

NET VIEW

Using the NET VIEW command, you can produce a list of computers from the network browse list. The network browse list provides a list of machines, so the list is easily gathered; however, this method is not always the most reliable because (by default) the network browse list only contains systems that are (or have recently been) active on the network.

The script that Listing 7.6 shows uses the greater than symbol (>) to direct the output of the NET VIEW command to a temporary file that is used by the script to create a clean list of computers. Because the output of the NET VIEW command contains a header and computer comments, the script uses the InStr and SubStr functions from the third character until the first space is encountered on lines that start with \\. The computer names are written to the file opened as 2, and when complete, file 2 is closed. Finally, the temporary file containing the NET VIEW output is also closed, then deleted.

```
Shell "%comspec% /c net view /domain:" + @DOMAIN + "
>%temp%\netview.tmp"
$rc = Open (1, "%temp%\netview.tmp")
$rc = Open (2, "%temp%\machines.txt", 5)
$line = ReadLine (1)
While @error = 0
      If InStr ($line, "\\")
            $rc = WriteLine (2, RTrim (SubStr ($line, 3, InStr ($line,
" ") - 1)) + @CRLF)
      EndIf
      $line = ReadLine (1)
Loop
$rc = Close (2)
$rc = Close (1)
Del "%temp%\netview.tmp"
```

Listing 7.6: Using NET VIEW to create a machine list.

Custom Script Examples

You can script most anything; the following sections provide examples of scripts that are used to accomplish common administrative tasks. In the following sections, we will make use of the computer name lists generated to restart machines, backup their event logs, and gather software inventory information.

Restarting Remote Systems

Restarting machines is an action that is becoming increasingly popular to script as a result of Group Policy. With Group Policy, when new software is assigned to a machine, a reboot is necessary in order to trigger the installation process. In the example that Listing 7.7 shows, the script reads the machines.txt file, and for each machine included, issues a shutdown command. The syntax for the KiXtart shutdown command is to include the name of the machine to be shut down, the message to be displayed, and the number of seconds to count down before shutdown begins. The last two values specify whether the user should be forced off and whether the system should reboot. In the fourth parameter, a zero indicates that if the user has unsaved work, the user should be prompted to save before the script logs off the user (if it were set to one, the user would be forcibly logged off without prompting). The fifth and final parameter indicates the system should be restarted (if it were set to zero, the system would simply be shut down). The ReadLine function reads a line from the specified script with each use. The script loops until this function tries to read a line and fails (when the end of the list is encountered).

```
$rc = Open (1, "%temp%\machines.txt")
$Machine = ReadLine (1)
While @ERROR = 0
$rc = Shutdown ($Machine, "Restarting To Install New Software", 30, 0,
1)
      $Machine = ReadLine (1)
Loop
$rc = Close (1)
```

Listing 7.7: Using a KiXtart script to restart remote systems.

Backing Up Event Logs

Managing event logs is yet another task best left to a script or other automated process. The script that Listing 7.8 shows reads each machine name from the machines.txt file and uses the BackupEventLog function to create a backup of the security log on a server share. If the task was successful, it will return an error level of zero (in KiXtart, you learn the returned error by looking at the @Error macro variable). If the variable is zero, we use the ClearEventLog function to clear the remote system's security log and move on to the next machine. When the end of the machine list is reached, attempting to read the next computer name will result in an error that will exit the loop, close the machines.txt file, and exit.

```
$rc = Open (1, "%temp%\machines.txt")
  $Machine = ReadLine (1)
  While @ERROR = 0
      $rc = BackupEventLog ("\\" + $Machine +
"\Security","\\server\share\Application_Log_" + @WKSTA)
      If @ERROR = 0
            $rc = ClearEventLog ("\\" + $Machine + "\Security")
      EndIf
      ? $Machine + " - " + @SERROR
      $Machine = ReadLine (1)
  Loop
  $rc = Close (1)
```

Listing 7.8: A script that backs up event logs.

Software Inventory

Software inventory is another task that can be performed a number of ways. We'll explore two methods here—reading the Uninstall key from the registry and querying the Windows Installer object with WMI.

Reading the Registry

The data presented in the Add/Remove Programs applet provides information about most every program installed on a system. Applications that do not provide an uninstall, such as simple command-line utilities, will not be detected; however, most applications that provide an installation program register themselves in the Uninstall subkey in the registry. Usually, each application provides its name and the command that is executed for removal (see Figure 7.3).

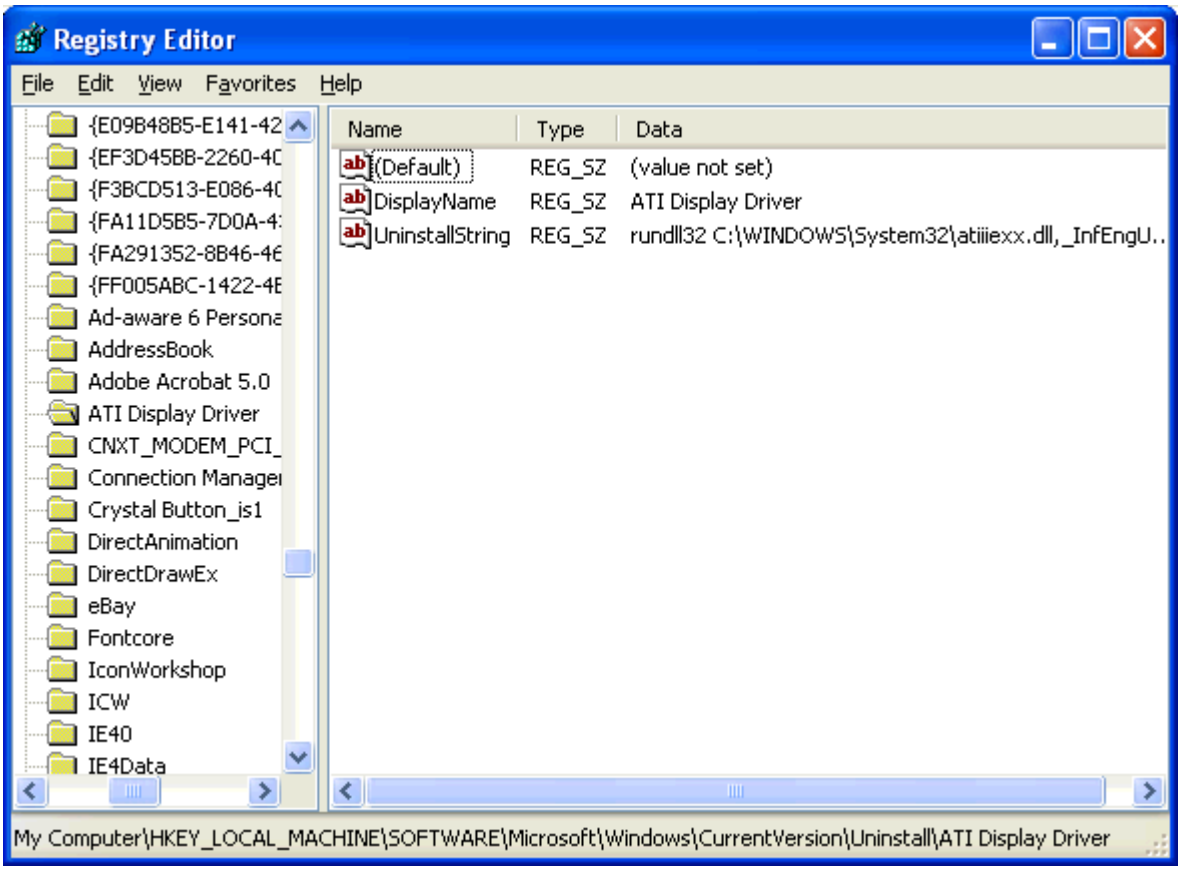

Figure 7.3: Uninstall subkey values contain a list of software installed on a system.

Although MSI setups provide a fixed set of values in the Uninstall subkey, those of non-MSI setups can be very inconsistent. In the example that Listing 7.9 shows, the script enumerates each of the values in the remote system's Uninstall key. Because we cannot count on a consistent value to identify the application, the script looks for DisplayName and then QuietDisplayName to identify the application. If neither are found, the name of its subkey within Uninstall is used.

```
$rc = Open (1, "%Temp%\Machines.txt")
$rc = Open (2, "%Temp%\Software.txt", 5)
$Machine = ReadLine (1)
While @ERROR = 0
      $rc = WriteLine (2, $machine + @CRLF)
      $UninstallSubKey = "\\" + $MACHINE +
"\HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall"
      $Count = 0
      $Key = EnumKey ($UninstallSubKey, $Count)
      While @Error = 0
            $App = ReadValue ($UninstallSubKey + "\$KEY",
"DisplayName")
            If $App = ""
                  $App = ReadValue ($UninstallSubKey + "\$KEY",
"QuietDisplayName")
            EndIf
            If $App = ""
                  $App = $Key
            EndIf
            $rc = WriteLine (2, $App + @CRLF)
            $Count = $Count + 1 $Key = EnumKey ($UninstallSubKey,
$Count)
      Loop
      $rc = WriteLine (2, @CRLF)
      $Machine = ReadLine (1)
Loop
$rc = Close (2)
$rc = Close (1)
```

Listing 7.9: Reading the registry to determine the system's software inventory.

Windows Installer

WMI lets you query installed software. However, this method will only return applications installed with Windows Installer (MSI). In the example that Listing 7.10 shows, the script reads each machine name from machines.txt, and connects to the WMI object in order to query for installed software. Each application found is written to a text file.

```
$rc = Open (1, "%temp%\machines.txt")
$rc = Open (2, "%temp%\software.txt",5)
$Machine = ReadLine (1)
While @ERROR = 0
      $rc = WriteLine (2, $machine + @CRLF)
      $WMI = GetObject ("winmgmts:{impersonationLevel=impersonate}!\\"
+ $Machine + "\root\cimv2")
      $Software = $WMI.ExecQuery("Select * from Win32_Product")
      For Each $Application In $Software
            $rc = WriteLine (2, $Application.Name + @CRLF)
      Next
      $rc = WriteLine (2, @CRLF)
      $Machine = ReadLine (1)
Loop
$rc = Close (2)
$rc = Close (1)
```

Listing 7.10: Using WMI to inventory a system's software.

The AutoLogon Process

AutoLogon provides you with the ability to have a machine automatically log on as a user so that you can interact with the desktop in a traditional fashion with your script. Some installations require that an administrator log on following the initial portion of the setup so that actions can take place following a reboot. It is also common to follow the imaging or unattended Windows installation process with a script that will perform additional installations and configuration changes.

There are several tools available to utilize an alternative set of credentials so that users triggering a process (such as through a logon script) can get past normal security restrictions. This method of utilizing the AutoLogon process is best suited for cases in which you cannot repackage or utilize silent switches to automate a task. In these situations, you might find it necessary to address prompts and dialog boxes with your script, which will require that a desktop profile be loaded. You can do so using the Windows resource kit ScriptIt tool or KiXtart's built-in SendKeys function. Both options require that dialog boxes be addressed by name, which requires that the dialog boxes exist on the desktop so that they may be addressed.

NT and later provide this capability by setting the following keys in the HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon subkey:

- AutoAdminLogon
- DefaultPassword
- DefaultUsername
- DefaultDomainName
- AutoLogonCount (Win2K and later)

Set AutoAdminLogon to 1 to instruct the system to attempt an automatic logon. The DefaultPassword, DefaultUsername, and DefaultDomainName values should be updated to include the account information with which you want to automatically log on. The downside is that the values are all in plain text and can be read by anyone who knows where to look. Thus, when you're completing your desired AutoLogon sessions, it is important to not only set the AutoAdminLogon value to zero but also delete the logon credentials stored in the other values. Win2K and later provide a useful feature to handle this cleanup for you—simply set the AutoLogonCount value to the number of AutoLogon sessions you want to initiate, and this value will be decremented with each automatic logon performed. When it reaches zero, it cleans up the logon credentials for you, and resets the AutoAdminLogon value back to zero.

In the example that Listing 7.11 shows, the system is configured to logon automatically two times with the AutoLogonCount value. The computer will attempt to log on to the appdeploy domain with a username of bkelly and a password of 12345678. Then the machine is restarted to begin the AutoLogon process.

```
$WinLogonSubKey = "HKLM\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Winlogon"
$rc = WriteValue ($WinLogonSubKey,"AutoLogon","1","REG_SZ")
$rc = WriteValue ($WinLogonSubKey,"DefaultUserName","bkelly","REG_SZ")
$rc = WriteValue
($WinLogonSubKey,"DefaultPassword","12345678","REG_SZ")
$rc = WriteValue
($WinLogonSubKey,"DefaultDomainName","appdeploy","REG_SZ")
$rc = WriteValue ($WinLogonSubKey,"AutoLogonCount","2","REG_SZ")
$rc = Shutdown("","Shutting Down For AutoLogon",10,1,1)
```

Listing 7.11: Scripting the AutoLogon process.

Restricting Access

Odds are that if you are having a script perform an automatic logon to the system, it is because you need to have the computer logged on as an administrator. To say the least, this is not likely something your security department will look kindly on without some measure taken to limit a user from interrupting the script and accessing the system with the administrative account. Locking the mouse and keyboard is a typical approach to addressing this issue. By disabling the mouse and keyboard devices, you can perform an automatic logon without worry of users interrupting the script.

Although you cannot stop the mouse or keyboard devices during a current Windows session, you can disable the devices so that they do not start the next time the system reboots. The device settings are located in the HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services subkey. The mouse device value is MouClass and the keyboard device value is KbdClass.

As the AutoLogon process requires a reboot already, disabling these two devices before the restart fits nicely into the process. However, in order to restore the system, you will need to undo the changes when you are done. The script that Listing 7.12 shows configures a computer for automatic logon and disables the mouse and keyboard drivers for next logon (by setting their start values to 4). Before restarting the computer, a RunOnce entry is added to the system to perform actions and unlock the system in order to control the AutoLogon process.

```
$WinLogonSubKey = "HKLM\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Winlogon"
$rc = WriteValue ($WinLogonSubKey,"AutoLogon","1","REG_SZ")
$rc = WriteValue ($WinLogonSubKey,"DefaultUserName","bkelly","REG_SZ")
$rc = WriteValue
($WinLogonSubKey,"DefaultPassword","12345678","REG_SZ")
$rc = WriteValue
($WinLogonSubKey,"DefaultDomainName","appdeploy","REG_SZ")
$ServicesSubKey = "HKLM\SYSTEM\CurrentControlSet\Services"
$rc = WriteValue($ServicesKey + "\MouClass","Start","4","REG_SZ")
$rc = WriteValue($ServicesKey + "\KbdClass","Start","4","REG_SZ")
$RunOnceSubKey =
"HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce"
$rc =
WriteValue($RunOnceSubKey,"AutoScript.bat","c:\temp\AutoScript.bat","RE
G_SZ")
$rc = Shutdown("","Shutting Down to Run AutoScript.bat as
Admin",10,1,1)
```

Listing 7.12: Locking the mouse and keyboard and restarting for AutoLogon.

Listing 7.13 shows an example script that lets you restore the system. You would precede this code with whichever actions you want to automate as administrator. The batch file that was added to RunOnce in the previous example would call the following script to restore control of the system.

```
$WinLogonSubKey = "HKLM\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Winlogon"
$rc = WriteValue ($WinLogonSubKey,"AutoLogon","0","REG_SZ")
$rc = WriteValue ($WinLogonSubKey,"DefaultUserName","")
$rc = WriteValue ($WinLogonSubKey,"DefaultDomainName","","REG_SZ")
$rc = DelValue ($WinLogonSubKey,"DefaultPassword")
$ServicesSubKey = "HKLM\SYSTEM\CurrentControlSet\Services"
$rc = WriteValue($ServicesKey + "\MouClass","Start","1","REG_SZ")
$rc = WriteValue($ServicesKey + "\KbdClass","Start","1","REG_SZ")
$rc = Shutdown("","Shutting Down to Restore System Control",10,1,1)
```

Listing 7.13: Unlocking the mouse and keyboard and restarting to restore control.

Summary

In this chapter, we covered some of the benefits and drawbacks to scripting as well as some tips about how to choose a scripting language. We discussed COM automation, and the ways you can use it to interact with directory services, databases, and applications. Some common ways to run your scripts were covered along with the order and security context for each. Finally, we went over some real-world implementations, including how to generate a list of machines on a network and ways to then use these lists to reboot computers, manage event logs, and gather software inventory. In the next and final chapter, we will discuss asset management—both software and hardware, including the benefits and tools available to do so.

Chapter 8: Asset Management

Asset management is the tracking of technology assets including hardware and software: knowing where they are, who is using them, and how they are being maintained. Asset management is made up of a combination of policies and procedures that provide you with an invaluable view of your desktop systems. Depending upon your environment, asset management may include any or all of the following topics, which we will cover in this chapter:

- Inventory—Managing and tracking the location and ownership of physical technology assets
- Software licensing—Tracking and managing software licenses
- Change control—Administering the changes to your managed configuration

Inventory

You may tend to think primarily of software inventory, but you also need to consider physical assets such as computers and their associated components (hard drives, scanners, printers, and so on). In fact, some organizations even include assets such as desks, chairs, phones and other office equipment in the inventory category. Tracking such things may seem irrelevant, and to you, the desktop administrator, they probably are. However, when establishing an inventory system, you should keep in mind that it may be utilized as more than a tool just for the IT department.

Depending upon your environment, it is often valuable to track additional information as it relates to your physical assets including:

- Agreements
- Leases
- Contracts
- Warrantee information
- Maintenance schedules
- Depreciation and other costs, such as repair

Benefits of Maintaining an Updated Hardware Inventory

Particularly when dealing with software and hardware information, an accurate and up-to-date inventory can help desktop administration efforts in providing valuable data for planning upgrades. Additionally, loss and theft can be more easily identified and reported. When determining which systems would most benefit from replacement and which systems meet certain minimum requirements, accurate computer inventory data is a critical tool. The following sections highlight the benefits of maintaining an update inventory.

Ensuring Minimum System Requirements Are Met

Every Windows software application provides a minimum and recommended hardware configuration. Outside high-end imaging products and games, it is commonplace to see minimum requirements far below what you would consider a worthy network client. However, unless you have established a workstation lifecycle that rotates old hardware off your network, you may very well find you have client systems in need of software that their hardware cannot adequately support.

In most environments, it is the upgrade to a new version of Windows that dictates the most spiked increase in minimum hardware requirements. Having the ability to quickly identify systems that require updates is a vital step in executing a smooth migration.

Minimum system requirements dictate the software or OS vendors' lowest possible configuration recommendation. Oftentimes these minimum requirements are not enforced, so it is possible for you to perform the installation on lesser systems. Aside from poor performance as a result of such installations, if failures occur, you are not likely to get much help from the product's support department.

Recommended system requirements are usually listed along with the minimum requirements. Whenever possible, it is these recommended requirements that should be your goal to meet (or, hopefully, exceed). As Table 8.1 illustrates, Windows' system requirements have become increasingly demanding with each release.

OS	Processor	RAM	Hard Drive	Video
Windows 95	386DX (486)	4MB (8MB)	55MB	VGA (SVGA)
Windows 98	486/66 (Pentium)	16MB (24MB)	355MB	VGA (SVGA)
Windows NT	Pentium	16MB (32MB)	110MB	VGA (SVGA)
Windows ME	Pentium 150	32MB	320MB	VGA (SVGA)
Windows 2000	Pentium 133	64MB	2GB	VGA (SVGA)
Windows XP	Pentium 233 (300)	64MB (128MB)	1.5GB	SVGA

Table 8.1: Windows minimum requirements (recommended requirements are listed in parenthesis).

The Windows Hardware Compatibility List (HCL) contains a list of computer systems and peripherals that have been tested and deemed compatible with Windows. The HCL is often overlooked until problems arise. Particularly if you have added components or peripherals to your systems, take the time to ensure that there are not compatibility issues prior to rolling out a new version of Windows. Compatibility issues have become such a matter of contention since the release of Win2K that Microsoft now provides a database of not only hardware, but also computer brands and software that have passed compatibility testing. You may search the database online at http://www.microsoft.com/whdc/hcl/default.mspx.

Because of the Advanced Computer Power Interface (ACPI) capabilities introduced with Win2K, a BIOS update may be required to ensure stability. Failure to install recommended BIOS updates may result in system hardware errors. For more information, see http://www.microsoft.com/windows2000/server/howtobuy/upgrading/compat/biosissue.asp.

A hardware compatibility test kit is provided by Microsoft for hardware manufacturers along with a submission tool and procedures at http://www.microsoft.com/whdc/hwtest/default.mspx.

Knowing Where to Replace Hardware

You have just learned of a new shipment of computers coming in—who will get them? Twenty new machines may mean many more than 20 users receive new computers—many organizations will replace a mid-grade system with a high-grade system, then take that mid-grade system and replace a low-grade system. Often referred to as *cascading*, more users benefit from new systems utilizing this practice. Without sufficient hardware inventory data, it is difficult to identify proper candidates for cascading.

One situation in which the replacement of systems may become routine is for companies that choose to lease computers for a 2- or 3-year period. This option might less expensive than the regular purchase of new systems, and (if it is deemed acceptable) the preinstalled OS and software that comes on a new system could reduce administrative efforts in their deployment.

What to Do with Those Old Computers?

Donating or recycling your old computers and peripherals can not only help the receiving organization, but will also provide your own organization a tax break. When donating, be sure to remove all personal files from the hard drive before you offer a computer for donation. You should also obtain a receipt to document the transfer of ownership. An example where this may become of value is if a computer with a battery is discovered in the trash (which is illegal) you can show that you are not longer responsible for it.

Ensure that you are comfortable with the organization receiving your donation. Ask them to provide proof of nonprofit status before you make your donation. Any legal nonprofit organization or school should be able to provide a letter of request on their letterhead. They should also be able to provide you with copy of their state or federal nonprofit organization registration form.

Share the Technology is an all-volunteer run non-profit organization with a mission is to help salvage recently retired computers while they are still useful, prevent their premature destruction, and get them into the hands of schools, people with disabilities, and nonprofit organizations that can put them to good use. Visit the Share the Technology Web site at http://sharetechnology.org.

Enhancing System Components

Where budgets are limited, it is often beneficial to purchase components such as larger hard drives, additional memory, and even faster processors to replace existing components that may require updating. Unless all the systems in your organization are identical, you will benefit greatly from inventory data to accurately identify which components of which systems should be upgraded. Additionally, inventory data will need to be updated or generated in order to account for the location of these new components in your organization.

Improving Help Desk Productivity

Gartner Group estimates that as much as 50 percent of the time spent on a Help desk call is used to determine the system's hardware and software configuration. Inventory reports could be used to provide Help desk technicians a detailed report about the troubled PCs configuration quickly.

The more information you have about what's happening on your user's desktops, the more efficiently you can resolve the user's problem. You can speed the troubleshooting process by having the details of the problem system available to you without having to ask the user or examine the system. Additionally, by not having to ask questions, user confidence may also be increased.

If the system reporting trouble has non-standard components and you are aware of these components as a result of an updated inventory, you might be able to look first at the non-standard component as a way to help you to identify a potential cause for the problem. Further, if you identify a problem with a certain device, you can proactively check on other systems that may be at risk. In addition, tracking changes often helps you identify a problem that begins occurring as result of a hardware or software change.

Deterring Loss and Theft

It is impossible to identify and replace lost or stolen hardware without first having some method of identifying the loss. Just the presence of a hardware inventory system may be a considerable deterrent in preventing theft by employees.

Serial Number Tracking

An automated hardware inventory system may have the ability to contain information that would be difficult or impossible to report on manually. Device serial numbers provide you with the added knowledge of not just which systems have which devices, but exactly which devices. For example, if your inventory reports include RAM serial numbers, you can identify its movement if someone were to take it from one machine and install it in another.

Hardware Inventory Methods

There are many ways to obtain hardware inventory data from a system. In particular, we will discuss the capabilities of WMI and the built-in WinMSD utility to report this information.

In Chapter 7, we discussed the ability of WMI to provide comprehensive system data. Among this retrievable information is a wealth of inventory data; the challenge lies in identifying and pulling meaningful information and displaying it in a user-friendly fashion. Most inventory tools for Windows do just that—collect and provide a clear view of data pulled from WMI. Some key collections you may reference in obtaining WMI data include:

- Win32_SCSIController—SCSI disk controller information
- Win32_IDEController—IDE disk controller information
- Win32_PhysicalMedia—Physical media information
- Win32_LogicalDisk—Logical disk information
- Win32_DiskDrive—Disk drive information
- Win32_Processor—Processor information
- Win32_NetworkAdapter—NIC information
- Win32_DesktopMonitor—Monitor information
- Win32_VideoController—Video adapter information

- Win32_BaseBoard—Motherboard information
- Win32_BIOS—BIOS information

> Although many inventory tools today take advantage of WMI to report hardware inventory data, other tools exercise their own proprietary methods to report such information. Later in this chapter, I'll provide a list of several vendors with tools that provide hardware inventory as a feature.

Listing 8.1 shows an example KiXtart script that will display the details of each installed network interface card (NIC). For an example of the output this script would generate, see Figure 8.1.

```
Break On
$Computer = "."
$WMI = GetObject("winmgmts:\\" + $Computer + "\root\cimv2")
$collection = $WMI.ExecQuery("Select * from Win32_NetworkAdapter",,48)
For Each $item In $collection
 ? "Adapter Type: " + $item.AdapterType
 ? "Auto Sense: " + $item.AutoSense
 ? "Availability: " + $item.Availability
 ? "Caption: " + $item.Caption
 ? "Config Manager Error Code: " + $item.ConfigManagerErrorCode
 ? "Config Manager User Config: " + $item.ConfigManagerUserConfig
 ? "Creation Class Name: " + $item.CreationClassName
 ? "Description: " + $item.Description
 ? "Device ID: " + $item.DeviceID
 ? "Error Cleared: " + $item.ErrorCleared
 ? "Error Description: " + $item.ErrorDescription
 ? "Index: " + $item.Index
 ? "Install Date: " + $item.InstallDate
 ? "Installed: " + $item.Installed
 ? "Last Error Code: " + $item.LastErrorCode
 ? "MAC Address: " + $item.MACAddress
 ? "Manufacturer: " + $item.Manufacturer
 ? "Max Number Controlled: " + $item.MaxNumberControlled
 ? "Max Speed: " + $item.MaxSpeed
 ? "Name: " + $item.Name
 ? "Network Addresses: " + $item.NetworkAddresses
 ? "Permanent Address: " + $item.PermanentAddress
 ? "PNP Device ID: " + $item.PNPDeviceID
 ? "Power Management: " + $item.PowerManagementCapabilities
 ? "Power Management: " + $item.PowerManagementSupported
 ? "Product Name: " + $item.ProductName
 ? "Service Name: " + $item.ServiceName
 ? "Speed: " + $item.Speed
 ? "Status: " + $item.Status
 ? "Status Info: " + $item.StatusInfo
 ? "System Creation ClassName: " + $item.SystemCreationClassName
 ? "System Name: " + $item.SystemName
 ? "Time Of Last Reset: " + $item.TimeOfLastReset
 ?
Next
```

Listing 8.1: Example KiXtart script to display the details of each installed NIC.

```
C:\WINNT\system32\cmd.exe

Adapter Type: Ethernet 802.3
Auto Sense:
Availability: 3
Caption: [00000005] Intel(R) PRO/100 VM Network Connection
Config Manager Error Code: 0
Config Manager User Config: 0
Creation Class Name: Win32_NetworkAdapter
Description: Intel(R) PRO/100 VM Network Connection
Device ID: 5
Error Cleared:
Error Description:
Index: 5
Install Date:
Installed: -1
Last Error Code:
MAC Address: 00:08:02:19:1B:38
Manufacturer: Intel
Max Number Controlled: 0
Max Speed:
Name: Intel(R) PRO/100 VM Network Connection
Network Addresses:
Permanent Address:
PNP Device ID: PCI\VEN_8086&DEV_2449&SUBSYS_00120E11&REV_03\4&11CD5334&0&40F0
Power Management:
Power Management: 0
Product Name: Intel(R) PRO/100 VM Network Connection
Service Name: E100B
Speed:
Status:
Status Info:
System Creation ClassName: Win32_ComputerSystem
System Name: PACKAGEBOX1
Time Of Last Reset: 20030715043414.000000-240
```

Figure 8.1: Example of network information script output using WMI.

Another way to gather system details in an automated manner is to utilize the WinMSD tool. WinMSD has been updated through the past few versions of Windows. For NT, WinMSD can specify the UNC path to a system and utilize any of the following command-line switches:

- /a—All details
- /s—Summary details only
- /f—Send output to a file <computername.txt> in the current directory
- /p—Send output to a printer

Run alone, NT's WinMSD will launch in GUI mode, as Figure 8.2 shows.

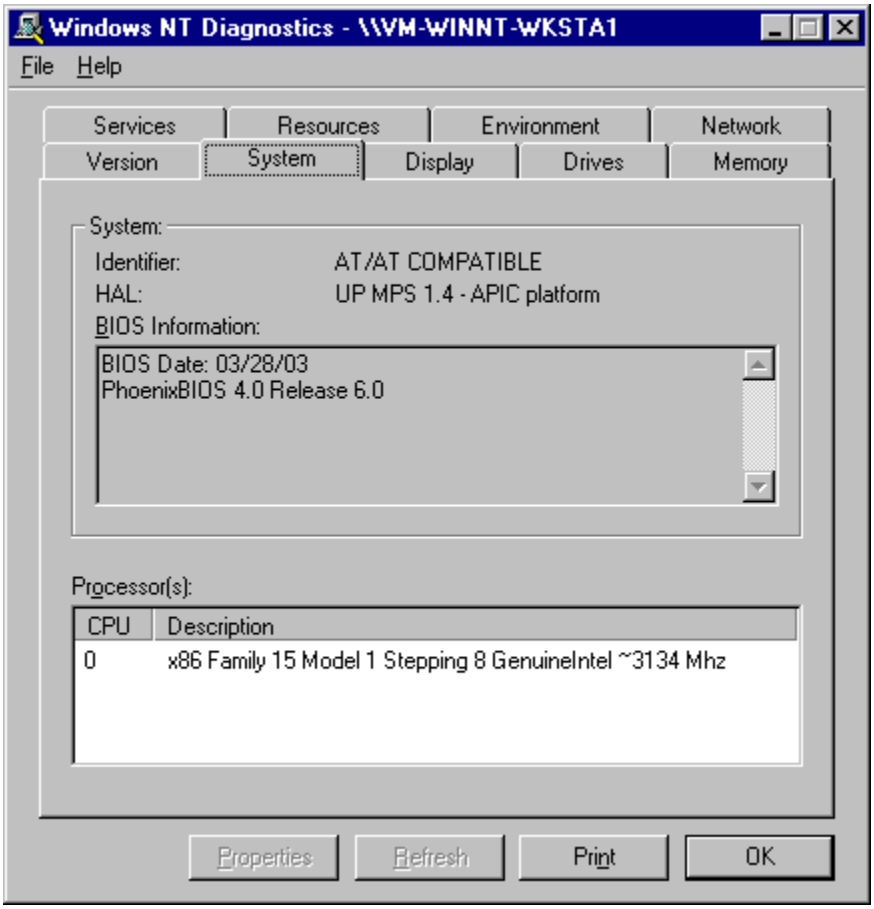

Figure 8.2: WinMSD for NT in GUI mode.

The Win2K incarnation of WinMSD (msinfo32) also provides a great deal of detail about the specified system. You can use the following command-line parameters to specify specific sections to output:

- /msinfo_file is used to open a specific NFO or CAB file
- /nfo or /s will send output to a specific NFO files
- /report directs text-formatted output to a specific file
- /computer connects and displays information regarding a specific remote system
- /categories displays the specified categories (used to limit output)
- /category sets the focus of the GUI display to the specified category at startup

> An NFO file is System Information File, which is a binary file that may be displayed using the System Information tool (msinfo32.exe) as a viewer.

You can use the following command to return all available information, which generates a detailed report in a text file named inventory.txt:

```
WinMSD.exe /report c:\inventory.txt
```

Figure 8.3 shows the Windows XP WinMSD tool interface.

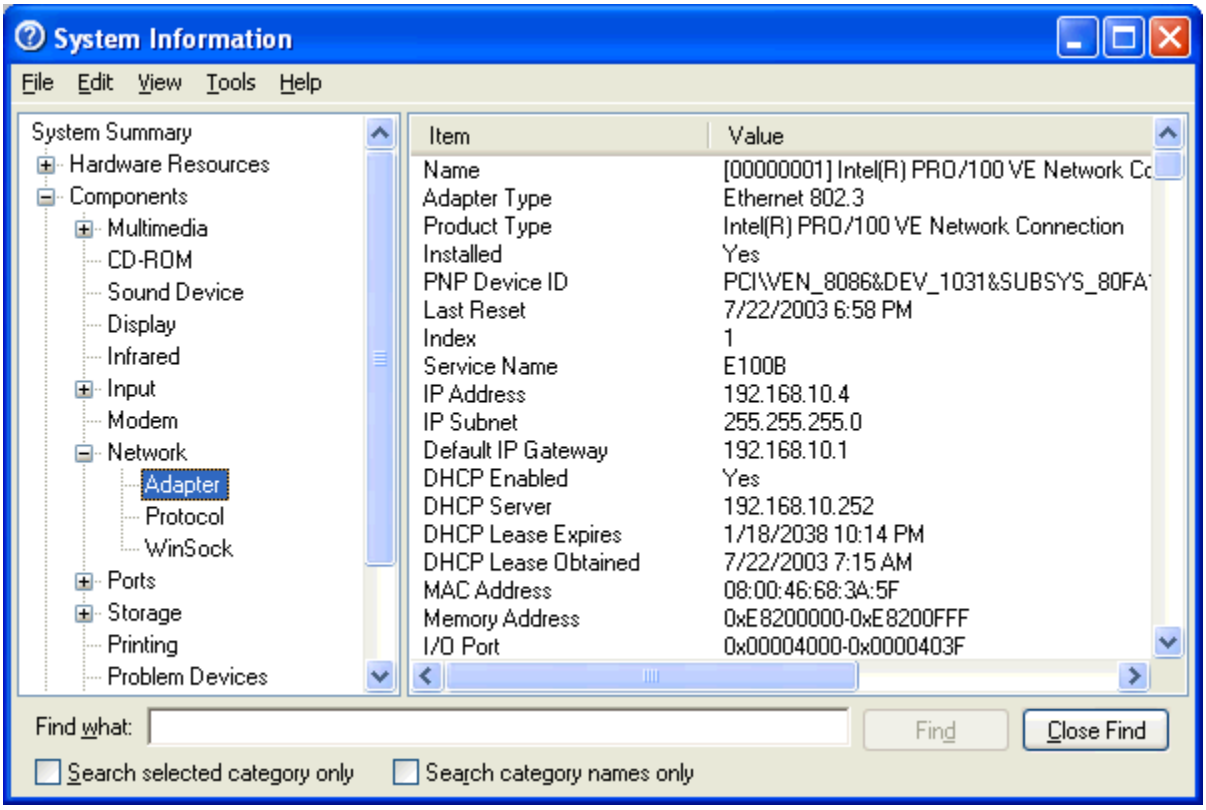

Figure 8.3: Windows XP WinMSD (msinfo32.exe).

> ☞ You can use the new msinfo32.exe filename or the older winMSD.exe filename to initiate this tool on Win2K and Windows XP systems. However, on Windows XP, you must call MSInfo32 directly to utilize the supported command-line switches.

> When automating WinMSD from the command line, be aware that if the Server Service is not running, WinMSD will show a warning dialog box.

Software Licensing

When many think of inventory and asset management, software inventory immediately comes to mind. Do you have enough licenses? Do you have too many? What software do you have installed throughout your network? We will cover these and other topics as they relate to software licensing in this section.

Ensuring License Compliance

Current legislature supports software abuse policies and heavy fines are now being imposed upon companies not complaint with software licenses. Failure to maintain a sufficient number of licenses on your network can be a costly situation.

More and more employees are utilizing applications on desktops and laptops. Take into consideration the number of licenses purchased, various software versions operating within your company, and the types of licenses you own for each application. With these and other variables, software license compliance tracking can be a challenge—even with the right tools. Operating without such tools can make license management a nearly impossible, and certainly time-consuming, task.

License Types

In addition to the many ways licenses may be enforced, there is also the matter of how licenses are to be allocated—some are to be tracked per user, others are to be tracked per seat (system), there are licenses intended for a limited number of active users, and finally some cover an entire organization.

- Per user—Licensing tracked per user or per developer means that you must have a license for each individual who makes use of the software. Per user licensing normally allows for multiple installations so long as the number of individuals that use the software is accounted for.

> ☞ Software licensed per user normally records licensing information in the user profile. In this way, the user will be able to use the software in more than one location, but other users attempting to use the software will be prompted for registration information and/or denied access if their user profiles do not contain this valid licensing data. Although the licensing information is sometimes stored in a file, it is more commonly found in the registry under the vendor's software key (for example, in the HKEY_CURRENT_USER\Software\<Vendor>\<AppTitle> key).

- Per seat—Per seat licensing is typically reserved for client/server applications for which many clients throughout a network have a client-side service or agent installed. Per seat commonly means per system; thus, regardless of the number of users on the network, each computer that executes the application must have its own license.

  Per seat licensing is usually enforced by the server-side component of the software, such as with many desktop management systems. The server counts the number of client systems that it has worked with and tracks this number against the number of purchased licenses. It is up to the vendor as to how license violations are to be handled; some will provide warnings, while others will cease operation (at least on any additional systems) until the number of per-seat licenses is appropriately increased.

> Particularly for server software products, an increasingly common licensing model is per processor, with which licenses are required for each CPU in the server.

- Concurrent usage—Concurrent licensing refers to the number of active instances at any one time. In this scenario, you may have only a dozen licenses for a particular application though you have hundreds of installed systems and potential users. Not until the thirteenth user of the software attempts to initiate the application do you have a license violation.

  Concurrent usage is often tracked by organizations using software metering tools. Metering tools record the times a software product is launched and exited. Once the maximum available licenses are in use, some metering products will even go so far as to halt a license violation from taking place by denying further execution of the application until other instances on the network are closed.

- Site licenses—A site, or enterprise, license indicates that an entire company, domain, or location can utilize the software without concern for the number of systems, users, or its concurrent usage. Although the most expensive, such a license is also the easiest to administer.

Licensing Methods

There are several ways that software vendors dictate licensing requirements for their software. As we will cover in this section, these methods may include the implementation of license management servers, registration codes, and files.

- License servers—A license server is usually used to track concurrent usage of a product. Per seat licensing is often tracked by the application itself, whereas a license server will actually do the job of recording the number of instances currently in use. When the number of purchased licenses is exceeded, further attempts to launch the application are typically met with a warning dialog box that instructs the user of the problem, then exits the application. Such being the case, you must either wait for licenses to become available (as others close the application) or purchase additional licenses to increase the number of allowable instances.

- License files—A typical license file implementation is in the format of a binary file provided when a purchase is made. The license file may include information such as the number of per seat licenses, concurrent licenses, company name, expiration date, and product support or upgrade coverage. The application provides a method for accepting this license file where the details of your purchase may be processed by the application in question.

- Serial/Registration strings—The most common implementation of software registration is for applications to request a serial number, registration, or product code during installation or upon first launch of the software. Many applications are usable in a limited configuration until such registration information is provided to “unlock” the application and allow full use of what the program has to offer. This setup provides vendors the ability to let you try the software in a limited capacity and/or for a certain period of time. Further, it saves you from the need to redeploy the software if the evaluation results in the purchase of licenses.

> ✎ Although some licensing implementations use information—such as the network card MAC address or other unique identifier—to pair a provided license code with a specific system, the same license code may often be used on multiple systems. Is this okay? To know for certain, you would need to ask each software vendor that you deal with. However, it is often considered to be acceptable to deploy software using the same registration information as long as you have the proper number of licenses in your possession.

Buying Only What You Use

Having a clear picture of what you have and what you are using can save a considerable amount of money when it comes time to purchase upgrades or extend support offerings. If software is deployed to all systems, and you are able to determine that only 10 percent of the users ever utilize it, renew the annual support contract for only those systems that require it. How do you know what is being used? Through software metering.

Software Metering

Software metering measures concurrent usage of applications on your network. This measurement may be limited to certain managed applications or may include all software across the enterprise. At its most basic implementation, metering software measures when the software is launched and when it is closed. This information alone can be very valuable when presented in concise reports. However, in addition to reporting, software metering solutions may also manage and enforce concurrent usage licensing limits. Simply collecting and reporting information about software usage is known as *passive metering*, license enforcement is referred to as *active metering*.

- Passive metering—More common to metering solutions is the offering of passive metering functionality. With this functionality, executions are monitored and reported upon. Some solutions may generate administrative alerts when licenses are exceeded; however, these alerts are usually not provided in real time. Usage data is often collected locally, then reported to a management server at a defined interval. When collated, or when reports are generated, any violations can be identified and licenses may then be purchased to remedy the situation.

> ☞ Depending upon the metering solution, usage data can simply be execution and exit times or it may actually track activity. The ability to see the difference can be helpful because people may simply leave a program minimized all day and not actually use it. Although knowing this behavior does not necessarily help with licensing, it does give you a more accurate picture of what is really being used in your environment. Among other things, you can use usage data to determine what training may be of value as well as assessing which software upgrades may be of the most value to your users.

- Active metering—License management servers are configured to track the execution of applications on the network. Although this tracking might be limited to certain applications, when launched, the client software checks with the licensing server to determine whether there are available licenses. If there are, the application is allowed to execute and the number of available licenses is decremented. When the number of configured licenses is exceeded, the metering solution may notify the user to try again later or may even provide the option to be notified when a license becomes available.

Naturally, the process of checking a license server for available licenses prior to allowing or disallowing execution can be a performance hit. In fact, there are a shrinking number of products providing this capability as a result of this performance hit. However, some implantations may be faster than others, and depending upon your network capabilities and licensing needs, this may be a suitable solution.

Although software metering might seem like an excellent solution to licensing problems, you would be wise to investigate your software license agreements to determine whether this is acceptable. If per seat licensing has been dictated by the vendor, it may be unacceptable to implement your own concurrent usage management.

Validating Your Environment

Suppose you have deployed a new software package to 300 systems. How many copies are out there? If there is less, you can use the information to help investigate possible failures. If there are more, you can use the information to help investigate unauthorized installations (or installations performed outside your deployment process).

Although many deployment solutions provide a means of reporting successes and failures, many do a questionable job of it. Is it installed because the command line used to perform the silent installation did not return an error? Is it installed because the shortcut is now there or because some other file you previously identified is now present? However you go about it, validating your deployment efforts using software inventory provides added assurance of your success.

Perhaps the software you are to deploy is an update to an existing application. If some do not have the previous version as you expected, your installation will need to take that condition into account. Further, there may be a very early version of the software on the network that you may not have otherwise accounted for. Examining software inventory reports for dependencies or prerequisites can help you to provide a deployment package catered to the conditions that exist within your unique environment—software inventory can show you what that unique environment really is.

Having established a process for software deployment, there will still be those that step outside that process. Either to side-step the approval process or to simply get what they want faster, manual installations can be a real concern. Comparing what people have been deployed to what they actually have can provide a picture of where these process violations may exist.

Software inventory can also tell you who is not playing by the rules. By comparing (manually or programmatically) reports that show what has been assigned to a machine via your change management process and reports that show what is actually installed, you may identify unmanaged software (for example, you can discover how many games are installed on your network).

Software Inventory Tools

There are several software inventory tools available that offer many features. Some features to watch out for include:

- Web reports—for reporting, web reports have quickly be come the preferred method for many organizations. With web reports, no client software is required to view them and they may be easily shared or utilized on an existing corporate intranet.
- Report access control—you may not want everyone to access these reports. Some inventory and usage tracking tools provide a means for users to see a report of their own system, but not that of others. When access is limited, learn how it is limited (group membership, account privileges, separate username and password, etc.)
- Software requirements—some solutions do not require a client application to be deployed. If it does require a client service, what deployment mechanisms are offered by the vendor?
- Assurance of licenses for key personnel—if the president of you company cannot launch his PowerPoint presentation, you do not want fingers pointed at you!

Table 8.2 provides more information about the features provided by some of the popular software inventory and metering products (this table is not an all-inclusive list of products).

Company	Product	Software	Hardware	Active Metering	Passive Metering
ABC Systems	LAN Licenser			X	X
Alloy Software	Network Inventory Navigator	X	X		X
Altiris	Client Management Suite	X	X		X
Executive Software	SiteKeeper	X	X		
Express Metrix	Express Software Manager	X	X	X	X
Globetrotter Software	SAM wrap	X		X	X*
iInventory	LAN auditor	X	X		
Integrity Software	SoftTrack	X		X	X
Microsoft	Systems Management Server	X	X	X	X
NetSupport	NetSupport TCO	X	X	X	X
Sassafras Software	KeyAuditor	X	X	X	X

Company	Product	Software	Hardware	Active Metering	Passive Metering
Scalable Software	Survey Suite	X	X		X
Software Innovations & CPS	SystemHound	X	X		
Tally Systems	TS.Census	X	X		X*
Tangram Enterprise Solutions	Asset Insight	X	X	X	X
Unipress Software	FootPrints Asset Management	X	X		

Table 8.2: Software inventory and metering products (* = requires optional add-in or module).

Software Inventory Methods

There are several methods that you can employ to determine which software is installed on a system. Through COM, the WindowsInstaller object can report on not only what is installed, but also the state of the installation itself. Listing 8.2 shows a KiXtart script that displays the installation state for each Windows Installer–managed application on the local system.

> For more information about the ProductState property of the WindowsInstaller object, check out http://msdn.microsoft.com/library/en-us/msi/setup/installer_productstate_property.asp.

```
$MSI = CreateObject("WindowsInstaller.Installer")
$Products = $MSI.Products
For Each $Product In $Products
 $StateVal = $MSI.ProductState($Product, "ProductName")
 Select
  Case $StateVal = 0  $StateTxt = "Broken"
  Case $StateVal = 1  $StateTxt = "Advertised or Removed"
  Case $StateVal = 2  $StateTxt = "Absent"
  Case $StateVal = 3  $StateTxt = "Local"
  Case $StateVal = 4  $StateTxt = "Source"
  Case $StateVal = 5  $StateTxt = "Default"
  Case $StateVal = -1 $StateTxt = "Unknown"
  Case $StateVal = -2 $StateTxt = "Invalid Argument"
  Case $StateVal = -4 $StateTxt = "Source Absent"
  Case $StateVal = -5 $StateTxt = "Incomplete"
  Case $StateVal = -6 $StateTxt = "Bad Configuration"
  Case $StateVal = -7 $StateTxt = "Not Used"
 EndSelect
 ? $MSI.ProductInfo($Product, "ProductName") + " is " + $StateTxt
Next
```

Listing 8.2: An example KiXtart script that displays the installation state for each Windows Installer–managed application on the local system.

Because the Windows Installer method will only report those applications managed by the Windows Installer service, scanning the target system for files is a more thorough means of determining which software is present. This scan can be a time-consuming process, but can be very accurate. Some inventory products scan the file system and compare it with a proprietary database that identifies the installed software. Other solutions enumerate the files of the OS and examine their properties for information (see Figure 8.4) and some may use a combination of these two methods.

Using KiXtart's GetFileVersion function, you can obtain the informational data for any specified file. When present, the following file properties can be returned. Keep in mind that all, some, or none of the file version information may be present for any given file. The following descriptions represent the intended use for each possible field; however, the developer is free to include any information desired. It is this same process that some software inventory products utilize to identify files:

- Comments—Returns any specified comments for the file. This property is usually used to provide additional information that can be displayed for diagnostic purposes.
- CompanyName—Returns the name of the company that produced the file.
- FileDescription—Returns a string used to describe the file. This string may be used to dynamically produce a description along with (or in place of) the associated file name, which can sometimes be difficult to interpret. Unfortunately, this field is not commonly used.
- FileVersion—Returns the version of the file.
- InternalName—Returns the file's internal name, if one exists. For example, this string might contain the module name for a Dynamic Link Library (DLL), a virtual device name for a Windows virtual device, or a device name for an MS-DOS device driver.
- Language—Returns the full English name for the file's language.
- LegalCopyright—Returns all listed copyright notices, trademarks, and registered trademarks that apply to the file. This property may include the full text of all notices, legal symbols, copyright dates, trademark numbers, and other copyright information.
- LegalTrademarks—Returns all trademarks and registered trademarks that apply to the file. This property may include the full text of all notices, legal symbols, trademark numbers, and other trademark information.
- OriginalFilename—Returns the original name of the file, not including a path. This property can be used to determine whether a file has been renamed from its original file name. If the file is specific to a non-FAT file system, this name might not be in the standard MS-DOS 8.3 naming format.
- PrivateBuild—Returns by whom, from where, and why this private version of the file was built.
- ProductName—Returns the name of the product with which this file is distributed.
- ProductVersion—Returns the version of the product with which this file is distributed.
- SpecialBuild—Returns how this version of the file differs from the normal version.

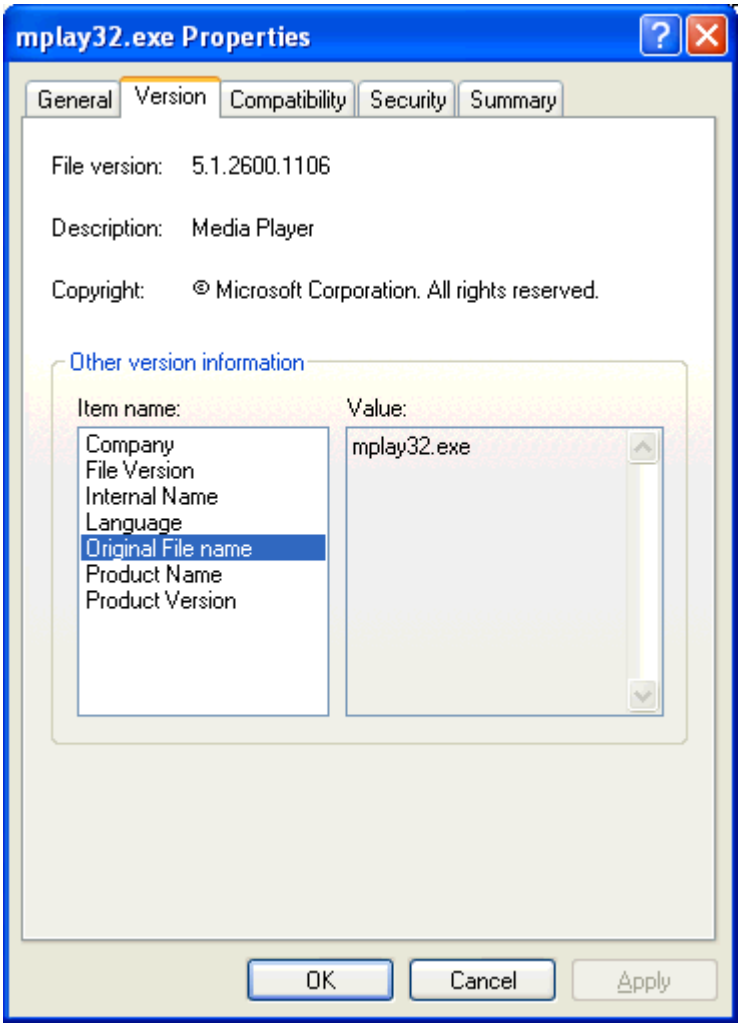

Figure 8.4: File properties displayed for Microsoft Media Player.

> ☞ By examining the OriginalFileName property, inventory solutions can identify files that may have been renamed by a user in order to hide them.

Another software inventory method is to use the Add/Remove Programs Control Panel applet, which lists the software installed on a machine that support an uninstall process. This list accounts for a great percentage of the software you are likely to have in your environment. By reading the values used by the Add/Remove Programs applet in the Uninstall key of the registry, you can obtain the names of each application installed on a system (and in many cases, much more).

> 🕮 See Chapter 7 for more information and sample scripts.

Change Management

In Chapter 2, we discussed establishing and deploying your initial baseline configuration. In Chapter 4, we discussed deploying software and updates to those systems. Remember that all the benefits of a baseline are quickly lost if you don't manage the changes that must occur throughout a system's *life cycle*. A system's life cycle covers its planned deployment, maintenance, and eventual replacement.

Suppose you have just rolled out the best documented, well-tested, and rock-solid baseline ever conceived. Thinking that it can stay that way very long is a big mistake. You must plan for the inevitable changes that will follow. First, let's discuss some of the reasons changes may be instigated in your environment:

- User requests—Aside from requests for new or upgraded software and the "My computer is slow, I need more RAM" request, consider that you may also need to account for evaluation software requests. Particularly with evaluations, it is important to have a plan for the inevitable removal, or even the possible in-place activation, of the software at the end of the evaluation period. Testing the removal of software becomes a step of increased importance in this situation. Hopefully, the process put in place will keep you from having to be the one to deny these regular user requests.
- OS upgrades—Some day, someone will need to upgrade to a new OS. Sometimes it is difficult to look so far ahead, but in the world of Windows, Microsoft has made it easier to see that day by publishing its Windows Product Lifecycle.

  This life cycle states that Windows desktop OS licenses will be available for purchase for a minimum of 4 years, with assisted support offerings available for up to 7 years after general availability. During these periods, Windows desktop product assisted support will move through a modified set of Product Lifecycle Phases. Standard support offerings will be available for the first 5 years after general availability. It will then go into an Extended Support Phase between 5 and 7 years after general availability. During this time, assisted support on an hourly basis and hotfixes may be available. However, there will be no requests for warranty support, design changes, or new features handled by Microsoft during this extended phase. Table 8.3 shows the Windows Product Lifecycles phases by date.

Desktop OS	Date of General Availability Start Date	OEM and Retail License Availability End Date
MS DOS 6.xx	1-Jun-94	31-Nov-01
Windows 95	15-Aug-95	31-Dec-00
Windows NT Workstation 4.xx	29-Jul-96	30-Jun-02
Windows 98	30-Jun-98	30-Jun-02
Windows 98 SE	30-Jun-99	30-Jun-02
Windows Millennium Edition	31-Dec-00	31-Dec-03
Windows 2000	31-Mar-00	31-Mar-04
Windows XP	31-Dec-01	31-Dec-05

Table 8.3: Windows Product Lifecycle dates.

For more information about Windows Product Lifecycles, visit http://support.microsoft.com/lifecycle.

- Software upgrades—Software release schedules can be annual, quarterly, or even monthly. Patches and fixes can be even more frequent. Although not all software will be critical enough to warrant this regular attention, much will. You must also consider the fact that if your software becomes too outdated, support and compatibility issues may also drive change. Plan for the likelihood of upgrades and patches and you may see opportunities to prepare for it when the time comes.
- Fixes—In particular, security fixes have been a common hot button for many environments. With Windows being scrutinized and taken apart by so many, security vulnerabilities surface on a frighteningly frequent basis.
- Hardware upgrades—As discussed earlier, increasing recommended requirements can be expected at unspecified intervals. For years, few software packages required more than a 386-based processor and varying amounts of RAM. Today, increases in hardware requirements, particularly in drive space and RAM, are increasingly frequent. Conversely, hardware upgrades may result in a need for software updates as well (as in the case of drivers and hardware support software).

Some organizations have an established hardware lifecycle in place. A percentage of the user desktops in the organization are retired and replaced at a regular interval. For example, 33 percent of the machines are replaced every year in a 3-year life cycle.

- Business changes—Reorganizations and relocations represent another common catalyst for change. For example, a newly formed group of people may need a specialized baseline or may have increased hardware requirements.

Establishing Policies for Change

The basis for a change management process is established policies and procedures. It is important to have a wide organizational involvement in the establishment and enforcement of these policies. These policies and procedures should be refined as improvements can be identified.

Goals and Objectives

What are you trying to accomplish with your change management process? In an attempt to attain a consistent environment, one mindset is to establish and maintain a common baseline by updating all systems together. Installing all software on all systems may seem like a licensing problem, but a software metering tool can provide a solution. Many organizations simply want to ensure that configurations are documented and managed in a controlled manner so as to minimize license violations and more easily associate specific problems to similarly configured systems. For example, if you know two software packages have an issue running on the same system, you can be proactive in addressing the problem if you can quickly establish which systems are at risk.

Roles and Responsibilities

There may be a great number of people involved in your change management process; thus, it is important to establish clearly defined barriers of responsibility. Roles may include making decisions or even just sitting in on meetings to represent a division or group.

Initiation Procedures

Depending upon the environment, requests could be initiated by a Web form, a board presentation, or a simple email. It is important that users should know where to go to request action and that a summary of the approval, development, testing, and deployment procedures be available.

Approval Procedures

Although in some organizations, it may seem like everything gets approved, there should be an approval process as a vital step in your change management process. Define what is required for approval or what factors will be weighed.

Establishing Business Reasons for Change

Installing the fifth different spreadsheet program on the network? Before software is ever purchased, there should be a business need presented for why the new software is required. The same goes for upgrades—the fact that an upgrade exists is not always the best reason for deploying it.

If the new software requested provides new functionality needed by your users, that reason is a good basis for a business case. An upgrade that makes a software product faster is a good basis for a business case. If the new or upgraded software can improve the work your users need to perform, this improvement can often be converted into time, and thus, into money.

Design, Planning, and Testing

As discussed in Chapter 4, establishing a process for designing, planning, and testing is essential. Not just in the implementation of new software and updates, but in hardware changes as well. How can the job be done? What is the best way to do it? And what could go wrong? After you answer these questions, validate the decision with testing in an environment as close to production as possible.

Scheduling

As mentioned earlier, having representatives from the various departments or business groups within your organization involved in meetings about change decisions is a very good idea. A representative from another department will have more intimate details of how changes may adversely affect their operations—in particular, schedules. Can the deployment happen at night? Over the weekend? Must it happen during business hours? If you need a user present, the decision is clear. If the change can be automated to occur during off hours, consider the possible affect of not having anyone present to address unexpected problems.

Communication

Are you going to announce deployments ahead of time? Some feel that advertising deployments causes users to look for problems where they may not exist. A typical example is for users to blame their computers' slowness on something the deployment team must have done to their machines. Despite this potential drawback, it is crucial to keep users informed. Letting users know of changes before they occur allows them to be aware of your successes and not just the failures. If users only learn of changes that affected them adversely, they will be justified in looking down upon the capabilities of the deployment team. What can the users expect? Will their machine reboot? Will the software they use behave differently? These are important topics to consider in change management.

Recovery and Support

Planning for problems and how to deal with what may go wrong beforehand can pay off if issues arise. Any change to be implemented should require a backout plan.

As a whole, a support process is typically a rather involved one. As it pertains to change management, think specifically how support will be implemented for the change. In some cases, it may make good sense to have the Help desk remain the primary focal point for support, but with instructions to route any problems related to the changeover to the team implementing the change.

📖 We explore support in more detail in Chapter 6.

Documentation and Tracking

Without documenting and tracking changes, maintaining control of your environment can quickly get away from you. Although software inventory can give you good insight into how things are, documentation will give you a picture of how things should be.

Waivers and Exceptions

No matter how streamlined your process, there will always be situations in which things must happen with greater speed. Establish a process for waivers and exceptions as well as an authoritative group for approving these exceptions. Although it is important to be ready to handle exceptions, it is even more important to avoid abuse. When every change request is classified as an urgent emergency, the benefits of the work done to establish a process for change may quickly diminish.

Summary

In this chapter, we have discussed software and hardware inventory as well as the benefits and methods that you can employ to maintain them. We covered the benefits of software license management as well as a number of tools available and features to consider when evaluating solutions. Finally, we covered change management, which should be developed and maintained from before the first workstation is even rolled out.