



Planning and Preparing for Active Directory

ITINERARY

- **Objective 1.01** Understanding a Directory Service
- **Objective 1.02** Understanding the Components of Active Directory
- **Objective 1.03** Installing Active Directory on Windows Server 2003



NEWBIE

1.5 hours

SOME EXPERIENCE

.5 hours

EXPERT

Optional

Before we start our study of Windows 2003's Active Directory, we must take note of a few base concepts upon which we will build our understanding. So while this introductory chapter will not provide vast volumes of information that will be tested when you sit down to take the 70-294 exam, it provides an essential overview of the concepts presented throughout the rest of the book.

Travel Advisory

Although these items are the objectives of the chapter, they are not Microsoft *exam* objectives. This chapter serves as a foundation for our Active Directory learning. In the chapters to come, however, the chapter objectives will map to Microsoft 70-294 exam objectives. That's not to say that this material isn't important—with a thorough understanding of the concepts involved, you stand a much better chance of assimilating the information to follow in this book.



It's entirely possible that you already possess the background knowledge necessary to begin delving into the inner workings of Microsoft's latest and greatest server operating system. If that's the case, or if you've only picked up this book to review vital exam material in as short a time as possible, you can probably begin your quest toward Active Directory Infrastructure certification in the next chapter. If, however, you have picked up this book as a primer for certification, or are new to Windows networking, it is recommended that you invest the time here to get comfortable with some of the underlying conceptual material. In any case, taking an extra half hour or two will do you no harm.

In this chapter, we will start with a look at what a directory service is, and then move on to a discussion of Microsoft's implementation of a directory service: Active Directory. We'll focus on the types of objects an Active Directory database can contain. Finally, we'll look at the process of installing a server computer with Active Directory, and some of the choices provided when you make the decision to implement a directory service on your network.

It's worth stating again: make sure you fully understand what's presented in this chapter before moving on to the rest of the book. Also, this chapter will provide a good reference point if you are confused by a term or concept that appears later in the book and need background information before continuing.



Objective 1.01

Understanding a Directory Service

As a starting point, you need to thoroughly comprehend what a directory service is. At the most basic level of computer networking, you have a server service that accepts connections from a client service for the purpose of making resources from one system available to other systems. In Windows Server 2003, these two software components are called Server and Workstation, respectively, and can be viewed by opening the Services management tool (select Start | All Programs | Administrative Tools | Services).

Travel Advisory

Keep in mind that a *server* is what a computer is *doing* in the network. Server 2003 is just a name of a Microsoft operating system. It's entirely possible that a system running Windows 98 will be the company's print server, and that a PC running Windows Server 2003 will be a client of the 98 machine. It's also entirely possible that this Windows Server 2003 will provide DNS server services to the Windows 98 machine.



Now, you don't just want anyone accessing a shared resource, so as networking software was developed (some of the earliest network operating systems included NetWare 3.12 and Windows for Workgroups 3.11), developers also included a way for these systems to provide authentication, so that the only users who could access network resources were the ones who were supposed to.

To perform authentication, a computer that's making resources available (for example, files or printers) will need a list of all the users who are allowed to interact with the resource. In the case of Windows for Workgroups, users were required to know a password in order to connect to resources on remote computers. These remote computers would have a list against which the username and password were checked to determine whether or not to permit access.

In its simplest iteration, the directory is the list. It's a database of users who have the ability to connect to a given system. The service is the software that makes the list available to the operating system. This service works in harmony with the server service (the software that makes resources available) so it can be checked when connection attempts are established.

Directory Services in Ancient Times

In older networks, like NetWare 3.12 or Windows for Workgroups, the directory was kept separate on each individual machine. This is still true in Windows peer-to-peer networks (or workgroups) where each machine keeps a unique database of the users who can use that machine. As networks grew, this became a prohibitively large administrative task as user accounts would need to be created at each individual server for which the user needed access. In other words, for one user, you often had many accounts.

In later versions of their respective networking software products, both Microsoft and Novell worked toward a centralized directory of users that could be checked for access to a network's resources. With a centralized database of users who were able to use network components, administrators no longer had to create multiple accounts for a single user, and administration could be centralized. Both of these companies' earlier versions, however, had varying degrees of difficulty accomplishing what some customers most wanted.

What you are working with right now, and are studying to master, is the result of years of Microsoft coders trying their best to build the ultimate piece of networking software. Lots of people trying to write a better mousetrap. (And just so you know, Apple is trying to do the same thing with their software, and has been working on it a while, too. If you're in the networking operating system business, you want your software in as many companies and government agencies as possible.)



Objective 1.02

Understanding the Components of Active Directory

Now that you know a little more about directory services and directory databases, you may be asking, what is significant about the current Microsoft version of a directory service—namely, Active Directory?

An excellent question. First of all, understand that Active Directory is built upon the networking technologies that preceded it. Active Directory, as a piece of Microsoft software, is certainly not unique in that regard. Microsoft and other software companies have taken existing technologies and/or the ideas of existing software, including those pieces of software that are openly available (such as the TCP/IP protocol suite), and built upon them. (Microsoft's detractors love to point this out, but there's nothing wrong with this; you can't copyright an idea.) They've taken what's out there and made it theirs, by taking the next evolutionary step.

In the case of Active Directory, Microsoft already had a working (if somewhat grumbled about) domain model built into their Server products. When Microsoft moved to the NT operating system, they added the capability to link domains together with trust relationships. Using one or more trust relationships between domains, the NT domain model became *scalable* because multiple domains could be made aware of each other's presence in an enterprise. This linking of domains was done for administrative reasons. Carefully implemented, the enterprise could still be managed centrally or, conversely, could be distributed among multiple administrative groups. This linking also provided for easy access to resources, and it accommodated business mergers and subsidiaries as organizations redefined themselves.

Travel Advisory

Scalable is a tech term that references a system's capability to grow and accommodate more users. Exactly what's scalable and what's not is rather nebulous and frequently a matter of conjecture; also, generally speaking, the term should be avoided in polite conversation. Software can be described as scalable, but hardware can be too. If it can grow relatively easily, it's said to be scalable. If there are limitations to adding more (users, computers, processors, and so on), it may be termed not scalable.



But scalability, like beauty, is a relative term. In other words, NT's domain model, though scalable, was not *very* scalable. As you started to add multiple domains to the NT mix, several problems occurred, and these problems usually compounded themselves as the organization began to grow.

For example, every time a user accessed a resource in another domain besides the one where their account lived, it increased network traffic. It also caused extra traffic as the domain controllers in these multiple domains maintained the trust relationships.

Local Lingo

Trust A logical link between two domains. It facilitates two things: cross-domain login, in which a user in one domain can submit his account credentials across the trust to the domain where his account lives, and universal resource access, in which a user has the ability to access resources in domains that trust the one where his account resides.



Remember as we discuss the significance of trusts here, that *ability* to access resources does not necessarily imply *permission*. It's entirely possible to be able to access something in another domain—to establish a connection—yet have read-only permission or have permission denied altogether.

The NT domain model became somewhat burdensome for growing enterprises. In its next iteration of its Server family of operating systems, Windows 2000, Microsoft set out to address two main areas of concern to help alleviate many of the problems encountered when working with the NT 4 domain model. The two things determined to be most needed in the next version of the directory service software were

- A global list of each domain's directory would need to be available at every domain.
- A system should exist to automatically manage trust relationships, lessening the administrative overhead involved in ensuring the benefits of multiple domains.

There's a corollary to these two driving characteristics. It was as if the folks at Microsoft, during a latte-fueled brainstorming session, thought, "If we're going to go to all this trouble to make this one great authentication database, wouldn't it be great if we could go beyond just authentication and fill this database up with handy data which would answer queries like, 'Which one of the printers on the fourth floor prints color?' or 'Is that computer located in the North building or the South building?' Let's design a directory like that."

So they did. These were the driving motivations for the design of the next domain model, the one that would carry Microsoft into the next millennium. The result was Active Directory, which made its debut in Windows 2000. Now Microsoft has carried forward this domain model and built upon it in its release of Server 2003.

Windows Server 2003 includes many improvements to Windows 2000's version of Active Directory, making it even more versatile, dependable, and economical to use. We won't go into too much detail here, but Active Directory improvements in Windows Server 2003 provide the following benefits:

- **Easier deployment and management** Improved migration and management tools, along with the ability to rename Active Directory domains, make deploying Active Directory significantly easier than when the directory service was first introduced in Windows 2000 Server. Better tools bring drag-and-drop capabilities, multi-object selection, and the ability to save and reuse queries. Plus, improvements in Group

Policy make it easier and more efficient to manage groups of users and computers in an Active Directory environment.

- **Greater security** Cross-forest trust provides a new type of Windows trust for managing the security relationship between two forests—greatly simplifying cross-forest security administration and authentication. Users can securely access resources in other forests without sacrificing the single sign-on and administrative benefits of having only one user ID and password maintained in the user’s home forest. This provides the flexibility to account for the need for some divisions or areas to have their own forest, yet maintain benefits of Active Directory.
- **Improved performance and dependability** Windows Server 2003 more efficiently manages the replication and synchronization of Active Directory information. Administrators can better control the types of information that are replicated and synchronized between domain controllers both within a domain as well as across domains. In addition, Active Directory provides more features to intelligently select only changed information for replication—no longer requiring updating entire portions of the directory.

Throughout the rest of the book, we will be dealing with the technologies that make these improvements possible. In fact, most of the terms and tools you will be learning about will impact one of these three areas in some way, and you can be sure that Microsoft will heavily emphasize these areas in the 70-291 exam as well.

The Directory Database

Active Directory’s job is to store and make available a directory database. While the term “database” can make cold shivers run down the spine of many a computer user, the concept can be simplified quite dramatically. At its core, a database is a list. If you’ve ever made a shopping list, you have worked—at least in concept—with a database.

The biggest difference between a list and a database (besides the fact that when we talk about databases, we are talking about things stored on a computer) is the way the information is organized.

In a database, one of the items in the list has a unique value. This unique value sets the item apart as a distinct entity and is known as a key value. In other words, a database stores a distinct object, and then a series of attributes of that object.

Take the example of one of the most common and most easily understood objects stored in Active Directory: the user account. To the Active Directory database, the unique part of the account is a number assigned to the account,

called a security identifier (SID). (We'll talk more about the SID throughout the book, especially in Chapter 5.) Everything else stored in the database about that user account are attributes. The logon name is an attribute of the SID. Likewise the first name, last name, and logon domain are all attributes. Lots of users can have the same domain attribute, and lots of users can even have the object attributes of Brian, as a first name attribute, and Culp, as a last name attribute, although as parents, teachers, coaches, and even book editors are fond of reminding me, one Brian Culp is more than enough, thank you.

The Schema

What's especially significant about Active Directory's list of objects that can be added to its directory database is that every Active Directory domain in an enterprise of domains contains the exact same list of potential objects and attributes. This list of what's possible to add to an Active Directory directory database is called the schema, and the schema is consistent throughout the Active Directory enterprise. In Chapter 3, we'll discuss all of these elements of the enterprise in further detail.

Active Directory groups a fixed set of attributes in the schema and defines them as a *class*. A class is used to describe a type of object. The class—and hence, the schema—can be added to, making Active Directory an extensible database. Active Directory is flexible enough to accommodate any kind of organization and can store just about any type of information. Say, for example, that you wanted to track a user's favorite charity as part of the information used to describe a user account. The Active Directory schema can be modified to include this information.

The Global Catalog

For each domain, we have a directory of information, and certain attributes from each domain will be copied to a Global Catalog. This “index” domain information is then shared among the multiple domains.

Active Directory uses a Global Catalog to provide a repository of the most commonly searched for objects and their attributes. Certain objects and attributes from each domain will be copied to the Global Catalog and made available to all connected domains.

You can think of the Global Catalog as an index for the domains in the enterprise. Each domain controller stores a copy of all the objects stored in its directory database. But users access the Global Catalog when they want to find information that's stored in domains other than their own.

The Global Catalog's main function is to make the following activities much easier:

- **Logging on at a domain other than the one you're in** In an Active Directory enterprise, it's possible to use the computers on one domain to log on to another. The Global Catalog is consulted for a list of possible logon domains.
- **Finding objects in other domains** If you are looking for color laser print resources throughout the enterprise, for example, the Global Catalog will likely store information about these printers.

The Active Directory Namespace

When you implement multiple domains in an enterprise, it will be easier to locate objects and to replicate information between domains if you impose some sort of consistent structure on them. The Active Directory software writers have done just this by borrowing from an existing domain structuring technology. This technology is used to provide the composition of the Internet domains, and thus provides a way of linking domains together. It's called the Domain Naming System (DNS), and now it's used to provide the backbone of your Active Directory domains.

The DNS namespace is one huge, hierarchical, distributed database of resource record mappings to IP addresses. It works by linking each and every domain into one contiguous namespace. You will need to have a pretty good understanding of DNS in order to successfully prepare for the 70-294 exam. Microsoft assumes that you bring DNS knowledge to the table before you study this material, but I haven't made the same assumption. Please refer to Appendix C for a detailed discussion of DNS.

What's significant as far as Active Directory goes is that, by using an existing hierarchical namespace in which the Active Directory domains will be named, you have created a system where trust relationships can be managed automatically. Why? Because the trusts relationships can be inferred from the domain names. For example, if you had an Active Directory domain called mcgraw-hill.com, and then created a child domain in the mcgraw-hill.com namespace, and named the child domain brianfanmail.mcgraw-hill.com, the trust relationship could be deduced from the names. If these were Windows 2003 Active Directory domains, the two domains would trust each other automatically. Trust relationships will be explored more fully in Chapter 4.

By using the DNS namespace, it's ensured that all your Active Directory domains will have names that are consistent. Additionally, there's a system that makes sure all this happens. The Domain Naming Master (explained in the next chapter) is the server that handles these duties for an Active Directory enterprise. This system ensures that all domains have names that fit cohesively in the namespace.

So, to summarize, these are the three keys to the engine that runs Microsoft's Active Directory:

- A common schema
- A common Global Catalog
- A common configuration

The Objects Stored in Active Directory

This may not exactly be news, especially if you've ever paid taxes or gotten a driver's license, but *you* are an *object*. At least you are to a Windows Server 2003 machine running Active Directory. In fact, everything is. Computers, groups, resources, Group Policies, and yes, even you, the user. I'll leave the decision about whether or not to fall into an Orwellian stupor over this piece of somewhat depressing information up to you.

It may come as some solace, then, to know why you are viewed the way you are by Active Directory: it's primarily to simplify security. All objects stored in Windows 2003's Active Directory database will have the following attributes attached:

- **Methods** Every object will have methods in common, such as creating the object, opening the object, and deleting the object.
- **Properties** All Active Directory objects have a set of properties or attributes, which you usually see by right-clicking on the object and choosing Properties from the context menu. At a minimum, an object will have a name property and a type property.
- **Collections** If an attribute can contain more than a single value (such as the members of a group object), these values are stored as collections or an array of values.

By treating everything as an object, Windows 2003 maintains security over the object no matter how a user accesses the object—whether locally, sitting at the domain controller, or remotely from another system. In fact, a significant collection attached to the objects in Active Directory is the Discretionary Access Control List (DACL), as seen in Figure 1.1. This list provides the one and only security model for access to objects in Windows 2003. Every time a connection to the object is attempted the DACL is checked before access is granted.

Now that you understand the concept that objects are stored in Active Directory, the question that remains is: What exactly are the objects that can be stored in the Active Directory directory database?

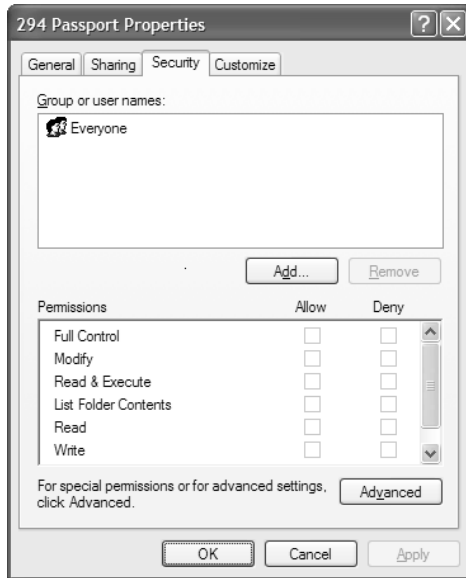


FIGURE 1.1 The Discretionary Access Control List

Computers

A computer object is a software representation of a physical entity, namely, the computer. It represents an important level of participation in the Active Directory domain. This level of participation usually has to do with security. You should also understand, when thinking of computer accounts, that only certain operating systems have this ability to create a computer account in the domain.

Any of the Windows Server 2003 or Windows 2000 (both Professional and the varieties of Server) operating systems can participate with a computer account. XP Professional, likewise, can create an account in a 2003 domain. However, the XP Home version, along with the 9x family of operating systems, cannot. This can be significant if you want to implement security options such as restricting a user's logon to a single computer or group of computers in an enterprise, or if you want to take the administrative steps of deploying software to computers using a Group Policy Object (Group Policies will be discussed at great length in Chapters 6–8). As will be reiterated throughout the book, you can only deploy Group Policy to computers that support it.

Travel Advisory

Windows NT 4 systems have the ability to participate as a computer account also, but since Microsoft is dropping support of NT 4, it's usually assumed for the purposes of computer account discussion that only the latest of Microsoft's operating systems will be used.



Users

User accounts comprise the meat and potatoes of Windows 2003 domain administration. All computing activities, whether it be access to a resource or backing up a file, occur in the context of a user account. An account is needed to interact with the network and is issued an *access token* at logon time. This access token is presented against a resource's Access Control List (discussed earlier in the chapter) to determine what level of access a user has. Without a valid user account, a user has no access to a Windows 2003 domain.

Groups

A group object is just another type of account, much like a user account. However, this account's purpose is to contain a list. In this list is an inventory of all the user accounts that belong to the container account (the group). It is also used at logon time, in conjunction with the user account, to help generate an access token. In other words, a group object is a gathering of other objects.

The advantage of a group is straightforward: it makes administration of permissions and rights simpler. When you grant a level of access permission to a group, the permission applies to all members of that group. This is especially advantageous as domain accounts grow to hundreds and thousands of accounts.

In Chapter 5, we'll take a more leisurely tour of what kinds of groups are possible in Windows 2003 domains and the significance of each.

Printers

In a Windows 2003 domain, you have the option of creating a software object in Active Directory for each shared printer in your enterprise. The advantage of creating an Active Directory object for each shared printer (rather than just creating the shared printer on a print server) is that it enables users to find an enterprise's printers more easily by conducting a search through Active Directory. That way, users don't have to know the physical location of the resource they're looking for. Just like when you use the phonebook, you don't have to know what street the business is on when you're looking for it, you just have to know what the business does.

Additionally, more attributes can be listed about the printer than can be with the shared object alone. When you share a printer, you just configure a share name. When you publish a printer in Active Directory, you can include information about the printer's functionality or what floor it's on, which can be a big advantage to users who want to send a job to it.

Note that even though the a printer object is created in Active Directory, the printer still exists as an object on a local machine, and that computer handles the security for that printer. The printer object that's stored in Active Directory is really just information about the printer.

Shared Folders

Much like printers, these resources are shared out from file servers in your enterprise. But also like printers, information about the shares can be published in Active Directory, facilitating easier searches when users are looking for resources. The computer hosting the share will still be responsible for managing the security permissions on that shared folder when it is accessed from the network.

Travel Assistance

For a full discussion of creation of shared resources and printers in a Windows 2003 Active Directory network, please see *Mike Meyers' MCSE/MCSA Windows® Server 2003 Network Passport (Exam 70-293)*.



Also keep in mind that the above are just some of the more common objects you create in an Active Directory directory database. You can also create Contact, InetorgPerson, and MSMQ Queue Alias objects, along with many others, as your needs require.

Now that you've learned about the background of Active Directory, and about what objects it can and commonly does contain, it's time to actually implement Active Directory in a Windows 2003 enterprise. The next section looks at how to do this.



Objective 1.03

Installing Active Directory on Windows Server 2003

Before you can take advantage of benefits afforded by Microsoft's Active Directory, you must know how to implement it. The process for installing a

Windows Server 2003 server computer begins by launching the Active Directory Installation Wizard, often referred to by its executable file, DCPRMO.

Travel Advisory

There are other ways to launch the Active Directory Installation Wizard, just as there are lots of ways to open the Control Panel. For example, you could choose Configure Your Server Wizard from the list of Administrative tools. The Configure Your Server Wizard is also the first thing that launches after you've got Windows 2003 installed. But for purposes of testing, you should know how to launch the Active Directory Installation Wizard from the Run dialog box, and the utility that launches it, as is demonstrated here. But if you like graphical interfaces for all your administrative tasks, you have my blessing. I mean, really, besides book authors with large disability insurance benefits covering carpal tunnel syndrome, who enjoys typing?



Using `dcpromo.exe`, you can install and remove Active Directory from a Windows Server 2003 computer at will. It provides a wizard-based interface that is as easy to use as any other wizard in the Windows environment. In fact, if you can set up a printer on a Windows 98 machine using the Add Printer Wizard, you have all the requisite button-pushing skill to install Active Directory. The hard part, as with all things Windows Server 2003, is not the button clicking, but possession of a full understanding of what will happen when the buttons are clicked.

Running the Active Directory Setup Wizard

As we go through this wizard, it is assumed that you are setting up a test environment, and therefore I will demonstrate the most straightforward way possible. As in most wizards, there are many different paths the Active Directory installation can follow, depending on your goals.

Before you start the Active Directory Installation Wizard, there are a few important requirements that must be met for successful installation. Here's a look at what must be in place before installing Active Directory, in no particular order. You probably won't be tested on this, but it wouldn't shock me if you saw a question demanding knowledge of these requirements:

- An NTFS partition for the SYSVOL folder
- Free space on your hard disk, about 1 GB

- The Installation Wizard must be run in the context of an account with administrative permissions in the domain or administrative permissions in the enterprise (the local Administrator account will be sufficient to create a forest root domain, which we will be doing here)
- DNS, which will be installed if an installation is not present.

It's also important to realize here that as we go further in this book, we will be taking different paths through the Installation Wizard than the one outlined here. These other paths will result in different roles for your Active Directory domain controllers, and will be examined in the detail necessary to fully understand them in later chapters. For now, our goal is just to build an Active Directory domain that we can work with later as necessary.

To launch the Active Directory Installation Wizard:

1. Click Start | Run, and then type **dcpromo** to start the Active Directory Installation Wizard.
2. Click Next to bypass the introductory screen of the Active Directory Installation Wizard.
3. On the Operating System Compatibility page, read the information and then click Next. Note here that the improved security characteristics of Windows Server 2003 set communications requirements of its clients that cannot be met with older operating systems such as Windows 9x and NT 4 with an outdated Service Pack.
4. Here's where the real forks in the road first appear. Since we are assuming that in this first chapter you may be setting up a test environment for learning purposes, we will create a new domain in a new enterprise of computers (your enterprise of computers, to be precise). On the Domain Controller Type page, select Domain Controller For A New Domain, as shown in Figure 1.2, and click Next.
5. Since we're starting a brand new enterprise, and not joining an existing one, from the Create New Domain page, select Domain In A New Forest, as shown in Figure 1.3, and then click Next.
6. Enter the full DNS name for the new domain and click Next. We'll talk extensively about the naming of your domains in the next chapter.



FIGURE 1.2 Creating a new domain

Travel Advisory

If you've ever surfed the Internet, you have used DNS domains like ebay.com or microsoft.com. You can use a similar namespace in the creation of your Active Directory domain. If you are creating a private namespace (it won't ever be searched for from a computer connected to the Internet), you can use any namespace you want, like mydoghas.fleas. For this example, I'll use the fictional namespace of lanscape.net. (At least as of the time of the writing, the name had not been registered.) But while the namespace will be fictional, the Active Directory installation will not. I just won't have resources available publicly.



7. From the ensuing dialog box, confirm the NetBIOS domain name. This will be important if you plan on having non-Active Directory clients, such as NT 4 clients, access your domain.
8. Confirm the location of the Active Directory database and log files. It is recommended practice that you locate these folders on different drives than the operating system for recoverability, although, as shown in Figure 1.4, the default locations are on the same partition as the operating system.

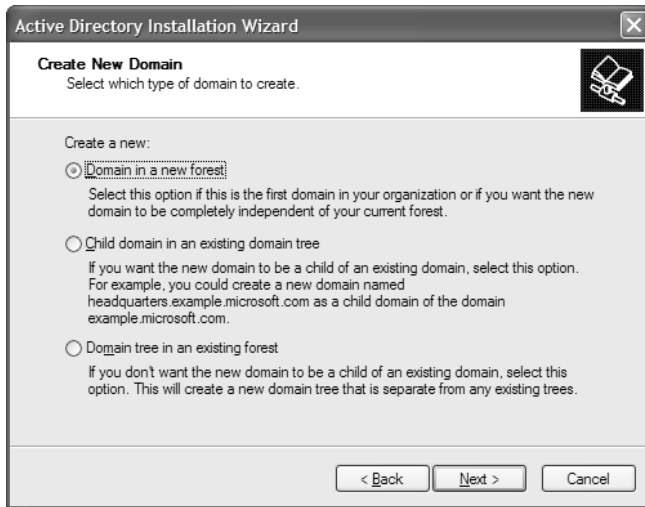


FIGURE 1.3 Creating a new domain tree in a new forest of domains

9. Confirm the location of the SYSVOL, or shared system volume, folder. This folder replaces the NETLOGON share functionality of NT 4 domain controllers and is the communications channel through which logon attempts are submitted.

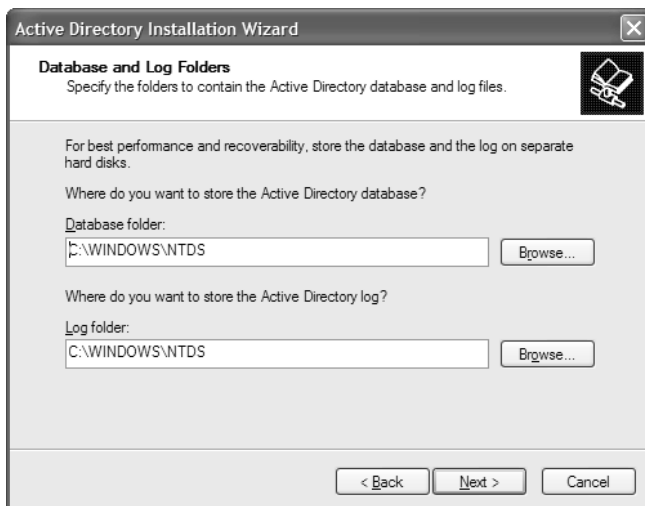


FIGURE 1.4 Specifying where the directory services and log files should be located

Travel Advisory

The SYSVOL folder must be placed on an NTFS-formatted drive. In other words, NTFS must be present for installation of Active Directory.



10. The wizard now confirms your DNS server installation on the DNS Registration Diagnostics page. You should receive a warning if a DNS server is not found. If a DNS server is not found, the wizard will offer to install DNS for you, as shown in Figure 1.5.

Travel Advisory

DNS installation by the Active Directory Installation Wizard is almost always a good idea when DNS cannot be located. If you choose No from this dialog box, Active Directory will still install, but you will receive numerous startup errors, and indeed Active Directory will not be functional until you address this problem. You must either find or create a DNS server that supports the necessary records that Active Directory creates. Windows 2003's DNS server supports these records, and the Active Directory Installation Wizard creates them automatically (you'll likely have to create them manually if using a UNIX DNS server solution, for example), so why not take advantage of the automation the wizard provides? Appendix C spells out in detail just what those records are.



11. Choose the option that begins "Install and configure the DNS server," and then click Next to let Windows 2003 set up DNS on its own.
12. On the next screen, you'll receive a security warning. In NT 4 and prior, the Remote Access Server (RAS) had to allow clients to read domain information before authentication. If your RAS servers are down-level (NT 4.0), you will need to allow this weaker level of permissions. Click Next after you've made your decision, which should be permissions compatible with Windows 2000 or Windows 2003 server operating systems unless absolutely necessary.
13. Next, you're prompted for the Directory Services Restore Mode password, as shown in Figure 1.6. This password does not have to be the same as your administrative password, but you should make sure it is safely stored—written down somewhere secure—so that it won't be forgotten.



FIGURE 1.5 The offering of DNS installation

14. The next screen will summarize your choices before starting the Active Directory installation. You can click Back to review and/or change any of your settings before clicking Next. Clicking Next starts the installation process. (If you've selected that DNS be configured, you

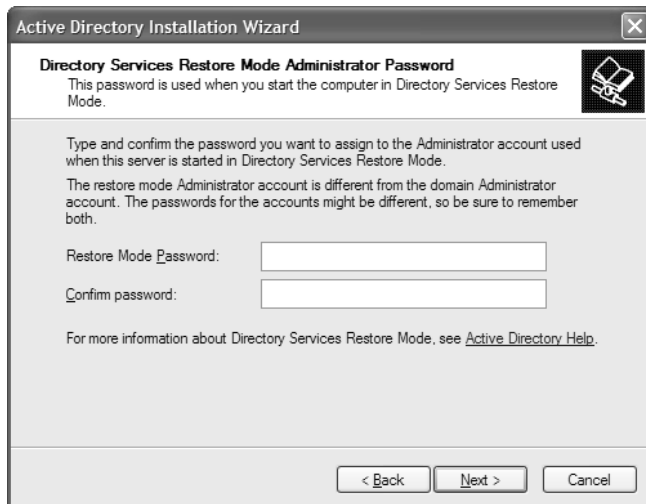


FIGURE 1.6 The Directory Services Restore Mode password

might be prompted for the location of your Windows Server 2003 installation files.)

15. If all has gone according to plan, you'll get a final dialog box, as shown in Figure 1.7, confirming the successful installation of Active Directory on the system. Once you've closed this dialog box, you'll be prompted to reboot the server.

Congratulations! You've just turned an ordinary Windows Server 2003 system into an all-powerful forest root domain controller. Now you've got a system that has all the directory services software necessary to run a computing empire such as mighty Microsoft itself, or, more modestly, to simply study the features of Active Directory from the comforts of your home (a somewhat smaller computing empire, I'll assume). Either way, the concepts will remain the same.

After the installation of Active Directory is complete, you will have three new Microsoft Management Console snap-ins added to your administrative tools so that you can, appropriately enough, manage what's in your Active Directory directory database. These three tools, which you will find by clicking Start | Programs | Administrative Tools, are the following:

Travel Advisory

Throughout this book, I will be giving instructions like "Open the Control Panel," or "Access the TCP/IP Properties page." There are lots of different ways to accomplish these tasks, but I'll assume that by this point in your certification studies, you know most of them, or at least the ones that work best for you. Most of my instructions will be given as if you are using my server, which I've configured with the XP Theme interface, but with the older Windows 9x-style Start menu and icons like My Computer and My Network Places added to the desktop.



- **Active Directory Users and Computers** This will be the tool used to manage most of your Active Directory objects, such as (who would guess!) User accounts, Computer accounts, as well as Group accounts, published printers, and shares, to name a few. You will also be coming here often to do your Group Policy management.
- **Active Directory Domains and Trusts** This tool will be used to manage trust relationships between the domains of your Active Directory enterprise and other external domains. Earlier I explained that one of the benefits of Windows 2003's Active Directory domain model, especially when compared to NT 4's model, was that

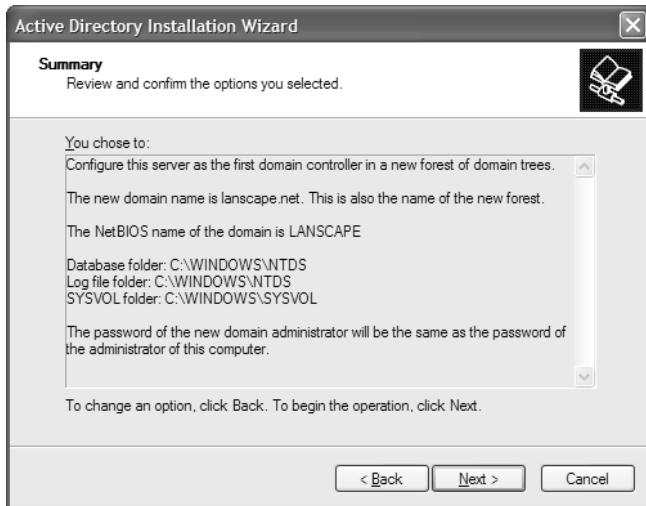


FIGURE 1.7 The final dialog box will sum up your accomplishment.

administration of trusts in your enterprise is virtually nonexistent. That's still true. What you will use this tool for is to specify and manage trusts between either older NT 4 domains or between domains of external organizations and yours.

- **Active Directory Sites and Services** This will be used to create and manage the directory services sites, and to further manage the replication of Active Directory information between these sites.

There are other tools installed as well, such as Domain Controller Security Policy and Domain Security Policy, but the tools most important to managing Active Directory objects are listed above. We will be using all of these tools, especially Active Directory Users and Computers and Active Directory Sites and Services, extensively throughout this book, so stay tuned if you're aching to find out more about the uses of these MMC consoles.

Travel Advisory

The Microsoft Management Console is the framework against which the management utilities are written. An MMC can be customized to best suit your needs, but several *preconfigured* MMCs are automatically installed when you set up Windows Server 2003, and more are added as you add more software components to manage. By itself, however, the MMC has no functionality.



The MMC should be very familiar to Windows 2000 administrators, or even to admins of any of the Microsoft Back Office software products (SQL Server, ISA Server, SMS, and so on). For further information on the MMC, please see the Microsoft Windows 2003 Resource Kit.

If you're upgrading a system—and indeed, a domain model—from Windows NT 4.0 (it's about time!), you will find great solace in the fact that Active Directory is just as easily uninstalled from any single domain controller. Under the Windows NT 4.0 environment, administrators had to reinstall the operating system in order to reconfigure. That's because the decision to operate as a domain controller was made at Windows installation time and was thereafter set. Changing server roles was not as flexible as it is today.

Ideal Domain Design

Additionally, keep in the back of your mind that Microsoft recommends that, whenever possible, you design your Active Directory infrastructure to include only a single domain. This is because a single Windows 2003 domain is scalable (there's that word again) to millions of objects. In other words, a single domain has the potential to accommodate organizations of almost any conceivable size, including Microsoft itself. When you implement a single domain, all of the Active Directory concepts still apply, but you can be blissfully unaware of almost every one of them and still have a good working network. If you don't have testing in mind, Microsoft makes it awfully easy to administer a small to medium sized company with its newest version of Server.

But that's not our goal here, is it? You picked up this book to quickly obtain the information that Microsoft expects you to know for the 70-294 exam. In the next chapter, we start focusing on precisely that information.



- ✓ **Objective 1.01: Understanding a Directory Service** In this chapter, we've examined the foundational information necessary to understand Windows 2003's Active Directory. With an understanding of these concepts, we will go forward to explore how to best implement Active Directory. We started with a look at the history of a directory service, and the role a directory database plays in a modern computing network. We also examined how a directory service serves to authenticate users on a network, and how it enables access to resources in multiple domains.

- ✓ **Objective 1.02: Understanding the Components of Active Directory** Next we looked at the parts of an Active Directory environment. These Active Directory components include users, computers, domains, global catalogs, schemas, and domains. We also mentioned that the Active Directory namespace is built on an existing, open-standards based naming technology: DNS. This vital Active Directory concept is further discussed in Appendix C.
- ✓ **Objective 1.03: Installing Active Directory on Windows Server 2003** In the chapter's final section, we walked through the steps to set up Active Directory on a computer running Windows Server 2003. You need to install Active Directory so that you can follow along with some of the steps later in the book. Also, it's generally a good idea install the software you're going to spend a considerable time studying. We also looked at the Microsoft recommendation for ideal domain design.

REVIEW QUESTIONS

1. Which of the following attributes are contained in every object created in an Active Directory domain? Choose all that apply.
 - A. A first name
 - B. A security identifier
 - C. A common name
 - D. A class
2. Which of the following components of Active Directory are shared by all domains in an Active Directory enterprise? Choose all that apply.
 - A. Schema
 - B. Global Catalog
 - C. Parent domain
 - D. Configuration
3. What are the two items that make up the Active Directory schema?
 - A. Classes
 - B. Names
 - C. Attributes
 - D. Objects
4. What is the utility used to install Active Directory on a Windows Server 2003 computer?
 - A. promotedc.bat
 - B. dcupgrade.exe

- C. adinstall.exe
 - D. dcpromo.exe
5. Which of the following components of Active Directory need to be installed on an NTFS partition? Choose all that apply.
- A. The SYSVOL folder
 - B. The Active Directory log files
 - C. The Active Directory database
 - D. The 2003 operating system

REVIEW ANSWERS

1. **B C D** Every object in Active Directory needs to have a unique security identifier (SID), a unique name that includes the domain name where the object resides, and a class from which the object was created.
2. **A B D** There are three elements that are common to all domains in a linked Active Directory enterprise. They are the schema (a list of what's possible to add to Active Directory), the Global Catalog (an index of commonly used objects and their attributes), and the configuration.
3. **A C** Classes are really preset templates of attributes. There are preset classes for computer, user, printer, and organizational unit Active Directory objects. Attributes store specific information about a class.
4. **D** dcpromo.exe launches the Active Directory Installation Wizard, which is used to install Active Directory on a given Windows Server 2003 system.
5. **A** In order for Active Directory to install, the SYSVOL folder must be placed on an NTFS drive. The scripts folder contained within the SYSVOL folder will be shared out as NETLOGON, and will be the location where logon requests to the domain are submitted.