Active Directory® FOR DUMMIES®

2ND EDITION

by Steve Clines and Marcia Loughry

WILEY

Wiley Publishing, Inc.

Active Directory® For Dummies,® 2nd Edition

Published by
Wiley Publishing, Inc.
111 River Street
Hoboken, NJ 07030-5774

www.wiley.com

Published by Wiley Publishing, Inc., Indianapolis, Indiana

Published simultaneously in Canada

LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: THE PUBLISHER AND THE AUTHOR MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS WORK AND SPECIFICALLY DISCLAIM ALL WARRANTIES, INCLUDING WITHOUT LIMITATION WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES OR PROMOTIONAL MATERIALS. THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR EVERY SITUATION. THIS WORK IS SOLD WITH THE UNDERSTANDING THAT THE PUBLISHER IS NOT ENGAGED IN RENDERING LEGAL, ACCOUNTING, OR OTHER PROFESSIONAL SERVICES. IF PROFESSIONAL ASSISTANCE IS REQUIRED, THE SERVICES OF A COMPETENT PROFESSIONAL PERSON SHOULD BE SOUGHT. NEITHER THE PUBLISHER NOR THE AUTHOR SHALL BE LIABLE FOR DAMAGES ARISING HEREFROM. THE FACT THAT AN ORGANIZATION OR WEBSITE IS REFERRED TO IN THIS WORK AS A CITATION AND/OR A POTENTIAL SOURCE OF FURTHER INFORMATION DOES NOT MEAN THAT THE AUTHOR OR THE PUBLISHER ENDORSES THE INFORMATION THE ORGANIZATION OR WEBSITE MAY PROVIDE OR RECOMMENDATIONS IT MAY MAKE. FURTHER, READERS SHOULD BE AWARE THAT INTERNET WEBSITES LISTED IN THIS WORK MAY HAVE CHANGED OR DISAPPEARED BETWEEN WHEN THIS WORK WAS WRITTEN AND WHEN IT IS READ.

For general information on our other products and services, please contact our Customer Care Department within the U.S. at 800-762-2974, outside the U.S. at 317-572-3993, or fax 317-572-4002.

For technical support, please visit www.wiley.com/techsupport.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

Library of Congress Control Number: 2008932078

ISBN: 978-0-470-28720-0

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

WILEY

About the Authors

Steve Clines, MCSE, MCT, has worked as an IT architect and engineer at EDS for over 18 years. He has worked on deployments of more than 100,000 seats for both Active Directory and Microsoft Exchange Server. Steve is the author of *MCSE Designing a Windows 2000 Directory Services Infrastructure For Dummies,* which is a study guide for the 70-219 MCP exam. He also maintains the Confessions of an IT Geek blog at `http://itgeek.steveco.net`.

Marcia Loughry, MCSE and MCP+I, is a Senior Infrastructure Specialist with a large IT firm in Dallas, Texas. She is president of the Plano, Texas BackOffice User Group (PBUG) and a member of Women in Technology International. Marcia received her MCSE in NT 3.51 in 1997 and completed requirements for the NT 4.0 track in 1998.

Marcia has extensive experience working with Windows NT 3.51 and 4.0 in enterprises of all sizes. She is assigned to some of her firm's largest customers in designing NT solutions and integrating UNIX and NetWare environments with NT.

Dedication

Steve Clines: I am dedicating this book to two people who are no longer with us. First is my mom Glenda. She is the one who really taught me about writing and how to see a project to its completion. The second person is my nephew Boomer. You have reminded me of how precious life really is and how we are to live each day with the joy that you did.

You are both missed.

Marcia Loughry: This book is dedicated to my family — my son, Chris, my parents, my sister, Karen — just because I love 'em all! Thanks for the love, laughter, and support.

Authors' Acknowledgements

Steve Clines: I have many people to thank for their support. Foremost is my wife, Tracie, who has been my constant support. I couldn't have done this without you. Also, thank you to my family and friends who have been a great source of continual encouragement to me.

Thank you to Marcia Loughry for getting me started down this road and giving me a great starting point for doing this edition. Also, thanks to all the great folks at Wiley Publishing for giving me this opportunity and being really easy to work with.

Lastly, thanks to my Lord and Savior. I can't do anything without you – Phil. 4:13.

Marcia Loughry: Special thanks to literary agent Lisa Swayne, of the Swayne Agency, for finding me, taking me on, and introducing me to the fun people at Wiley Publishing.

Many, many thanks to the fine folks at Wiley Publishing: Joyce Pepple, who get me excited about this project; Jodi Jensen, who suffered and planned with me and generally kept me in line; Bill Barton, who didn't strangle me over my consistent use of passive voice; and the rest of the Wiley team who made the book and CD possible.

And finally, heartfelt thanks to Jackie, Mary, Sherri, Michelle, Anne, Clifton, Sam, Steve, Kent, Sylvana, Nate, Clay, and all the other friends who make every day so fun.

Publisher's Acknowledgments

We're proud of this book; please send us your comments through our online registration form located at www.dummies.com/register/.

Some of the people who helped bring this book to market include the following:

Acquisitions, Editorial, and Media Development

Sr. Project Editor: Christopher Morris

Acquisitions Editor: Kyle Looper

Copy Editor: Brian Walls

Technical Editor: John Mueller

Editorial Manager: Kevin Kirschner

Editorial Assistant: Amanda Foxworth

Sr. Editorial Assistant: Cherie Case

Cartoons: Rich Tennant (www.the5thwave.com)

Composition Services

Project Coordinator: Katherine Key

Layout and Graphics: Stacie Brooks, Reuben W. Davis, Laura Pence, Ronald Terry

Proofreaders: Caitie Kelly, Bonnie Mikkelson, Amanda Steiner

Indexer: Rebecca Salerno

Publishing and Editorial for Technology Dummies

Richard Swadley, Vice President and Executive Group Publisher

Andy Cummings, Vice President and Publisher

Mary Bednarek, Executive Acquisitions Director

Mary C. Corder, Editorial Director

Publishing for Consumer Dummies

Diane Graves Steele, Vice President and Publisher

Joyce Pepple, Acquisitions Director

Composition Services

Gerry Fahey, Vice President of Production Services

Debbie Stailey, Director of Composition Services

Contents at a Glance

Table of Contents

Introduction

Welcome to the wonderful world of Active Directory! Over the last eight years since Active Directory (AD) was released in Microsoft's Windows 2000 Server product, AD has become one of the most (if not the most) popular directory service products in the world. It has also become one of the central technologies on top of which many other Microsoft products are built. If you are an Information Technology (IT) professional who designs and supports directory services or solutions created with Microsoft products, then you really need to have an understanding of what AD is and how it works. That's where this book comes in.

My goal with this book is to take the anxiety and stress out of mastering this complex technology. I hope that you find the book a clear, straightforward resource for exploring Active Directory.

This Book Is for You

Whether you've purchased this book or are browsing through it in the bookstore, know that you've come to the right place. Maybe you are like me. When I'm looking through a book that I'm considering purchasing, I always look at the first sections to try to get an idea of who the book is written for and exactly what it's going to cover. So let me just get this out of the way right now. This book is for you if you're any of the following:

- A savvy system administrator with previous NT experience who needs to find out about Active Directory
- An administrator that has AD experience with previous releases in Windows 2000 Server and Windows Server 2003
- Someone who wants to know more about Active Directory Domain Services in Windows Server 2008
- Someone who wants to find out about the new components of Active Directory in Windows Server 2008, including Active Directory Lightweight Directory Services, Active Directory Federation Services, Active Directory Certificate Services, and Active Directory Rights Management Services

- A newbie (to networking or to information technology) who wants to pick up information on Active Directory
- A student preparing for AD certification exams
- Someone who's merely interested in intelligently discussing Active Directory

For the experienced Windows Server administrator or other IT professional, *Active Directory For Dummies* provides you with an unpretentious resource containing exactly what you need to know. It presents the fundamentals of the program and then moves right into planning, implementing, and managing Active Directory — what you're most interested in knowing right now!

Welcome! And, thanks for making *Active Directory For Dummies* your first resource for figuring out one of Microsoft's hottest technologies!

How This Book Is Organized

I've divided this book into six parts, organized by topic. The parts take you sequentially from Active Directory fundamentals through planning, deploying, and managing Active Directory. If you're looking for information on a specific Active Directory topic, check the headings in the table of contents. By design, you find that you can use *Active Directory For Dummies* as a reference that you reach for again and again.

Part I: Getting Started

Part I contains the "getting to know you" chapters. These chapters contain the answers to your most fundamental questions:

- What is Active Directory?
- What are its benefits?
- What are the buzzwords?

The information you find here helps you determine what you must do to prepare for Active Directory in your environment. Also, in this part, I provide you information that can help you gather information about the environment you're deploying AD in and how to develop the requirements that will drive your Active Directory design.

Part II: Planning and Deploying with Active Directory Domain Services

Active Directory Domain Services contains both a logical and a physical structure that you must carefully design before deployment. The logical structure comes first and includes the following steps:

- Planning the DNS namespace
- Designing the forest/domain/organizational unit (OU) model

After you plan your logical structure, you move on to developing a plan for your physical structure. This part ends with you putting all this planning into action as you build your Active Directory forest by creating domain controllers.

Part III: New Active Directory Features

In Windows Server 2008, Microsoft has added a number of new components to Active Directory that expand the product beyond being simply a directory service. Many of these components can be used to develop an overall identity and access management solution. These components support interaction between external users — even other companies — and your internal AD environment. If you're familiar with Active Directory from a previous Windows Server release and need to find out about the new parts of AD in Windows Server 2008, this is one part you want to check out!

Part IV: Managing Active Directory

Part IV covers the daily management of an Active Directory environment. Active Directory introduces the capability of delegating administrative authority and also introduces security concepts. The chapters in this part prepare you for managing security, users, and resources within the Active Directory tree.

Part IV also covers managing replication traffic. Optimized replication traffic is vitally important to the Active Directory environment. In these chapters, you discover how to propagate updates, schedule replication traffic, work with the Active Directory schema, and maintain the Active Directory database.

Part V: The Part of Tens

In true *For Dummies* style, this book includes a Part of Tens. These chapters introduce lists of ten items about a variety of informative topics. Here you find additional resources, hints, and tips, plus other nuggets of knowledge.

Part VI: Appendixes

In the appendixes, you find information that adds depth to your understanding and use of Active Directory. I provide a listing of command line utilities for managing Active Directory as well as a glossary of terminology.

Icons Used in This Book

To make using this book easier, I use various icons in the margins to indicate particular points of interest.

TECHNICAL STUFF

Sometimes I feel obligated to give you some technical information, although it doesn't really affect how you use Active Directory. I mark that stuff with this geeky fellow so that you know it's just background information.

WARNING!

Ouch! I mark important directions to keep you out of trouble with this icon. These paragraphs contain facts that can keep you from having nightmares.

TIP

Any time that I can give you a hint or a tip that makes a subject or task easier, I mark it with this little thingie for additional emphasis — just my way of showing you that I'm on your side.

REMEMBER

This icon is a friendly reminder for something that you want to make sure that you cache in your memory for later use.

Part I
Getting Started

The 5th Wave By Rich Tennant

"We're much better prepared for this upgrade than before. We're giving users additional training, better manuals, and a morphine drip."

In this part . . .

For many things in life, you have to start at the beginning before you can move on to the rest. That start for Active Directory is here. The first chapter is an introduction to Active Directory and its terminology. Chapters 2 and 3 step back from the technology of Active Directory and instead discuss how to prepare for an Active Directory design and deployment by looking at what requirements you have and developing an implementation plan. Welcome to Active Directory!

Chapter 1

Understanding Active Directory

In This Chapter

- Defining Active Directory
- Examining the origins of Active Directory: X.500
- Understanding Active Directory terms
- Investigating the benefits of Active Directory: What's in it for you?

Since the release of Active Directory in Windows 2000 Server, Active Directory has become a very integral part of many information technology (IT) environments. As such, Active Directory has become a very popular topic with the people that have to design and support it. Because of all the terms and technology surrounding Active Directory, you might already be a bit intimidated by the prospect of working with it yourself.

But Active Directory doesn't need to be difficult! In this chapter, you find out in clear and simple language what Active Directory is, what it does, and what benefits it brings to your organization and to your job.

What Is Active Directory?

If you visit the Microsoft Web site seeking a definition of Active Directory (AD), you find words such as *hierarchical, distributed, extensible,* and *integrated*. Then you stumble across terms such as *trees, forests,* and *leaf objects* in combination with the usual abbreviations and standards: TCP/IP, DNS, X.500, LDAP. The whole thing quickly becomes pretty overwhelming. (Appendix B has a glossary that defines these abbreviations for you!)

I prefer to define things in simpler terms, as the following sections demonstrate — drum roll, please . . .

Active Directory is an umbrella

What? Am I saying that if it's raining you had better have AD with you? No, I would still recommend a real umbrella in a rainstorm. I'm saying that in Windows Server 2008, the scope of what Active Directory is has greatly expanded. Active Directory has become an umbrella for a number of technologies beyond what AD was in Windows 2000 Server and Windows Server 2003. (See Figure 1-1.)

You discover new uses for Active Directory in the paragraphs that follow.

Active Directory Domain Services

What was *AD* in the two previous Windows Server operating systems is now *Active Directory Domain Services,* or *AD DS,* in Windows Server 2008. The majority of this book deals with this component of Active Directory because this is the most commonly deployed component of the AD umbrella. But don't worry; I discuss all the other technologies found beneath the Active Directory umbrella as well.

Active Directory Lightweight Directory Services

Beginning with Windows Server 2003, Microsoft created a directory service application separate from Active Directory called *Active Directory Application Mode* or *ADAM* for short. ADAM was designed to address an organization's needs to deploy a directory service that didn't necessarily need all the features that Active Directory provided. Microsoft includes this application in Windows Server 2008 but renamed it *Active Directory Lightweight Directory Services* or *AD LDS.* I talk about AD LDS in Chapter 8.

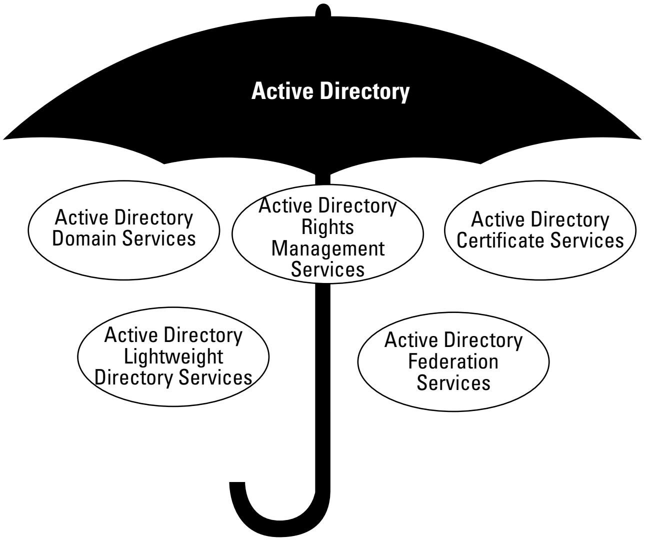

Active Directory
Active Directory Domain Services
Active Directory Rights Management Services
Active Directory Certificate Services
Active Directory Lightweight Directory Services
Active Directory Federation Services

Figure 1-1: The Active Directory umbrella.

Active Directory Federation Services

Beginning in the R2 release of Windows Server 2003, Microsoft included an optional software package called *Federation Services*. As you see later in this book, federations provide a Single Sign-on (SSO) service helping to minimize the number of logon IDs and passwords users must remember as well as simplifying how users can access resources in other IT environments. This software is now a part of the Windows Server 2008 AD umbrella and has been renamed *Active Directory Federation Services* or *AD FS*.

Active Directory Certificate Services

Certificate Services has been around in Windows Server software for a while now. With this software, you can provide certification authorities that can issue public key certificates used for such things as authentication via smart cards or encrypting data before it's transmitted over a network. Certificate Services also provides the necessary management of these certificates so that they can be renewed and revoked. In Windows Server 2008, Certificate Services is a part of Active Directory and is referred to as *Active Directory Certificate Services (AD CS)*.

Active Directory Rights Management Services

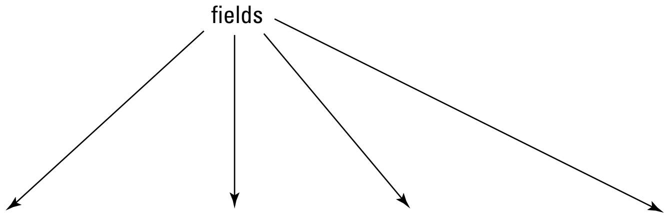
Managing what users can do with data has always been an issue for most organizations. Although Active Directory did a good job of controlling whether a user could access a document, it didn't have the ability to control what that user did with the data after he or she got it. Enter *Active Directory Rights Management Services (AD RMS)*. With a properly deployed AD RMS environment, organizations can retain control over sensitive documents, for example, so that they cannot be e-mailed to unauthorized users.

REMEMBER

I use the term *Active Directory* interchangeably with *Active Directory Domain Services*. This is because in previous versions of Windows Server software, Active Directory was what is now called *Active Directory Domain Services.* When I refer to the Active Directory umbrella as Active Directory, I make it clear that I'm not just talking about AD DS. Additionally, when I refer to the other elements of AD, such as Active Directory Federation Services, I call it that or use its acronym.

Active Directory is an information store

First and foremost, Active Directory is a store of information. This information is organized into individual objects of data, each object having a certain set of attributes associated with it. A telephone white pages directory, for example, is an information store. Each object in this store represents a home or business that contains attributes for such information as names, addresses, and telephone numbers (see Figure 1-2).



LAST NAME	FIRST NAME	ADDRESS	TELEPHONE NUMBER
Adams	Alison	123 ABC Place	000-123-4567
Baker	Joe	234 Tree Street	000-123-4568
Smith	Alex	456 Forest Drive	000-123-4569

Figure 1-2: A telephone directory is a store containing fields of information.

This store of data as well as the capability of retrieving and modifying the data makes Active Directory a *directory service*. Why then don't I consider Active Directory to be a database? It certainly shares some common functionality including storage, retrieval, and replication of data, but there are some important differences, too. First, directory services are normally optimized for reads because these are the vast majority of the operations executed, and the data is generally non-changing. Also, the data is structured in some sort of hierarchy that allows for it to be organized in the directory store. Repeating my phone book analogy, the Yellow Pages organizes objects by types of business. This makes finding what you're looking for easier. The same can be said of a directory service — you can organize your objects into a hierarchy of containers so that finding the objects is easier. In comparison, a relational database, such as Microsoft SQL Server, is designed to optimize both reads and writes to the store because the data is frequently being read and written to. Also, a database generally doesn't force a hierarchy on the data like a directory service does.

Where did it come from?

Active Directory Domain Services has evolved, but it actually began its life as the directory service for Microsoft Exchange Server V4.0 through V5.5. AD DS actually derives from a directory service standard — *X.500.* The X.500 standard is a set of recommendations for designers of directory services to ensure that the products of various vendors can work together. These are the X.500 protocols:

- Directory Access Protocol (DAP)
- Directory System Protocol (DSP)
- Directory Information Shadowing Protocol (DISP)
- Directory Operational Binding Management Protocol (DOP)

Active Directory, however, actually uses the Lightweight Directory Access Protocol (LDAP) Version 3 (defined in RFC 1777 and RFC 2251), to access the directory database instead of using any of the preceding X.500 protocols. Therefore, Active Directory is X.500 *compatible,* meaning that it can work with other X.500-based directory services, but not X.500 *compliant* — it doesn't strictly adhere to all the X.500 specifications.

In Active Directory, the term *object* can refer to a user, a group, a printer, or any other real component and its accompanying attributes. Active Directory is an information store containing all the objects in your Windows 2008 environment.

Active Directory has a structure (Or hierarchy)

A directory service, such as Active Directory, allows for the objects in it to be stored in a hierarchy or structure. This structure is one of the areas that you design as a part of deploying Active Directory. This structure has two sides:

- **A logical side:** The logical structure provides for the organization of the objects. These AD objects can represent users, computers, groups, and a variety of other items that are in your IT environment. This structure is primarily dependent on how you want to administer your IT infrastructure as well as how your organization is structured.
- **A physical side:** All the services under the Active Directory umbrella are provided by servers running the AD software. These servers represent physical objects that must be placed within your network. After these servers are placed, you must define how these servers speak to each other and how users are directed to them. This physical topology is critical to proper AD functionality.

Staying with the phone book analogy, unless the books are placed in the proper locations (homes, restaurants, pay phones), no one can find the books to utilize the information contained within them.

Active Directory can be customized

As you can with an electronic phone book, you can search Active Directory for the objects that you want to access. Unlike a phone book, however, you can customize Active Directory to include additional objects and object attributes that you deem important. This feature makes Active Directory *extensible,* which means that you can add to it.

Getting Hip to Active Directory Lingo

Experience shows that new terminology often accompanies new technologies, and Active Directory is no exception. Although most of the terms that you use in describing the system might seem familiar, they take on new meaning in relation to Active Directory. So before beginning to plan and implement Active Directory, you need to master its new language.

The building blocks of Active Directory

Active Directory embodies both a *physical* and a *logical* structure. The *physical structure* encompasses the network configuration, network devices, and network bandwidth. The *logical structure* is conceptual; it aims to match the Active Directory configuration to the business processes of a corporation or organization. In the best logical structures, Active Directory resources are structured for how employees work and how the environment is administrated, not to simplify construction of the network.

If you logically organize the components within the Active Directory, the actual physical structure of the network becomes inconsequential to the end-users. If user JoeB wants to print to a printer named A5, for example, he no longer needs to know which server hosts the printer or in which domain the print server resides. In Active Directory, he simply pulls up an Active Directory list of all available printers and chooses printer A5.

Although you might think that this process sounds too good to be true, this new functionality doesn't quite configure itself! You, the system administrator, must first design the logical structure of your organization's Active Directory, matching its structure to how employees interact within the organization. Chapters 2 through 7 help you to plan and implement, but first, you must be familiar with the individual components that you use for planning the physical and logical structures.

Domain

In Active Directory, Microsoft defines a *domain* as a security boundary or an administrative boundary, which means that all the users within a domain normally function under the same security policy and user-account policy. If you want to assign different policies to some users, those users belong in a separate domain.

JohnB, for example, is a regular user in the Sales department who must change his password every 30 days. SueD, on the other hand, is a user in the Treasury department who has access to sensitive information and, therefore, must change her password every 14 days. The two departments — Sales and Treasury — have different user-account policy settings. Because you assign user-account policies according to domain, users in these two departments belong in separate domains.

WARNING!

In Windows Server 2008, the lines between domain boundaries and password policies has blurred somewhat. Normally, all users in a domain receive the same password policy; however, in 2008, you can do some fine-tuning so that users in the same domain actually receive different policies. I cover this in more detail in Chapter 14.

Here are some other important characteristics of an Active Directory domain:

- A domain has at least one *domain controller*. A domain controller is a server that *authenticates* (validates the password and ID) users seeking access to the domain. You find out more about domain controllers in a moment.
- A domain's directory database *replicates* between all domain controllers in the domain. Replication is the exchange of updated information among domain controllers so that all the domain controllers contain identical information.
- A single domain can form a *tree* (which you find out more about in the following section).

In the design process for the logical structure of an Active Directory database, you typically use a triangle in the design flowchart to represent a domain (see Figure 1-3).

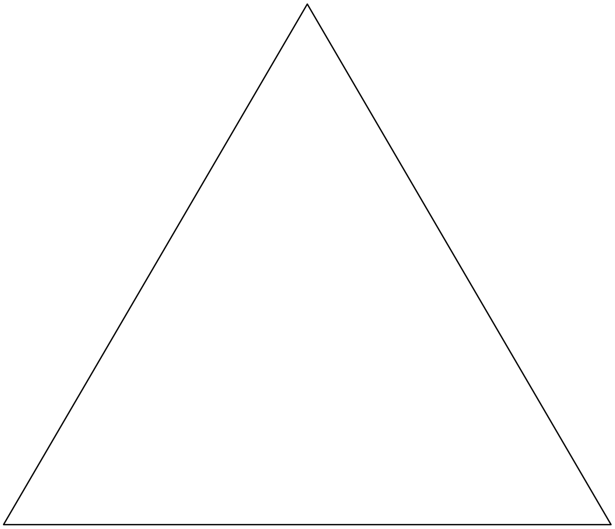


Figure 1-3: A triangle represents a domain when drawing an AD logical design.

TIP

Consider defining an additional domain to keep replication traffic local — confined among domain controllers connected by a local area network (LAN). The transmission speed between domain controllers in a LAN is much faster than it is between domain controllers that are connected by a slower, wide area network (WAN). The exchange of updated database information among domain controllers during replication causes additional traffic that can clog the network and result in slower response times. So by keeping your replication local, you can keep replication time to a minimum and ensure that the network lines are available for other traffic. (I talk more about defining domains in Chapter 5.)

Tree

A *tree* is a hierarchical grouping of domains within the same namespace. A *namespace* is a logically structured naming convention in which all objects are connected in an unbroken sequence. (I talk more about namespaces later in this chapter and in Chapter 4.) When you design an Active Directory tree, you begin with the topmost domain, which oddly enough is the *root* (or *parent*) domain. Subdomains (sometimes *child domains*) branch downward from the root, as shown in Figure 1-4. Supposedly, if you turn your logical structure drawing upside down, it resembles a tree. (Go on — turn the book upside down and look for the image of a tree in Figure 1-4!)

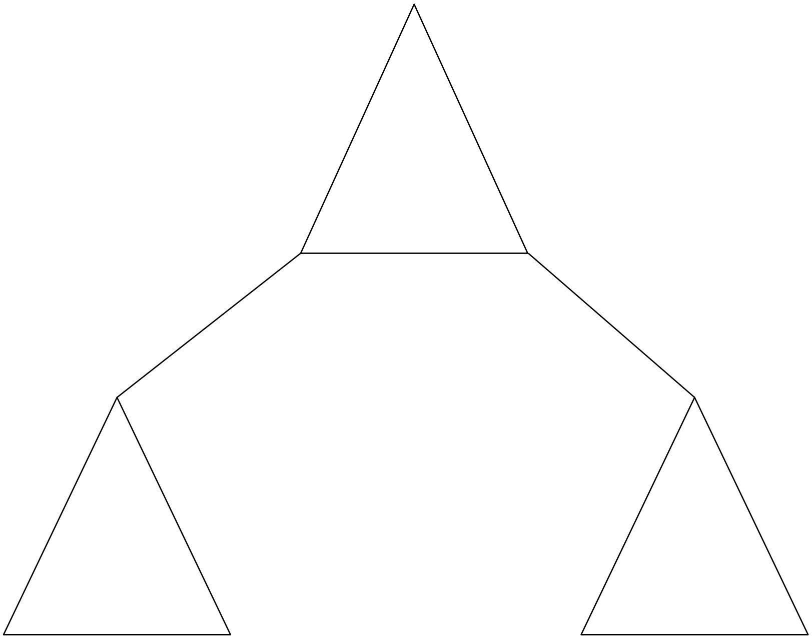


Figure 1-4: A tree diagram in Active Directory.

Regardless of whether you actually see a tree when you turn the book upside down, the term *tree* is one that you use often in discussing directory services. And the arboricultural (it's a real word — honest!) terminology doesn't stop there — as you discover when you find out more about Active Directory.

When you add domains to an Active Directory tree, you automatically create *transitive* trust relationships. Transitive trusts extend the relationship between two trusted domains to any other domains that those two domains trust. These trusts are bidirectional and enable users in one domain to access resources in the other domain. In an Active Directory tree, all domains are connected through transitive trusts, so a user in one domain can access any other domain in the tree.

You can also link trees or forests through *explicit,* or one-way, trusts. By creating an explicit trust between Tree A and Tree B, for example, you can specify that users from Tree A can access resources in Tree B, but users in Tree B cannot access resources in Tree A.

Forest

A *forest* is a logical grouping of trees that you join together in a transitive trust relationship, as shown in Figure 1-5. A forest has the following characteristics:

- Each tree in a forest has a distinct namespace.
- The trees in a forest share the same schema and global catalog. (I discuss schema and global catalog a little later in this chapter.)

Chapter 5 helps you determine when to create a tree and when to create a forest.

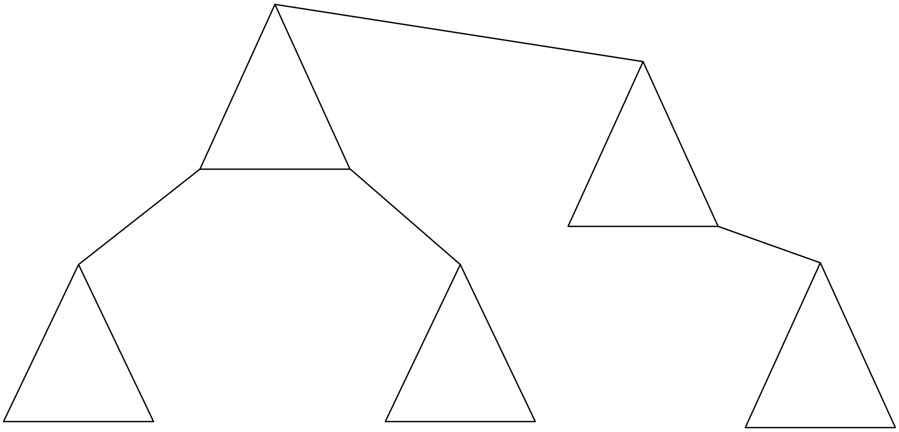


Figure 1-5: A diagram of an Active Directory forest.

Organizational unit (OU)

An *organizational unit* (or OU) is nothing more than a container within a domain. You use it to store similar objects so that they're in a convenient location for administration and access. Here are some of the objects that you store in an OU:

- Printers
- File shares (a folder located anywhere on the network that has been designated as *shared* so that others can access it)
- Users
- Groups (a grouping of users that can be jointly administered)
- Applications

While you plan your Active Directory structure, you also plan the logical structure of the OUs within each domain. Keep the following points in mind as you become familiar with OUs:

- You can *nest* OUs within each other to create a hierarchical structure.
- Each domain can have a hierarchy of OUs, or the OU hierarchy can be identical in each domain. You cannot, however, extend an OU across domains. OUs are always completely contained within a single domain.
- Structure OUs correspond with the business practices of your company. Earlier in the chapter, I talk about matching the logical structure to where employees work. OUs can help you organize network resources so that they're easy to locate and manage.

Many factors can influence your OU structure or model. An OU model might reflect the administrative model of the organization or the company's structure either by organizational chart or by work locations.

A domain that you name West, for example, represents your company's western region of the United States. This domain includes OUs that you name California, Washington, and Oregon, as shown in Figure 1-6. The California OU contains two nested OUs that you name San Francisco and San Diego. The Washington OU contains objects that you organize in OUs that you name Tacoma and Seattle. To ease administration by keeping things similar, the East domain follows the same conventions used in the West domain.

If you want, you can further organize the city OUs so that San Francisco, San Diego, Tacoma, and Seattle each contain nested OUs for user objects and printer objects.

REMEMBER

You can create transitive trusts between forests A and B so that all the domains in Forest A trust all the domains in Forest B and vice versa. Having forest-level transitive trust can greatly simplify your life!

Object

An *object* is any component within your Active Directory environment. (I talk briefly about objects in the "Active Directory is an information store" section earlier in this chapter.) A printer, a user, and a group, for example, are all objects. All objects contain descriptive information, or *attributes*.

Sites and Site Links

A *site* is a grouping of IP subnets connected by high-speed or high-bandwidth links. Sites are part of your network's physical topology (or physical shape), and each site can contain domain controllers from one or more domains.

During your planning stages for implementing Active Directory, you define a site topology for your environment. You use sites to optimize a network's bandwidth by controlling replication and logon-authentication traffic. (Chapter 12 tells you how to use sites to control traffic.)

By dividing the network into sites, you can limit the amount of replicated Active Directory data that you must send across slow WAN links. Domain controllers within a single site exchange uncompressed data because they're connected by fast links; domain controllers spread across different sites exchange compressed data to minimize traffic.

Of course, you can't just define sites and then expect the sites to start magically communicating with each other. You must define site links that connect your sites. These site links define how the replication and logon-authentication traffic flows between sites.

I devote Chapter 12 to a discussion of controlling replication traffic. But for now, just be aware that replication occurs whenever the domain controllers within a domain exchange directory database information. Updates or additions to the database trigger replication between domain controllers within a site.

You also use sites as authentication boundaries for network clients. Although any domain controller throughout the domain can authenticate a user, designating any but the closest one to do so isn't always the most efficient use of the network. After you specify your site boundaries, the closest available domain controller within the client's site authenticates a client logon. This setup minimizes authentication traffic on the network and speeds response time for the client.

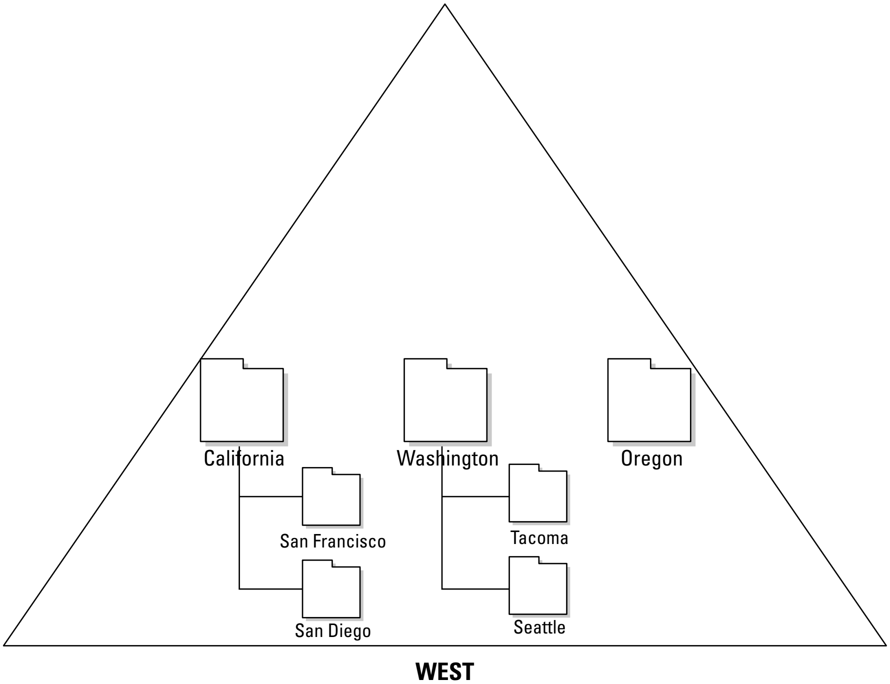


Figure 1-6: Nested organizational units (OUs) in Active Directory.

Object Identifiers (OIDs)

If you decide you want to create your own schema changes, you will need your own *Object Identifier.* Object Identifiers are dotted decimal numbers that the American National Standards Institute (ANSI) assigns to each object class and attribute. ANSI assigns a specific root identifier to a U.S. corporation or organization, and the corporation then assigns variations of its root identifier to the objects and attributes that it creates. For example, Microsoft's OID is 1.3.6.1.4.1.311, which maps to the following path:

```
iso.org.dod.internet.private.
   enterprise.microsoft.
```

The Active Directory schema

Along with the basic Active Directory components that I discuss in the preceding sections, you must also be familiar with the Active Directory *schema*. The schema contains definitions of all object classes (or object categories) and attributes that make up that object. That is, the schema is where the rules are about what kind of objects can be stored in the directory and what attributes are associated with each type of object.

Normally, an AD administrator doesn't make changes to the schema on a regular basis. The majority of the time, you modify the schema only when you're installing an application that uses Active Directory to store and retrieve information. One good example is Microsoft Exchange Server. A number of new attributes and object classes must be created and modified so that Exchange can work. But there can be instances where you might perform a schema modification on your own. For example, let's assume that all the employees of Steveco Corp. have a company-specific attribute (say, an employee number) associated with them and you want to put that information into Active Directory. There isn't any attribute in the default schema called `SteveCoEmpNum` so you must make the necessary changes to the schema to include this attribute.

At the time that you install Active Directory, you also install a base schema by default. This schema contains the object class definitions and attributes of all components available in Windows Server 2008. While your directory tree grows, you can extend or modify the schema by adding or altering classes and attributes as follows:

- You can create a new object class.
- You can create a new attribute.
- You can modify an object class.

- You can modify an attribute.
- You can disable an object class.
- You can disable an attribute.

(In Chapter 13, I show you how to do all the schema modifications shown in the preceding list.)

REMEMBER

By definition, an object must have defining attributes; each object has *required attributes* and *optional attributes*. Among the required attributes of any object are the following:

- Name
- Object Identifier (OID) (See the "Object Identifiers (OIDs)" sidebar.)
- List of required attributes
- List of optional attributes

Doesn't it seem odd that a list of *optional* attributes is a *required* attribute for an object? Of course your list of optional attributes could be empty!

Not just anyone can modify the directory schema. Only members of the Schema Administrators group can do so. The Schema Administrators group is a built-in group installed by default when you install Active Directory. The group is preconfigured with the appropriate privileges for performing particular tasks. As system administrator, you can assign particular users to this group by adding their user IDs to the group. (See Chapter 11 for the details on adding users to groups.)

WARNING!

Limit the number of administrators in your organization's Schema Administrators group to protect yourself against unintended results! Every organization should have a precise change-control policy that governs changes to the directory schema. The schema affects an entire forest, so any change is replicated to every domain in the forest. The potential for disaster is huge!

Domain Controllers and the global catalog

Domain controllers (DCs) are the servers that actually provide all the AD DS services as well as the actual storage of the directory data. The AD data on the DC is split into four types of regions or partitions:

- **Domain Naming Partition:** Each domain in the forest has at least one domain controller that is a member of that domain. The Domain Naming Partition is where the copy of all the objects within this domain controller's domain is stored. This information is replicated to all other domains controllers within the same domain. Every DC has a single domain naming partition because the DC can only be in one domain.

- **Configuration Partition:** This partition is used to store information that's needed across all domain controllers in the same AD forest. Within the configuration partition, the information about the physical environment, including site and site link definitions is held. This partition is located on every domain controller in the forest.
- **Schema:** Every domain controller in a forest has an identical local copy of the Active Directory schema stored in a schema partition. That way, every DC understands the rules of what objects and attributes can exist.
- **Application:** Application partitions are optional partitions that can be used to store data that is to be replicated between a set of domain controllers and used by an AD-enabled application. One good example is DNS, as I discuss in Chapter 4.

The replication of these partitions between the domain controllers is handled with a *multimaster model.* What does that mean? Multimaster model means that changes to these partitions can be on any DC and those changes will be replicated to every copy of that partition in the forest. Of course, there are some exceptions to this rule (you knew there would be!). Because of a schema's critical nature, only one DC in the forest has a writeable copy of the schema — the Schema Master. Table 1-1 summarizes these partitions and their replication method and scope.

Table 1-1 Active Directory Partition Replication

Partition Type	*Multimaster*	*Replication Scope*
Domain naming	Yes	Domain-wide
Configuration	Yes	Forest-wide
Schema	No	Forest-wide
Application	Yes	Domain controller–specific within the same forest

Windows Server 2008 AD DS introduces a new type of domain controller — a read-only Domain Controller, or RODC. I cover RODCs in detail in Chapter 6, but for now, understand that there's a special case when you can configure a DC where none of the partitions on a DC are writeable. You will see that RODCs are a great solution for deploying AD DS services in smaller, less secure locations.

Domain controllers provide two primary services to users: network authentication and directory object storage and retrieval. *Network authentication services* are provided by a DC through the Kerberos Key Distribution Center (KDC). In Active Directory security, Kerberos is everything. Every Active Directory user must get a Kerberos key at login. This key identifies the user to the network and controls what resources the user can access. In addition to the KDC, DCs provide the ability to store and retrieve the directory information in the partitions that the DC holds.

One other option on a domain controller that you need to understand is the *global catalog*. A global catalog (GC) is a searchable index that enables users to locate network objects without needing to know their domain locations. A partial replica of the Active Directory, GCs contain a list of objects in the forest but don't necessarily list all the attributes of every object in the forest. GCs aren't separate from domain controllers: They're an option that you can select on the DC's configuration. In other words, all GCs are DCs but all DCs aren't necessarily GCs.

The global catalog enables searches among trees in a forest. You can also use it to speed lengthy searches within a single tree. By default, the global catalog doesn't contain all the attributes of every object. The default global catalog configuration includes only those attributes that you're most likely to use for a search, such as a user's first or last name. Similarly, you can search the global catalog for all color printers instead of browsing through all the printers on the network.

The default schema settings determine which object attributes appear in the global catalog. All objects appear in the global catalog, but only a small subset of the objects' attributes are included. To add additional attributes to the global catalog, you have to modify the schema. (See Chapter 13 for additional information on modifying the schema.)

By default, the first domain controller that you create in a forest becomes a global catalog server. If the environment consists of multiple sites, you can optimize network traffic by creating a global catalog server in each site.

The *global catalog* is a service that runs on domain controllers. You manage the service by using the Active Directory Sites and Services snap-in for the Microsoft Management Console (MMC). The MMC is a Windows 2008 Server system file that you access by choosing Run from the Start menu and then typing **mmc**. From within MMC, open the Console menu, choose Add/Remove Snap-in, and then choose AD Sites and Services from the list that appears.

The DNS namespace

DNS (Domain Name Service) is the predominant name-resolution service on the Internet, so Microsoft chose to use DNS to translate host names to IP addresses in the Active Directory service. The DNS namespace is the single most important requirement for a successful Active Directory implementation, and the two are tightly interwoven. If you don't plan the DNS namespace appropriately, your Active Directory service is difficult to administer and doesn't adequately serve the user community.

A thorough understanding of DNS and of TCP/IP is essential for planning and implementing Windows 2000 and Active Directory. A good source of information is *TCP/IP For Dummies* by Cameron Brandon (published by Wiley).

I discuss in Chapter 4 that you must plan the DNS namespace before you can design the Active Directory. You use the DNS namespace design that you create (or one that already exists for your organization) to design a domain namespace for Active Directory.

If you're not using the Microsoft DNS service, you must use another DNS service that's compliant with RFC 2136 and RFC 2052.

Because It's Good for You: The Benefits of Active Directory

I don't know about you, but whenever Mom told me to eat my vegetables because "they're good for you," I still wasn't particularly motivated. I needed to know more about what that broccoli was actually going to do for me.

So maybe, like me with my vegetables, you need to hear about the real benefits you ultimately can realize if you bite the bullet now and make the management and design changes required by Windows 2008 and Active Directory.

Active Directory offers appealing features for administrators and end-users alike:

- Ease of management because of the centralized nature of the Active Directory database.
- Enhanced scalability (it can get lots bigger!) that enables the Active Directory database to hold millions of objects without altering the administrative model.
- A searchable catalog that enables you to quickly and easily search network resources. The network becomes less intrusive, enabling users to concentrate on their work rather than their tools.
- Active Directory forms an infrastructure backbone that many IT platforms and applications can utilize.

I encourage you to follow through all the planning and testing steps that I present in this book. With the right preparation, Active Directory can offer tremendous advantages for both you and your organization.

Chapter 2

Analyzing Requirements for Active Directory

In This Chapter

- Gathering business information and goals
- Gathering technical information
- Best practices for analyzing AD requirements

Before you can design Active Directory (AD) for a company, you have to understand the company and its needs. Active Directory can have a broad impact on a company and its operations. If you don't gather the needed information ahead of time and use it in designing the directory, you're setting yourself up for failure.

In this chapter, I discuss the business and technical analysis process that must be conducted before any AD design work can take place. I examine in detail each of the corporate business aspects that must be processed. With this information, you can answer the questions that come up as you move through the AD design process.

Why Gather Information?

For successful implementation of Active Directory, you must spend an appropriate amount of time in the information gathering and analysis phase of the project. Not spending enough (or, in some cases, any) time gathering and analyzing this information is probably the number one reason why Active Directory implementations fail. What do I mean by a failed AD implementation? Most failed AD implementations can be characterized by one or more of the following elements:

- The Total Cost of Ownership (TCO) per user for support of Active Directory has increased over the previous IT architecture.
- User productivity has decreased because of the AD implementation.

- Active Directory wasn't implemented to complement existing or upcoming corporate business processes.
- The company has experienced an ongoing increase in its help desk call rate.
- The company has experienced a decrease in the reliability of the IT infrastructure.
- The mean time to resolve user problems has remained the same or increased.
- Management, users, or the IT support staff's satisfaction with the IT infrastructure has remained the same or decreased.

Characteristics other than those I mention might also be evident, but you get the idea. If you're responsible for implementing Active Directory and your implementation experiences one or more of these attributes, more often than not you can trace the problem to a poorly executed information gathering and analysis phase.

I state earlier in this section that the most common mistake is underestimating the amount of time this phase takes. Don't fall into this trap! Most companies implementing successful IT infrastructures experience that the planning phase can take one half to two thirds of the time it actually takes to implement the infrastructure. By reserving an appropriate amount of time upfront for gathering information, you greatly increase your chances of creating a design that results in a successful implementation.

You must collect information in two areas of assessment:

- Information about the company and its business
- Information about the technical environment, including the current IT infrastructure (if any)

In addition to conducting this survey, you must spend time developing a list of requirements or goals that you intend to achieve with this AD implementation. With this information, you can have a winning AD design.

Gathering Business Information

One of the first questions to ask a company interested in implementing Active Directory is, "Why?" The answer shouldn't be something along the lines of, "We need Active Directory because it has the newest technology" or "It's cool." Remember that technology is a means to an end — the end is the fulfillment of a business goal or objective. Don't implement technology for technology's sake. If you can't justify the implementation of Active Directory by stating one or more business goals, chances are good that you shouldn't introduce Active Directory into that environment. For this reason alone, you must gather this business information to document the company's justification for implementing Active Directory. These justifications can guide you through the AD design process.

Exactly what information do you need to gather? You're looking for information to help you make decisions as you move through the AD design process. You don't want to design in a vacuum; in other words, you don't want to design an AD infrastructure that doesn't take into account any information about the environment. Most IT engineers (myself included) are guilty of this practice to one degree or another, and it's dangerous — especially where Active Directory is concerned. The scope and impact of Active Directory is just too broad to design it without this information.

Before you can go somewhere, you have to know where you are. Just like driving a car, you can't get to your destination if you have no idea where you're starting. Therefore, the first step is to document the business environment.

WARNING!

A common mistake during the information-gathering phase is failing to collect the data in an organized fashion. I go into the detailed information to gather in this chapter, but make sure that you store this information (for example, documents, spreadsheets, and so on) in such a way so that you can find it later. To spend time gathering all this good information and then be unable to find it later when you really need it would be a real shame.

Surveying the business environment

Every business is different; therefore, one Active Directory design can't meet every company's needs. When designing Active Directory for a company, you need to be intimately familiar with that company's business environment. With previous generations of Network Operating Systems (NOS), such as Windows NT or Novell NetWare, designers generally had to take the approach (sometimes unconsciously) of conforming the company's processes to meet the NOS's features, when they should have conformed the NOS to aid the company in conducting its business. Designers took this approach because many of the older NOSs didn't have the capability to scale to the size of many medium- to large-enterprise-sized corporations. With Active Directory's *scalability* (the capability to host millions of objects) and its flexibility, architects can now design a directory service that can meet the company's needs and provide some real benefits.

So before you design the service, know the company that you're designing Active Directory for. You need to understand the ins and outs of the company; otherwise, you can't possibly hope to create a directory service that provides real value.

The company's business model

You need to understand the business your company is in. More than anything else, the company's mission statement should clearly convey this. Of course, you have to delve deeper than that. Here are some of the aspects of the business that you need to have a clear understanding of:

- **Corporate priorities:** What are the company's current objectives? Why are they in business? Investigate if there are any current corporate initiatives that could potentially affect your design either during or immediately after deployment of Active Directory. Pay close attention to these initiatives — especially if they happen to conflict with the implementation of Active Directory.
- **Corporate growth:** Is this company growing, either in the number of employees or in scope of business? Are there any potential mergers or spinoffs in the works? These things could affect your design or, at the very least, force you to consider making the design flexible enough to address these changes.
- **Relevant laws and regulations:** Understanding what laws affect how this company conducts business is important, especially in multi-national and government-regulated businesses. For example, some countries have laws on their books that make exporting employee information outside the country without prior approval illegal. This restriction could definitely affect how AD will replicate data and where that data can be stored. Remember that ignorance of the law is not an excuse!
- **Company's tolerance of risk:** How much risk can the company take? Remember that this company has customers to serve. What happens if, because of the implementation or design of Active Directory, the level of customer support changes? What level of support is the company contractually obligated to provide? Understanding these risks can help you design Active Directory so that it provides fault tolerance in the areas necessary to reduce risk.
- **Information technology costs:** How much does the company spend (in both people-power and money) to provide IT services to the company? Is the percentage of IT costs compared with total revenue acceptable, or are changes required so that you can reduce costs? Reduction in IT support costs should be one of your objectives in implementing Active Directory.
- **Business processes:** What business processes does the company use to conduct its daily operations? How does the IT infrastructure support those processes? Do these processes have potential for improvement? Understanding these issues might help you justify your AD implementation by being able to show some concrete savings from improved business processes.
- **Service and product life cycles:** Are there any life cycles related to the services or products that this company produces? These cycles could affect the number of employees, network traffic, frequency of changes to the

directory, and so on. These life cycles also tend to create lines of communication. Active Directory can be structured to support these communication lines.

- **Decision making and information flow:** How are decisions made in the company? Are they made from a central location, such as corporate headquarters, or does each location work independently of the others? You should understand how information is communicated within the company and design Active Directory to support these processes.

One of the downfalls here is that some of this corporate information is probably going to be difficult to pry from company executives. Sometimes, the secrecy is for good reason, such as a confidential upcoming merger; at other times, you might just have difficulty nailing down an executive long enough to get this information. Make sure that you communicate the risk and potential added costs the company could incur by not releasing this information.

The company's structure

Understanding how the company is organized is crucial to a successful AD implementation. During the information-gathering phase, you should capture two views of the company: an organizational view and a functional view. The *organizational view* is the typical organization chart (shown in Figure 2-1) that most companies create to communicate how employees fit into the hierarchical management structure. Through this structure, you can see how information is communicated from top to bottom and get an idea of how the decision-making process works. Because communication and decision making are essential parts to any business, you need to understand how these parts work so that Active Directory can support, not hinder, these activities.

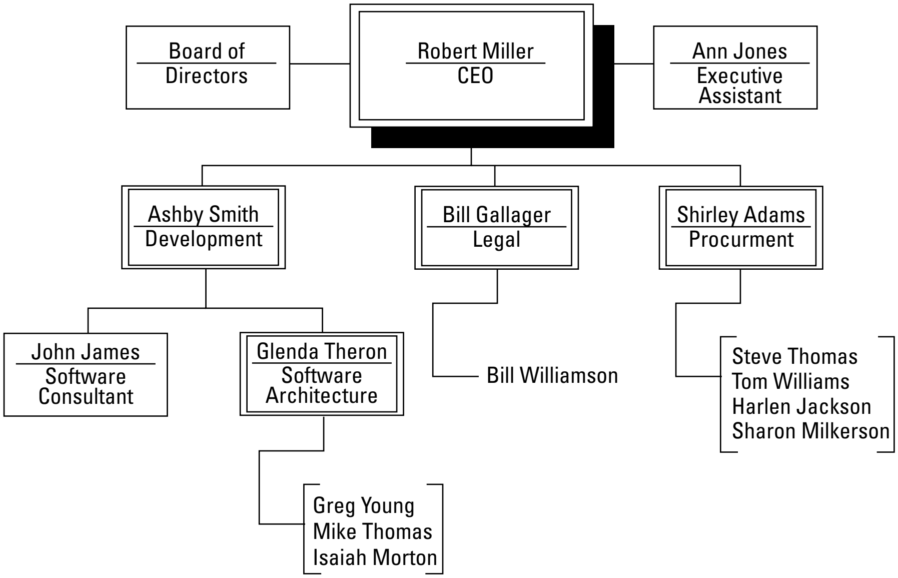


Figure 2-1: A typical organization chart.

TIP

Although recording the organization chart is important, if that chart is about to change (especially during or before deployment of Active Directory), you should attempt to get a new chart quickly or, at the very least, gain an understanding of the organization's changes.

Understanding the organization chart is necessary; however, the chart doesn't always convey how things really work in a company. You also need to understand the *functional view* — how the company is divided into different functional areas. This understanding is necessary because the organizational view might not clearly communicate the business functions. The functional view is created by charting the operations that must occur within the company for the normal day-to-day business to function correctly. These functions are then examined to understand how they interoperate with each other. The output can be various formats, including flowcharts or block diagrams (shown in Figure 2-2). The format isn't important — you're trying to gain an understanding of how functional groups communicate with each other. That knowledge can help you as you plan your AD design.

The geographical model

Understanding the geographical model the company fits under allows you to get an initial idea of the magnitude of the AD design that must be created. Most companies fit one of the following geographical models:

- **Regional:** A small- to medium-sized company located within a single country. The number of offices involved generally ranges from one to ten. In most cases, many business and IT functions are centralized because the company is not large and has only a small number of locations.
- **National:** A medium-sized company located within a single country that has a larger number of offices. Because of the larger number, some functions become decentralized. You might also have to deal with more political issues among the various locations within the company than in the regional model.
- **International:** A medium- to large-sized company located in multiple countries. An international company is likely to be the most challenging geographical model to create an AD design for. The challenge stems from cultural, language, and political issues, as well as any laws that might affect the design within each country where the company is located. Also, although the company still has a single headquarters, many business and technical functions probably are decentralized. Your AD design should take into account these decentralized areas.
- **Branch Office:** Typically a medium- to large-sized company with one or more major locations and a number of remote offices that work autonomously. These branch offices can be located within a single country or internationally. Each branch office must communicate with the headquarters. Communication among the branch offices is likely limited.

Because each branch office can operate as its own separate, independent entity, pay careful attention when you're designing Active Directory to support this decentralized operation.

- **Subsidiary:** Usually, a medium- to large-sized company that has multiple external identities, each operating as a separate subsidiary company. Each of the subsidiaries operates independently of the others and communicates only with the headquarters.

Note: Your company might not cleanly fit into any of the models or it might fit into a combination of the models. The idea is to get an understanding of where the company is located and how these locations work together.

Business relationships

You also need to understand any standing relationships that the company may have with other partners, vendors, and customers. Here, you're looking for processes that are using (or could use) the IT infrastructure. This can affect how you deploy Active Directory Domain Services (AD DS) in that you might need to deploy an additional instance of AD in the company's extranet network or even possibly look at creating a federated relationship with the external partner using Active Directory Federation Services (see Chapter 9).

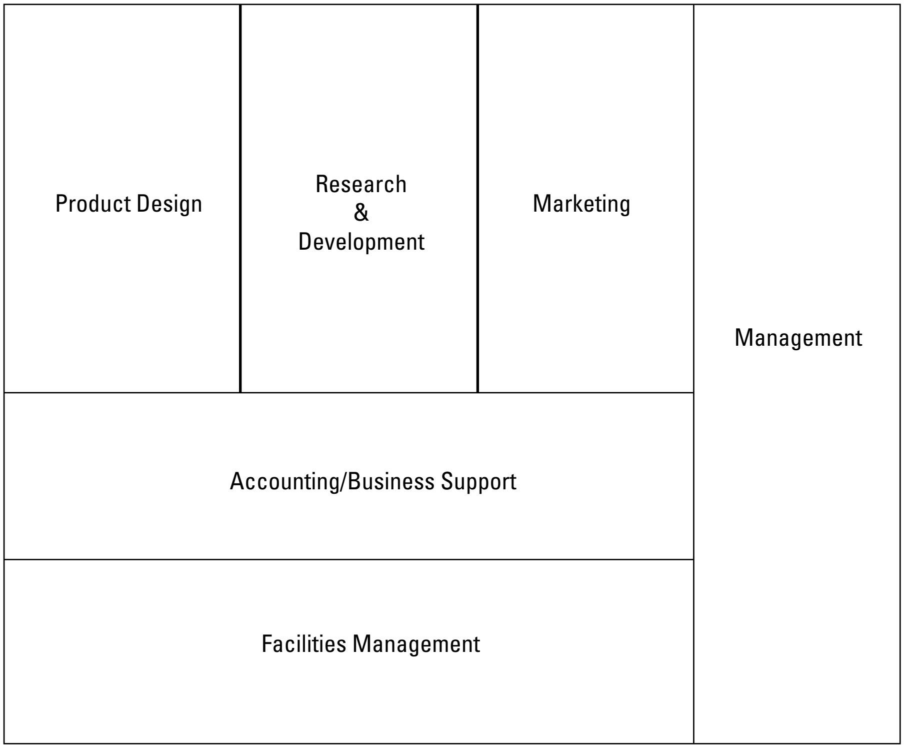


Figure 2-2: A functional block diagram.

The administrative model

The number one cost of any IT system (both servers and desktops) is its ongoing administrative support. Therefore, understanding how Active Directory and the Windows Server OS are supported is vital. Because most aspects of Active Directory are geared around decreasing the administrative load, you find the administrative model, more than anything else, drastically affects the final AD design.

TIP

Active Directory should always be designed with the administration model in mind over any other considerations.

Just as for the other business and technical aspects, you want to design Active Directory to support the administrative model instead of conforming the administrative model to fit Active Directory. Architects can potentially gain some significant cost savings if they're allowed to create a support structure that can administrate Active Directory in an efficient and productive manner.

The main question to ask is how to structure the AD administrative model. You have three possibilities:

- **Centralized:** A centralized administration model implies that a single organizational group is responsible for providing AD support. The personnel are usually centrally located as well. The benefits here are that users have a single point of contact for support and that the accountability is easy to figure out. A downside to the centralized model is that trouble tickets might take longer to resolve because all support calls are handled by a single help desk. Although this is a common model in smaller companies, many larger companies are also moving to this model because centralizing support has cost savings potential.
- **Decentralized:** By following a decentralized model, multiple administrative groups are responsible for support. This support can be broken out at both an organizational level and an AD partitioning level. Therefore, you might have a group in one region that is responsible for administrating a portion of Active Directory that corresponds to either that group or region. A benefit is that user requests can usually be satisfied more quickly. The downside, however, is that you must carefully coordinate the activities of the different administrative groups to ensure that no support conflicts occur. The decentralized model is common in larger companies.
- **Combination:** Your support model might not cleanly fit into either of the categories. You might have centralized the support of certain network services and decentralized others. Although this model can work, it also can be difficult to deal with if the IT infrastructure isn't designed for delegating support from the centralized teams to the decentralized ones. A great thing is that support of this model is one of the areas where Active Directory really shines.

In conjunction with understanding the structure of the administration model, the other question to ask is whether to manage the environment tightly or loosely. Will the client desktop configurations be locked down so that users cannot change configurations or install, upgrade, or remove any applications? Or, will the environment be loosely managed so that users can reconfigure client desktops as they see fit. The answer depends on the corporate environment and the needs of the users versus the cost of supporting that model.

You should also understand how the IT administration is funded. Does the funding come from company overhead, or is each group billed internally? Be sure that you can appropriately staff for the support of Windows Server 2008 and Active Directory. You should also understand how decision making and change management are handled.

In some cases, companies outsource their IT management to other companies. You have to work with the company providing IT services closely during the AD design process to make sure that it can support the design.

After you determine the current model, ask whether the model will support Active Directory. The implementation of Active Directory is a perfect opportunity to reevaluate your current support model and determine whether a change should be made for efficiency's sake. If a new model will be created, make sure that you thoroughly document it so that you can implement Active Directory to fit that model.

Determining business goals

After you gather information about the existing and planned business environments, find out exactly what goals or requirements you must meet for the AD design to be considered successful. Usually, a Chief Information Officer(CIO), Chief Technology Officer (CTO), or a group of managers responsible for the costs of the project defines the goals of the implementation. Identifying the *project owner* (the person signing the checks for the project) early in the process is critical. The project owner is ultimately the person who should be identifying the business goals that the implementation is to achieve. If you don't get these requirements from the project owner, you risk developing a system that doesn't meet the business objectives the check-signer has in mind.

You need to ask two goal-probing questions at this point:

- **"What do you need?"** Give the manager the opportunity to communicate his expectations for what the AD design will accomplish. Take this opportunity to make sure that the requirements are reasonable and attainable.

- **"What do you need that you donít know you need?"** Sometimes the role of an AD designer requires that you also put on your consulting hat. By analyzing the information you've gathered, you might discover other requirements that can generate additional benefits or cost savings that the manager hasn't thought about.

These requirements can come from several areas, including the following:

- **Business process improvement:** These goals can include items such as improving business processes and accommodating new ones.
- **Increased user productivity:** Users should be able to complete work in a shorter amount of time. This requirement can be measured by a decrease in help desk tickets.
- **Cost reduction:** Most companies are always looking for how to reduce the TCO of their IT environment. If this is a goal, make sure that you have thoroughly documented the current IT costs.

After you gather these requirements, it's a good idea to generate a document that captures these goals and categorizes them by scope and priority. Having this information during the design process is important because you might find that you have to eliminate certain lower-priority goals to meet the higher-priority ones.

Also, try to stay away from nonspecific goals. For example, if you want to reduce IT support costs, specify an actual value. Otherwise, after the implementation is complete, you won't be able to determine whether you met that goal.

Gathering Technical Information

Gathering business information is important; however, you also have to focus on the IT technology aspect of the company. This requires examining the environment based on its present and future state, particularly noting all anticipated changes to the environment. After collecting and analyzing this information, you can then use it to create the AD design. The technical information that you gather is mostly in two categories:

- A technology survey of the current IT environment
- An established set of technical goals that the AD implementation is to achieve

The technology survey information gives the architect a good idea of the technical environment into which she plans to implement Active Directory. By combining this information with the technical goals, the architect can detect problem areas that need adjustment (such as the network topology) before implementing Active Directory.

Chances are good that some of the necessary technical information might not be readily available. For this reason, build a planning team of individuals from different IT arenas that have access to all the necessary information.

Surveying the technical environment

The ultimate goal is to implement Active Directory correctly, so the architect needs to understand the IT environment before attempting to design a new environment. With this information, the architect can match Active Directory to the technical needs of the company.

Necessary information includes

- The physical and logical network architecture
- An understanding of how the company's employees and work processes match up to the physical and logical network architecture
- The likely impact Active Directory might have on the company's established work processes and network infrastructure

The demographics

You need to understand where the company is located and gather information about each location. I use the term *demographics* to describe this information.

To determine what the company's demographics are, you need the following information:

- All the company's physical locations, listed by address
- A list of the groups (by organization chart and functional chart) at the locations and the number of users in each group
- The number of users who work remotely and the number of users who work at home
- The employee turnover rate by location
- The growth and reorganization potentials
- Other attributes of each location:
  - *Does the company own the building?*
  - *Can the building facilities support computer equipment locally?*
  - *Are there any laws or regulations at this location that might affect the deployment of Active Directory?*
  - *Are there any critical functions or personnel at this location that have special support requirements of the IT infrastructure?*

You primarily use demographics in the design process to ensure that Active Directory is properly sized to each location. The process of sizing Active Directory determines at what locations to implement AD servers and services, such as domain controllers and global catalog servers, and whether the network at the locations can handle the traffic generated by the services. You can also use the demographics to determine that no servers need to be deployed at the location, perhaps because of an insufficient number of users or low network bandwidth.

Keep in mind that you need time to gather this information — especially in a larger company. Also, keep in mind that in large companies, by the time you gather the information, the numbers at one or more of the locations might have already changed. Don't sweat this too much. Unless there's a radical change, the numbers you have should be good enough for you to size Active Directory to the location.

TIP

You want to view this data in various ways and potentially manipulate the data to answer other questions during the design process. I strongly suggest that you place this information into a spreadsheet or database format.

The network infrastructure

Because Active Directory and Windows 2000 are major components of the company's IT strategy, creating an AD implementation requires an understanding of the networking environment that supports the directory structure. Because Active Directory is a networked application, its design must take into consideration the network it's implemented within. This network infrastructure has both physical and logical aspects that must be analyzed.

Assessing the physical networking infrastructure

First, you need a diagram of the wide area network (WAN) that exists within the company. Your goal is to develop a diagram that shows how the locations you identified in the demographics are connected. Figure 2-3 shows an example of a physical network diagram.

Figure 2-3 gives you an idea of an enterprise-level diagram that shows how the locations are wired. The network speeds on the WAN connections are shown on the diagram. However, one important piece of information is missing. Although the network speeds on the links are listed, the net available bandwidth on these links isn't. Although the speed information is important, it's even more crucial that you have the available bandwidth of those connections. Net available bandwidth is calculated as:

```
Net available bandwidth = Speed of network link * (1 - %
          Average network utilization)
```

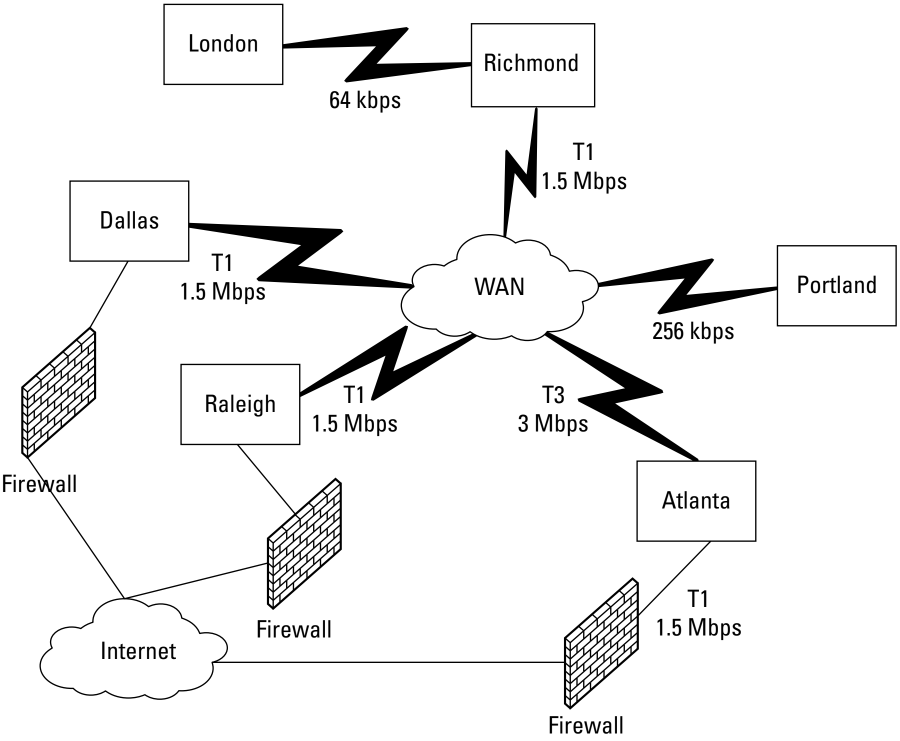


Figure 2-3: An example of a physical network diagram.

If you are using leased network links from a telecommunication carrier, the carrier should be able to provide this information to you. If have dedicated network lines and you don't have the right network monitoring tools in place to obtain this information, consider getting them. Knowing how much available bandwidth is on the network links is critical to properly designing the physical side of Active Directory. Without this information, you run the risk of designing Active Directory that exceeds the capabilities of the network.

TIP

After you have the total and available bandwidth numbers, combine this information into the same file with your demographics information. Many of the decisions you make in the design process require weighing the network speeds and capacity against the number of users and the activities at that location. Having this information in one file is very useful.

In addition to the WAN diagram, the following list details important network infrastructure information that is useful in creating an AD design:

- **LAN diagrams:** A network diagram of each location can give you an idea of where servers might be placed at these locations.

- **TCP/IP subnets:** Add a list of the TCP/IP subnets on the network to the demographics file. The physical design of Active Directory includes the creation of AD *sites.* A site is a collection of well-connected IP subnets (I discuss the term *well connected* in Chapter 6). By adding the subnets into the demographics spreadsheet or database, you can quickly see how the subnets can be grouped into AD sites.
- **Data pattern and performance needs:** To implement Active Directory, you need to understand data patterns and performance needs within the physical network. This information helps determine the best time to conduct AD replication. The company might have scheduled daily or weekly job cycles that make certain demands on the network. What you want to ensure is that AD replication doesn't affect these cycles. Therefore, it's important that you know when these jobs are executed and how and where they affect the network.
- **Hardware and software inventories:** Access to hardware and software inventories — particularly when working on a migration — is important from a reuse standpoint to understand what hardware you have available. Additionally, you need to ensure that software licensing compliance is maintained during implementation and afterward.

Assessing the logical network infrastructure

Along with the physical network information, you need to understand the network infrastructure from a logical viewpoint. You can see how the architecture is designed from a conceptual point of view by taking a logical view of the infrastructure. For example, if the company has a Windows NT network, you might develop a logical diagram similar to Figure 2-4.

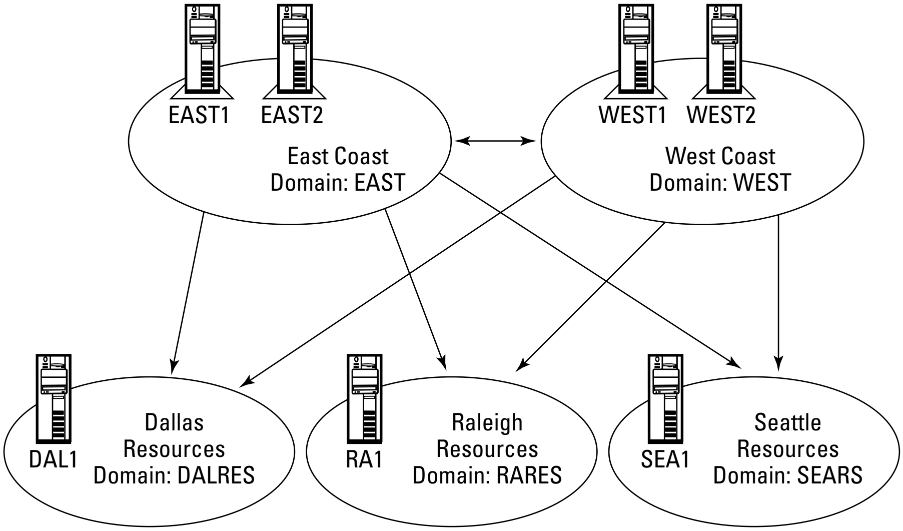


Figure 2-4: A logical network diagram.

From this logical view, you can

- See what NT domains exist and what their names are
- See the number of servers and their names
- View the trust relationships that exist between these domains

Of course, this isn't the only type of logical network diagram that you can create. You can diagram any network-provided service that is distributed across the network, such as DNS or DHCP, to show how the service has been implemented, what servers exist, and how these servers interact. In addition to charting these services, document their configurations upfront. If you know how the current services are deployed, you avoid risk later if you have to upgrade these services and encounter problems. Also, you might recognize services that can either be replaced by or integrated with Active Directory.

Examine what security standards are in place in the environment. Ask whether these standards apply to Active Directory or need updating. Spend time in this area because Active Directory and Windows Server 2008 can have a significant impact on security standards.

TIP

Comparing the logical and physical architectures against each other is a worthwhile exercise. Very often, you find that the logical architecture was designed around the constraints of the physical architecture. This was a common way of designing Windows NT systems. Don't fall into this trap with Active Directory!

Existing directories

Because Active Directory is an X.500-compliant directory service, ask about other preexisting directory services and determine what interaction between AD and these other directories is required. You might also need to consider whether Active Directory can replace those systems, or if synchronization of data between these other directory services and AD is necessary. This might also point you to deployment of Active Directory Lightweight Directory Services (AD LDS) as a replacement for any of the non-OS directory services.

The impact of Active Directory

Because installing Active Directory introduces a new directory service into the IT environment, the architect needs to determine what effect Active Directory is likely to have on that environment. Introducing any significant change into an IT environment always entails an element of risk. By examining these potential risk areas, you can reduce the danger.

Potential risk areas include

- **Existing systems and applications:** Examine what applications and tools are in place in the IT infrastructure and the effect that AD could have — both positive and negative. Particularly, pay attention to Windows server applications that might need upgrading because of the migration to Windows 2008 and Active Directory. Also, try to identify applications that might benefit from the availability of an X.500-compliant directory service like AD DS or AD LDS.

- **Existing and planned upgrades and rollouts:** Are there any IT implementation projects ongoing or planned? If so, assess how an Active Directory installation affects that project. If there is an impact, spend time prioritizing these projects before going any farther.
- **Technical support infrastructure:** How is Active Directory going to affect your IT support staff? The company probably has to invest time and money in training personnel on Active Directory and Windows 2008. If the design is done properly, you might free up some of the IT staff to work on all those other pending projects you've been meaning to get to!
- **Existing and planned network and systems management:** What existing or planned tools are in place to perform network and systems management? If any of the tools are specific to NT, then it's likely that you need to upgrade. Also, if you don't use any tools, you probably should spend time assessing whether investing in these tools is warranted.

The client desktop environment

Spend time understanding and documenting the desktop environment. Active Directory can have a significant effect on the user's desktop experience, but you need to understand the current environment before you can improve it. Gather client desktop information regarding:

- **The end-user work needs:** How do the users make use of their client computer? You should understand what applications the users use and how those tools are used to accomplish work. Also, find out what tasks the end-users find difficult to accomplish in the IT environment and try to identify how Active Directory might improve that process.
- **The end-user technical support needs:** The average technical expertise of the end-user varies from company to company. This affects the end-user need for technical support. In a company with many nontechnical users, it makes sense to consider locking the desktop configuration settings to keep users from accidentally changing things. By doing this, the company can realize an IT support cost savings. Comparatively, locking the desktop might be a bad idea for some users, especially power users, because it could negatively affect their work. If you have both groups in your company, it makes sense to treat these users differently and to structure Active Directory to match the situation.
- **The client computer environment:** This environment includes both the desktop hardware and the software applications installed locally. Pay close attention to hardware and software upgrade issues that might arise when you move to Windows Server 2008 and Active Directory. You should also understand where these client computers would be used. For example, you might have a sales force that uses laptops on a regular basis as they travel.

TIP

Don't just talk to the IT management or IT support staff to gain an understanding of the desktop environment. Talk to the actual end-users (both the average ones and the power users). By doing so, you can discover additional information that the IT staff is unaware of.

Determining technical goals

Document the company's technical goals as they relate to implementing Active Directory. A CIO or IT manager within the company usually defines these requirements. This person may not necessarily be the same manager who defines the business goals. Although you have both business and technical goals, keep in mind that the technical goals should be developed to support the business goals.

Examples of technical goals include the following:

- **Systems management goals:** What systems management features do you expect to gain from Active Directory? Through the application of security and application policies (using Group Policy objects), you can achieve such tasks as software distribution and installation and desktop lockdown. Document in detail what you expect these services to look like. (I look more at group policies in Chapter 14.)
- **Performance goals:** Establish goals regarding the speed and reliability of the IT system.
- **Security goals:** Define how Active Directory should meet or exceed the IT security policies that are in place (if any). Through Active Directory and Windows Server 2008, you can implement many new features including public key infrastructure and smart card technologies.

TIP

Document these technical goals and prioritize them in case of a conflict later in the design process. Also, remember to keep these goals specific so that you can later determine whether these goals have been met.

Best Practices

While you work through the analysis phase in preparation for implementing Active Directory, keep the following practices in mind:

- **Implement Active Directory to meet a set of business goals, not just for the sake of implementing it.** Get those goals from the project sponsor and no one else. It's the only way to ensure that the project will meet the sponsor's objectives.

- **Never design Active Directory in a vacuum.** Always spend a significant amount of time in the information- and requirements-gathering phase.
- **Don't underestimate the amount of time that you should spend gathering information.** With the appropriate amount of information at the correct detail level, you stand a much better chance of creating a successful directory design.
- **Delve deeper than the information you find on the company's organization and functional charts.** These documents can be helpful, but remember that your goal is to understand how the company and its workgroups interact with each other.
- **Design Active Directory to fit the administrative model — it's the most important piece of information you can obtain.** Active Directory is probably the most important tool that you will use to administrate a Windows server/desktop environment, so designing it to meet the support model eases the requirements for administration and improves the end-user's IT service level.
- **Quantify your goals into measurable terms.** This way, you can easily determine success or failure in meeting the requirements.
- **Be organized as you obtain and document the information you gather.** You refer to it often while you go through the design process.
- **Don't only gather information about the company's environment; determine whether there are plans for change.** Gathering data on planned changes in the company is just as important, if not more so.
- **Gather accurate demographic information.** You use demographics to determine the best way of sizing the environment to the company and to decide where servers and services should be placed in the physical network.
- **Examine the IT infrastructure from both a physical and logical perspective.** You can identify areas that Active Directory can improve by comparing these two views.
- **Make sure that you get the net available bandwidth on the network links and not just the link speed.** Remember that using a 256 Kbps link at 50 percent capacity is faster than using a T1 link at 95 percent capacity.
- **Document the desktop environment.** Understand how the desktops are used and you can identify which processes Active Directory should support and possibly improve.
- **Make sure that technical goals are clear.** You can easily determine whether the goals have been met if the requirements have specific measures.

Chapter 3

Designing an Active Directory Implementation Plan

In This Chapter

- Creating an AD implementation plan
- Putting together the AD planning team
- Creating the AD planning documents
- Tracking project implementation
- Creating the AD design

Throughout this book, I present various Active Directory design principles that you need to understand if you're going to roll out Active Directory successfully in your company. Unless you understand how the pieces fit together to form an overall AD design, you won't be able to actually implement that design. Even if you're able to roll out Active Directory to a company without a well-thought-out plan and rollout schedule, you're likely to end up with a directory service that took longer than you planned to implement and doesn't meet your company's needs.

So before I get into the real technical information in this book, in this chapter I discuss forming a planning team, creating planning documents the team needs *before* implementing Active Directory, creating an AD project plan to schedule and track the individual tasks required to implement Active Directory.

Why You Need an Implementation Plan

As the old axiom goes, "No one plans to fail, they just fail to plan." This is never truer than when implementing an Active Directory design. No one wants to fail, but so many times the failure to create and document a plan is the core reason that a project fails.

So why do you need an Active Directory implementation plan? As with most IT projects, an implementation plan can help you

- **Develop a common understanding:** The main reason you need to create an implementation plan is to develop a common understanding between the IT implementers and management that's footing the bill for the project. The technical implementers often view the project goals differently than management does. The implementation plan also builds a common understanding between multiple IT departments in large companies. This is especially critical in multinational companies where IT organizations are spread across the globe. By developing an objective, detailed implementation plan and making that plan readily available to everyone, you can reduce and perhaps eliminate confusion and misunderstandings beforehand instead of experiencing them in the middle of your Active Directory rollout.
- **Mitigate risk:** All projects have inherent risks. A *risk* is a scenario that could adversely affect the scheduling and eventual outcome of the project. Include a list of risk scenarios in your implementation plan and think about how you can counter those scenarios, should they arise, so that the project can continue.
- **Determine a budget:** Your project is likely going to require funding. Without a good implementation plan that specifies resource needs in terms of labor, training, computer hardware, and software, and a project plan that shows when you need those resources, developing a budget for the project is impossible.
- **Develop a goals road map:** You must identify the company's business goals and technical goals (Chapter 2 covers this subject). Particularly with large, complex projects, you can easily lose sight of these goals. Your implementation plan needs to document these goals clearly and show how the Active Directory design meets each of these goals specifically. As you go through the implementation, you can use these goals objectives as a compass to keep from being lost in the day-to-day issues that arise during the implementation.
- **Establish scheduling:** One of the most important components of your implementation plan is the development of a project plan. Within the project plan, you should establish a fixed set of dates — milestones marking when you're implementing particular parts of Active Directory. Although developing an initial plan is critical, it's equally important that you update this plan and make it available to all stakeholders involved with the project.
- **Support planning:** As with any new IT system, it's necessary to allow the supporting IT organizations to plan for the new system in terms of training and personnel. This can be particularly true with Active Directory, especially when the IT administrators don't have any previous experience with directory services or Microsoft server technology. Training can include offering formal classroom training, making books (including this one!) and magazines available to personnel, or offering hands-on training in a lab. Obviously, training isn't something that happens overnight; it requires time and money. Consider these needs in your implementation plan.

By creating an implementation plan, you generate the documentation that addresses each of these areas. This documentation includes the functional specification document, project plans, test plans, and contingency plans.

REMEMBER

Because this book covers Active Directory, I address the implementation plan from that perspective. However, Active Directory is just one part of Windows Server 2008. You're likely to develop an implementation plan for Windows Server 2008 with Active Directory as just one part (albeit a big part) of that plan.

Building the Active Directory Planning Team

So, who does all this planning? Although you may be tempted to try to create designs on your own (especially the smaller ones), I recommend that you form a planning team and give it AD design responsibilities. That team should comprise a diverse set of individuals representing all areas of the company that are affected by Active Directory. By having broad representation on the planning team, your chances of getting design agreement across the company are good. In the ideal work of IT projects, politics shouldn't play a role. Politics, however, has a considerable influence, unfortunately. Diversifying the planning team minimizes the negative impact politics might have on the design and rollout of Active Directory.

Besides creating the design, the team's other responsibility is to get executive approval and sponsorship of the project. Without support from upper management for the AD implementation, the project has a high risk of failure. This management support comes in several ways, including financial support and technical direction support. Obviously, the financial support is critical to any IT project. But, support for the technical direction that the design creates is just as important, if not more so. The company's Chief Information Officer (CIO) or Chief Technology Officer (CTO) should have a clear technology policy and direction in mind for the company, and Active Directory should fit into that policy. If this isn't the case, the planning team should work with the CIO and CTO so that Active Directory does have a role to play in the company's technology direction before designing Active Directory.

TIP

Early in the process, you have to gather the necessary business and technical information from across the entire company (I cover this subject in the Chapter 2). Because the planning team should comprise individuals from across the company, it's a good idea to place the information gathering responsibility, as well as the design tasks, with this team.

The planning team plays a number of roles. Depending on the scope and size of the AD implementation, multiple team members could perform these roles, or an individual could provide multiple roles. The following list describes the planning team's roles:

- **Executive Sponsor:** Secures needed resources for the project and ensures that the rollout of Active Directory (and Windows Server 2008) is supported at an executive level. The sponsor also helps develop (and deliver to the executives) the vision of what the implementation is to achieve. Typically, a company's CIO, CTO, or someone who directly reports to the CIO or CTO, serves as the executive sponsor.
- **Visionary:** Develops a strategic direction for the company's IT infrastructure and for how Active Directory helps the company meet this objective. Again, this person could be the company's CIO or CTO.
- **Lead Design Architect:** Develops the actual AD design that supports the strategic direction developed by the visionary. A senior engineer with experience in designing IT systems typically performs this role.
- **Subject Matter Experts:** Possess intimate knowledge of the existing IT systems. Because Active Directory is likely to interoperate with Domain Name Service, e-mail systems, firewalls, network infrastructure, and so on, these experts must be involved with the planning of Active Directory. Also, the IT-support organizations providing support of Active Directory after the implementation should be represented within this role.
- **Testers:** Verify that the design meets the goals and vision developed by the executive sponsor and visionary. People in the testing role provide the facilities, test cases, and reporting that's necessary to perform this verification. This testing also includes conducting a small pilot deployment of AD involving selected end-users and IT-support personnel who can provide input as to how well the design works and meets their needs.
- **End-User and IT Support Trainers:** Provide training and coordinate timeframes needed to complete the training so that training time is incorporated into the overall iproject plan. Training must be provided to end-users who need to know how to use the system, as well as to IT administrators so they know how to best support the new system. Depending on the size of the company, this training can be done in-house or externally.
- **Project Manager:** Provides the creation and maintenance of a project plan, which tracks the project during its lifetime so that delays can be identified and resolved quickly. The project manager provides logistical support to ensure that each team works together and that the right resources are available at the right time. The project manager also coordinates the communication among the teams involved in the project, the executives, and the rest of the company.

Creating Active Directory Planning Documents

A big part of most projects is the development of documentation that records the designs and decisions that are used in the implementation. By writing down this information (either in documents or publishing on a Web site), you can communicate this information throughout the company, especially to the members of the planning team who need this information to complete their role in the project. The following sections describe the documentation that the planning team needs to create.

Business and technical assessments

As a starting point for the development process, you must conduct a business and technical assessment to gather the information you need (see Chapter 2). This information comes in the form of surveys about the existing business and technical environments including the business structures, processes, demographics, network topologies, existing IT systems, and so on. The other piece of information to gather through the assessments is a set of business and technical goals that the company wants to achieve through the implementation of Active Directory.

Vision Statement

Goals that a company records in its business and technical assessments need to be mapped into a strategic direction called a *vision statement.* A vision statement isn't necessarily an achievable or realistic goal. This statement provides an idealized picture of what a company intends to achieve by using the Active Directory design. By recording a vision statement, the planning team has an overall goal to work toward as the implementation progresses.

Requirements/scope document

After you create a vision of what you want your company to accomplish by implementing Active Directory, your next step is to create a *requirements/ scope document.* The purpose of this document is to set realistic expectations and prioritize the Active Directory features to implement. Because the vision is a "blue-sky" type of statement that might not be completely achievable, this requirements/scope document is what brings you back to reality.

To create this document, you must have a good understanding of the capabilities of Active Directory and understand how to utilize these capabilities to provide business and technical value. Analyze the vision statement and match the business and technical requirements that the vision statement entails to the features of Active Directory. After you create this mapping, work on these specific requirement statements to make sure that the requirements aren't vague but instead represent concrete, attainable objectives.

Gap analysis

An additional way you can develop a list of objectives for your AD implementation is to create a *gap analysis*. A gap analysis document compares the business and technical environments with the desired environment as described by the vision statement. Through this comparison, the "gaps" between these two environments become obvious. The planning team should focus on gaps when it chooses which Active Directory features to implement and use those features to "fill in the gaps." The gap analysis document also helps you prioritize the features to implement so that you address the bigger gaps first.

Functional specification

After identifying and prioritizing the Active Directory features to implement through the development of the requirements/scope document and the gap analysis, you need to document specifically how you'll use those features. This document, known as a *functional specification,* establishes an agreement between the planning team and the management stakeholders paying for the project on how to implement Active Directory. The functional specification document describes how Active Directory is to be implemented, although it doesn't address steps for creating that implementation.

The functional specification includes the design for the following items (the numbers in parentheses following each item is where you can find out more about this topic):

- Active Directory namespace and DNS design (Chapter 4)
- Active Directory forest/OU design (Chapter 5)
- Active Directory site topology design (Chapter 6)
- Active Directory service placement, including the domain controllers, global catalog server, and the operations masters (Chapter 6)
- Security within Active Directory (Chapter 14)
- The Active Directory schema (Chapter 13)
- Design of AD LDS, AD FS, AD CS and AD RMS (Chapters 8-10)

Your objective is to define an AD design that's detailed enough that you can use it in a set of scenarios based on the existing environmental information to see how well the design meets the company's needs. Unless you're lucky, or don't have a complicated set of requirements, you're likely to create multiple versions of this document before you reach a design that fits the company's needs perfectly.

TIP

If you haven't already guessed, most of the AD design work is done as a part of creating the functional specification document.

Implementation standards

Along with establishing design standards, you also need to establish a set of standards to follow when implementing the design and for the continued administration of Active Directory. These standards include the following items:

- **Active Directory naming standards:** Include naming standards for objects, such as users, computers, groups, Group Policy Objects, and printers.
- **Hardware builds:** Define a set of standards for the hardware that is used for domain controllers and other member servers providing related services, such as DHCP and DNS. Depending on the scope and size of the implementation, you may need various server sizes so that the servers are properly scaled to the environment.
- **Schema policy:** Develop a policy for managing the Active Directory schema. This policy dictates when schema modifications are justified and how to perform those modifications.
- **Security standards:** Develop a set of standards related to security, dictating such things as the password policy and the standard groups that are used to assign administrative authority in Active Directory. Within this document, you can also specify things that aren't directly within the bounds of Active Directory, such as how IPsec is implemented or NTFS permission settings.

Risk assessment/contingency plan

All projects have inherent risk. This is an unavoidable part of life. Risks are those inevitable events that have a negative impact on your project, such as delaying the rollout schedule or running over budget. You can't eliminate risk altogether but you can mitigate the impact of risk on your projects by developing a risk assessment.

The planning team as a whole needs to brainstorm to identify potential problems that could occur during the AD implementation. After the team develops this list, it should develop a contingency plan for each risk scenario to minimize the impact. Although a contingency plan doesn't prevent risk, if a problem occurs during the project, you can address the problem immediately and keep the impact at a minimum.

Tracking Project Implementation

In addition to the planning documents, you also need to create a project plan that tracks the progress of implementation. By creating this plan, you can control the following areas:

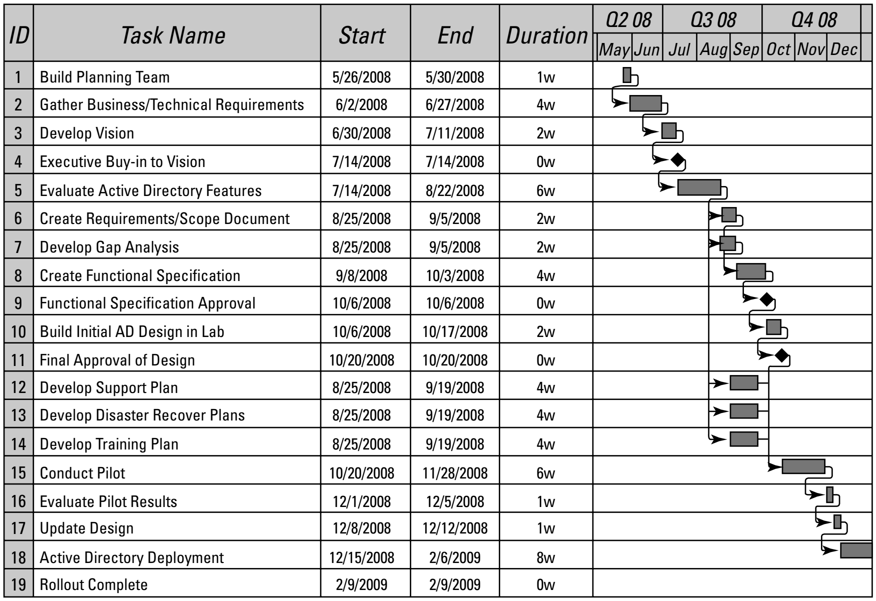
- **Task planning:** Your primary goal when creating a project plan is to record the tasks necessary to accomplish the goal of the project. Along with documenting these tasks, you also determine their sequence and their dependence on each other. By providing time estimates for each task, you can generate a timeline that helps you predict when each task needs to be done and how much time you need to finish it.
- **Resource planning:** You need various resources at different phases in any IT project. Resources can be labor, computer hardware and software, and network and power connections. You should record the resources you need for each task in the project plan. As you start to execute each task, you can easily identify the needed resources for that task and make sure they're available. Planning for resource availability helps ensure that you maintain the established timeline in the project plan.
- **Training planning**: One of the items that is commonly overlooked is the training needs. Primarily for this plan we are talking about the training needs for those that will be deploying and operating the AD environment. But depending on your situation, you may need to also consider the training needs of the end-users depending on how AD will impact them.
- **Timeline planning:** The project plan also serves as a measuring stick to determine whether the project is on schedule. Part of the planning team's responsibility in its project management role is to regularly compare the actual project schedule with the schedule in the project plan. If the project falls behind the project plan baseline, the planning team can be notified and can take action.

Recording the project plan into a document that can be shared with the entire planning team is important. That way, people on the team can anticipate what tasks they're responsible for and when they need to complete them. Figure 3-1 shows an example of a project plan.

A project plan should follow these general steps:

1. Build the planning team.
2. Develop the design documentation.
3. Conduct a pilot.

Don't overlook the idea of conducting a design pilot before rolling out the AD design to the entire company. A pilot verifies that the design actually works in the company's environment. Set a time frame for how long the pilot lasts. At the pilot's completion, survey the pilot users and support staff to see how well the design worked. If the design needs to be tweaked, you can do so with a minimal impact to the users because only a small subset of users is involved in the pilot. After you successfully complete the pilot and update the design documentation, you can begin the production rollout of Active Directory.



ID	Task Name	Start	End	Duration
1	Build Planning Team	5/26/2008	5/30/2008	1w
2	Gather Business/Technical Requirements	6/2/2008	6/27/2008	4w
3	Develop Vision	6/30/2008	7/11/2008	2w
4	Executive Buy-in to Vision	7/14/2008	7/14/2008	0w
5	Evaluate Active Directory Features	7/14/2008	8/22/2008	6w
6	Create Requirements/Scope Document	8/25/2008	9/5/2008	2w
7	Develop Gap Analysis	8/25/2008	9/5/2008	2w
8	Create Functional Specification	9/8/2008	10/3/2008	4w
9	Functional Specification Approval	10/6/2008	10/6/2008	0w
10	Build Initial AD Design in Lab	10/6/2008	10/17/2008	2w
11	Final Approval of Design	10/20/2008	10/20/2008	0w
12	Develop Support Plan	8/25/2008	9/19/2008	4w
13	Develop Disaster Recover Plans	8/25/2008	9/19/2008	4w
14	Develop Training Plan	8/25/2008	9/19/2008	4w
15	Conduct Pilot	10/20/2008	11/28/2008	6w
16	Evaluate Pilot Results	12/1/2008	12/5/2008	1w
17	Update Design	12/8/2008	12/12/2008	1w
18	Active Directory Deployment	12/15/2008	2/6/2009	8w
19	Rollout Complete	2/9/2009	2/9/2009	0w

Figure 3-1: A sample AD implementation project plan.

Creating the Active Directory Design

In the next few chapters, I cover the various aspects of designing an Active Directory Domain Services (AD DS) environment. You can approach creating your AD DS design in two ways: a physical-first approach or a logical-first approach. How you order the physical design and logical design process is primarily what differentiates these two design approaches. With a *physical-first* approach, you conduct the design of the physical aspects of Active Directory (DC/GC/Operations Masters placement, sites, and site links). The advantage here is that you work on what are the more complicated subjects first. After you complete the physical design, you can complete the namespace planning and logical design.

Table 3-1 shows the order in which the design tasks are completed with a physical-first approach and where in the book I cover the details of the individual design tasks.

Table 3-1 Physical-First Design Approach

Step	*Design Task*	*Description*	*Chapter*
1	AD Service Placement	Determine where to place DCs, GCs, and Operations Masters on physical network	Chapter 6
2	AD Site Topology Design	Create the AD sites and site links to allow replication to occur between the DCs placed on the network in Step 1	Chapter 6
3	AD Namespace/ DNS Design	Determine DNS domain names to be used for AD; develop standards for naming; design DNS system to support AD	Chapter 4
4	AD Forest/OU Design	Create the forest/domain/OU structure for Active Directory, using the namespace planning from Step 3	Chapter 5

A *logical-first* approach involves conducting the logical design first. This includes the design of the forest, domain, and then organizational unit structure as well as the DNS namespace design. The advantage here is that you can immediately address the logical portion of the design, which helps when the namespace planning or logical structure planning might take longer to work out.

Table 3-2 shows the order in which you complete the design tasks with a logical-first approach and where in this book I cover the details of the individual design tasks.

Table 3-2 Logical-First Design Approach

Step	*Design Task*	*Description*	*Chapter*
1	AD Namespace/ DNS Design	Determine DNS domain names to be used for AD; develop standards for naming; design DNS system to support AD	Chapter 4
2	AD Forest/OU Design	Create the forest/domain/OU structure for Active Directory, using the namespace planning from Step 3	Chapter 5
3	AD Service Placement	Determine where to place DCs, GCs, and Operations Masters on physical network	Chapter 6
4	AD Site Topology Design	Create the AD sites and site links to allow replication to occur between the DCs placed on the network in Step 1	Chapter 6

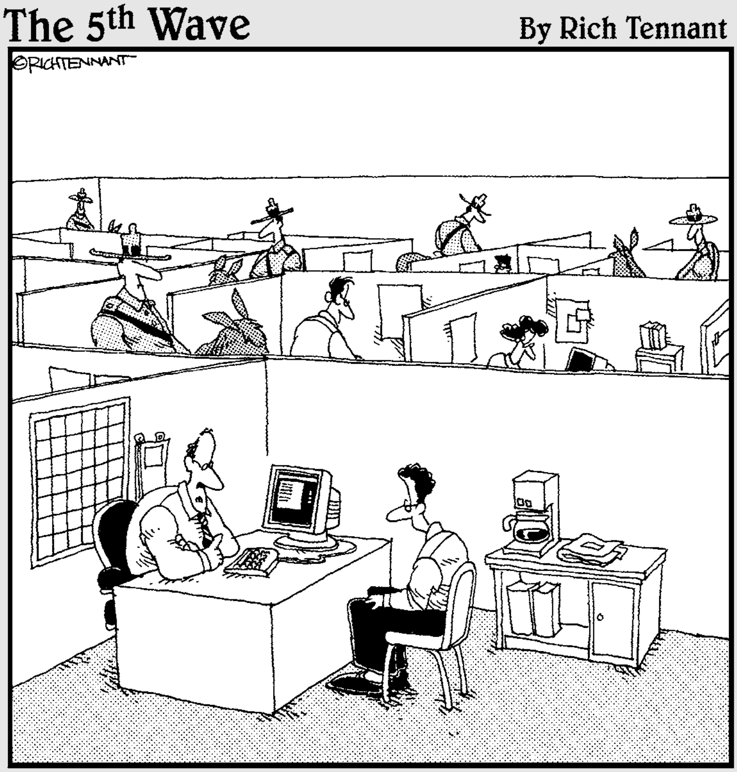
Best Practices

As you design your AD implementation plan, keep the following practices in mind:

- **Ensure that everyone involved with the AD implementation project has a clear idea of what business and technical goals need to be met through Active Directory's deployment.** Pay particularly close attention to company management and make sure that they understand the purpose and objectives of the project and that they support this implementation. You can achieve this through well-written documentation, making that documentation readily available, and through regular communications to all groups involved.
- **Build an AD planning team to develop the design. Don't try to do it on your own.** Active Directory is a complicated IT service to design, especially in large environments. The design benefits when a diverse team of individuals that have different perspectives within the company creates it. As the old saying goes, "Two heads are better than one!"
- **Don't rush to create the functional specification without doing the requirements/scope and gap analysis documentation first.** Your company's needs (as identified in these documents), instead of what the planning team thinks, must drive the AD design.
- **Develop a risk assessment that provides a contingency plan in case problems arise during the implementation.** By having this assessment developed and contingencies for these risks planned ahead of time, you greatly increase your chances of remaining on schedule.
- **Develop and publish the AD implementation project plan.** Use this plan to track all the tasks and resources you need to complete the implementation. Moreover, by keeping the project plan up-to-date, you can easily identify and correct delays in the rollout.

Part II

Planning and Deploying with Active Directory Domain Services



"We take network security here very seriously."

In this part . . .

Now you're ready to look at the process of designing and then deploying your Active Directory Domain Services environment. You really have to examine two sides of this: the logical design and the physical design. In Chapter 4, we look at the Domain Name Service and logical namespace design that needs to be considered before you can get to the forest/domain/OU design that we cover in Chapter 5. Chapter 6 looks at the physical side of AD DS design as well as examines one of the new features in AD DS in Windows Server 2008, the read-only domain controller. Chapter 7 walks you through how to deploy domain controllers that enable your AD DS design.

Chapter 4

Playing the Name Game

In This Chapter

- Reviewing DNS basics
- Understanding the Active Directory requirements for DNS
- Exploring the Active Directory namespace
- Determining the types of Active Directory naming
- Designing an Active Directory namespace
- Discovering what's new in Windows Server 2008 DNS

Beginning with Active Directory in Windows Server 2000, Domain Name Service (DNS) has become a vitally important subject that must be understood by anyone wanting to design or support Active Directory. This certainly hasn't changed in Windows Server 2008!

DNS is a huge topic, with numerous books devoted to its concepts and implementation. However, I don't go into all the details here. In this chapter, I cover the DNS basics that directly relate to implementing and managing Active Directory.

The Need for DNS

Humans and computers are very different animals. Humans typically find it easier to remember words than numbers. For example, most people can quote the Pledge of Allegiance from memory. But, how many of you can remember your locker combination from high school? Comparatively, computers, at the most basic level, are designed to work with numbers — not words.

When a user on a TCP/IP-based network wants to communicate with a host computer, the user has to refer to that computer in some manner. Most users naturally refer to the textual name for the computer. Unfortunately, at a base level, computers can't find other computers on the network by using names like *Spock* or *Jimmy's Cool PC*. Computer-to-computer network communication on a TCP/IP-based network occurs by using IP addresses. But given the limitations of human brains, trying to remember the IP addresses of all the hosts that you want to communicate with is impractical (unless you happen to be an IP savant). Therefore, some mechanism is required to convert the host computer's name to an IP address. This conversion process is *name resolution*. DNS is the application in TCP/IP-based networks that provides name resolution services.

Using the Internet without accessing DNS is nearly impossible. Every time you type a Web address (for example, `www.microsoft.com`) or click a hyperlink on a Web page, you're using DNS to resolve a Web host name to an IP address. The delivery of e-mail across the Internet also depends on DNS functioning properly.

DNS can also act as a locator service on networks. A *locator service* enables a computer to find a particular network-provided service (such as an Active Directory domain controller) without having to know the name of the computer providing the service. Microsoft recognized the need for an enterprise-class locator service to use with Active Directory and selected DNS. Clients and servers use DNS as a locator service to find the various AD-related services on a network.

Essential DNS

DNS is a name-resolution service. A network client searching for a host uses DNS to resolve the host's name to its IP address. In simple (really simple!) terms, the process goes something as follows:

1. A network client transmits a message to a DNS server, asking for the IP address that matches a given host name.
2. The DNS server searches its DNS database for the host name and locates the IP address that corresponds to the host name.
3. The DNS server returns a message containing the IP address to the network client.
4. The network client then directs a message to the desired host by using the appropriate IP address.

In its simplest form (are you seeing a simple pattern emerging?), an entry in the DNS database table looks something like the following example:

```
L01.corp.com   IN   A   10.50.4.41
```

This entry is a *resource record*. The preceding example matches the IP address `10.50.4.41` to a server with the name `L01` in the `corp.com` domain. A *DNS table* consists of a listing of resource records similar to this example.

Identifying resource records

DNS resource records define more than just names and IP addresses. Various types of resource records identify servers, domains, zones, and services. For example, a resource record identifying a *canonical name* (an alias or nickname) for a server would look like the following:

```
exchange   IN   CNAME   10.50.4.48
```

Table 4-1 lists the most common types of resource records.

Table 4-1 Common Types of Resource Records

Type	*Purpose*
A	Address resource records match an IP address to a host name.
CNAME	Canonical name resource records associate a nickname to a host name.
MX	*Mail exchange* resource records identify mail servers for the specified domain.
NS	*Name server* resource records identify servers (other than the SOA server) that contain zone information files.
PTR	*Pointer* resource records match a host name to a given IP address. This is the opposite of an Address record, which matches an IP address to the supplied host name.
SOA	*Start of authority* resource records specify which server contains the zone file for a domain.
SRV	*Service* resource records identify servers that provide special services to the domain.

Active Directory Requirements for DNS

Active Directory depends on DNS to act both as a name resolution and locator service. For this reason, you need a preexisting DNS infrastructure before you can implement Active Directory. DNS must meet only one requirement for it to able to support Active Directory — DNS must support SRV records. The following sections discuss how Active Directory depends on DNS.

Examining SRV records

DNS is great for giving you an IP address when you know a server name, but how does DNS act as a locator service for Active Directory? DNS uses SRV records to map AD services, such as a domain controllers and global catalogs, to specific server names.

Every domain controller at boot time registers its host name with an A record in DNS. The domain controller also registers multiple SRV records in DNS that identify the domain controller as an LDAP, a Kerberos, and potentially a global catalog server. When the client must locate a domain controller, the client queries DNS for a domain controller service name. DNS locates the appropriate SRV record in its database and returns the host name of the domain controller. The client then queries DNS for the A record of the host name to obtain the domain controller's IP address. At this point, the client can communicate with the domain controller through the IP address. Without this capability to locate AD services, clients would be unable to authenticate into Active Directory or search the global catalog.

Understanding RFCs

I frequently sprinkle the abbreviation RFC throughout this book. A *Request for Comments,* or *RFC,* is a document that proposes a new Internet specification or protocol. The Internet Engineering Task Force (IETF) publishes these documents. The IETF is an organization of international participants with an interest in the operation of the Internet. Additional information about the IETF is available at the following Web site:

```
www.ietf.org
```

An author submits RFCs to the IETF as *Internet Drafts*. After approval by the IETF, an Internet Draft becomes an RFC, and the IETF publishes it. By assuring that products adhere to the standards of a particular RFC, vendors ensure interoperability between products.

Active Directory introduces a great deal of new technology, so referencing the RFCs on which the technologies are based is a common practice. Following are some important RFCs that relate to Active Directory:

- RFC 1777, LDAP version 2
- RFC 1779, LDAP naming conventions
- RFC 1823, LDAP API
- RFC 2052, SRV records
- RFC 2136, Dynamic DNS
- RFC 2247, LDAP naming conventions
- RFC 2251, LDAP version 3

You can find these RFCs on the IETF Web site that I list in this sidebar.

Don't worry too much about remembering the RFC number associated with a particular technology. (I usually mark those by using the Technical Stuff icon, which means, "Here are the specifics, in case you want to know.") The only two that you might hear mentioned frequently in relation to Active Directory are RFC 2052 and RFC 2136.

Exploring dynamic updates

Because you're using DNS as a locator service in Active Directory, the A and SRV records must be stored in the DNS server. In the past, the only way that records could be added to a DNS server was to have the DNS administrator edit a text file that contained the DNS records. For a large network, having the DNS administrator edit this file for every device on the network was extremely laborious. Fortunately, there's a better solution. An enhancement to DNS allows computers to automatically register records into the DNS store themselves. With this enhancement referred to as dynamic DNS (defined in RFC 2136), a DNS client can submit a registration request to the DNS server and the DNS server software will perform the registration itself without needing the DNS administrator to lift a finger.

Although it's not an absolute requirement, I strongly recommend that DNS servers supporting Active Directory support dynamic updates. You might be able to get away with not running dynamic DNS (DDNS) in a small network, but maintaining a large environment without dynamic DNS is almost impossible.

REMEMBER

You should understand that you're not forced to use Microsoft's implementation of DNS. Other vendors have DNS software for sale that supports Active Directory including DDNS. However, be careful in using someone else's DNS product because you can't be sure that it completely supports Active Directory without some testing. Of course, you probably don't have to be concerned about this if you use Microsoft's DNS product. So that's one less thing to worry about!

Storing and replicating DNS information

So how is all this information in the form of resource records stored on a DNS server? The information is placed in one or more *DNS zones*. A DNS zone is a contiguous portion of the DNS domain name tree that a DNS server specifically hosts. Although you can host multiple domains within a zone, you can't split a domain across multiple zones. To illustrate, looking at Figure 4-1, you can see a portion of the DNS domain name tree. The dashed circles represent two contiguous portions of the tree that you could create zone files for. The first zone contains the `steveco.net`, `eng.steveco.net`, and the `hr.steveco.net` domains. Because all these domains share a common root domain (`steveco.net`), they can be placed into the same zone file. Similarly you could have a separate zone for a single domain — in this case, `acct.steveco.net`. What you can't do, though, is create a single zone file that would, for example, contain `eng.steveco.net` and `microsoft.com` because those two domains would not share a common root domain within the zone.

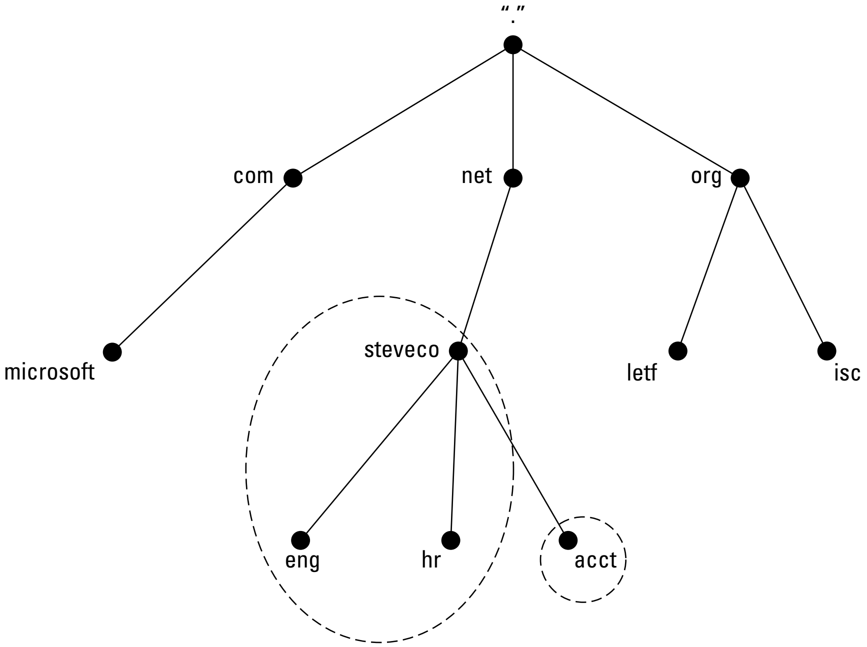


Figure 4-1: DNS zones.

You refer to a DNS server as being *authoritative* for a zone when that server stores a copy of the zone. In other words, that DNS server is responsible for providing name resolution for that portion of the domain name tree.

Microsoft's DNS server supports three types of zones, as follows:

- **Primary zones:** A primary zone is the master copy of a zone that is writable. In other words, when changes to the zone file need to be made, they're made on the DNS server that is holding the primary zone file.
- **Secondary zones:** A secondary zone is simply a read-only copy of the primary zone that is created to distribute the zone file to other DNS servers for performance and redundancy reasons. DNS has a built-in mechanism for replicating zone data from a DNS server holding a primary copy of a zone to a DNS server that holds a secondary zone copy.
- **Stub zones:** A stub zone is a little bit harder to explain. Imagine a situation where you work with another company and you need to perform frequent DNS queries to that company's namespace. You would be tempted to create a secondary zone copy of that company's namespace to speed up DNS queries. Two problems arise when doing this though. First is the amount of network traffic that is generated by the replication of the zone from the primary zone holder DNS to your DNS. If their zone is large, and

> changes to the zone are frequent, you might have a lot of traffic to deal with. Second, if the other company's IT staff changes the IP address of the primary DNS zone holder, and they forget to tell you, your DNS server becomes an orphaned zone holder with no new updates being sent to you. A stub zone addresses this issue by being a secondary copy of the primary zone, but it only contains the zone's SOA record and then the NS and A records of only the DNS servers holding this zone data. Therefore, another way of looking at stub zones is they're a pointer to another DNS server that holds a copy of the zone data that is being requested.

Zones can be stored in one of two ways on a Windows DNS server. Either a zone is stored as a text file on the server's hard drive or, if the DNS server is an AD domain controller, the zone information can be stored and accessed from the AD directory store. Microsoft's DNS server supports text file primary, secondary, and stub zones, but for these AD stored zones (*AD-integrated* zones), only primary and stub zones can be stored. Why is this? Consider how AD domain controllers work. In Chapter 1, I state that Active Directory follows a *multimaster* replication model, meaning that you can write changes to an AD domain at any domain controller. Supporting secondary zones that are read-only copies on domain controllers that have writable directory data wouldn't make sense.

So, should you store the zone data in a text file or in Active Directory? Storing the data in Active Directory is more compelling than in a text file for a couple of reasons. First, if you store the zone in Active Directory, the normal AD replication process takes care of replicating the zone information to the other domain controllers. Second is the security that Active Directory provides. Each DNS resource record stored in Active Directory is an AD object with an Access Control List (ACL) on it specifying who has what permissions to that object. By having an ACL on an A record, for example, you can control what users or computers can make changes to that record. This prevents two computers from trying to register the same name in DNS, which is important when you're supporting dynamic DNS.

When storing zone data in Active Directory, you have a number of options available to you to control which domain controllers and DNS server get a copy of the data. This helps to reduce the AD replication traffic so that only the domain controllers running DNS that actually need the zone data receive it. You have the option of replicating a zone to all DNS servers in the Active Directory forest, all DNS servers in the domain that is contained in the zone, all domain controllers in the domain that is contained in the zone (regardless of whether you're going to install DNS on it), or you can store the information in an *application partition.* Application partitions are useful when one of the other replication options just doesn't fit the replication model you want to use. In Chapter 7, I show you how to create an application partition and how to control which DNS servers get a copy of it.

Microsoft's DNS software features the support of incremental zone transfers. Without the support of incremental zone transfers, DNS servers have to transmit the entire zone file to the secondary zone holders when a single change is made. Incremental zone transfers allow only the changes to be sent, greatly cutting down on the amount of network traffic used for zone replication.

The Active Directory Namespace

Active Directory contains a directory listing of objects on your network for users to use and administrators to administrate. The AD structure should be designed to make the directory as easy to use as possible by employing descriptors to name the directory's objects. Users constantly search the directory for objects, and these searches are simple when the objects have meaningful names. For example, if a user needs to find a printer on the network, descriptors (such as the location of the printer and type of printer) simplify the search. Some of the popular AD objects include the following:

- Domains
- Sites
- Organizational Units
- Domain Controllers
- Computers
- Printers
- Groups
- Users

All these objects exist within the Active Directory namespace. The following sections explain the concept of a directory namespace.

Defining the Active Directory namespace

A *namespace* is a bounded region in which you can resolve a name to an object. When a user queries Active Directory for an object (see Figure 4-1), the user must specify one or more attributes of that object that fit within the Active Directory namespace. Every object in AD has multiple attributes that you can use to refer to objects in Active Directory.

The Internet is an example of a DNS-enabled namespace (a really big one!). The root domain is the dot (.) domain and the namespace contains all the Internet devices (Web server, mail servers, FTP servers, and so on). You can make queries to the objects (such as `www.microsoft.com`) within the Internet namespace to obtain information.

Comparing an Active Directory namespace to a DNS namespace

It's important for you to understand how Active Directory and DNS relate. Both DNS and Active Directory use the term *domain,* but these are two separate, although related, entities. It's easy to be confused by the two different uses of this term. A *DNS domain* is a collection of resource records that describes the hosts and services within that domain. An *Active Directory domain* refers to how objects in Active Directory are partitioned. Each AD domain requires a corresponding DNS domain because you use DNS as the locator service for Active Directory. You must register the Active Directory services and computers of a particular AD domain to the corresponding DNS domain so that your AD clients can locate these services and computers by using DNS queries. An example of this mapping is shown in Figure 4-2. The right side of the figure depicts a DNS domain name tree that shows the structure for the `steveco.net` domain. Each one of these domains is associated to an AD domain that is shown on the left side of the figure. All the DNS records for a particular AD domain are placed into the corresponding DNS domain.

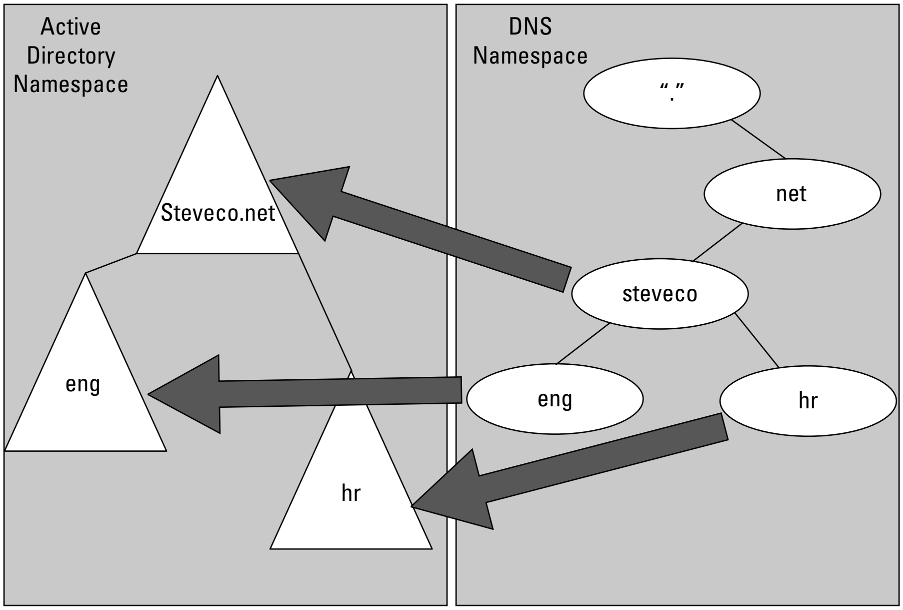


Figure 4-2: Mapping the DNS namespace to Active Directory.

Types of Active Directory Naming

Active Directory has a variety of standards and protocols that follow certain naming conventions. Before I introduce you to implementing Active Directory, you need to understand the terms — or names — used to describe various directory objects. They're used in many of the installation wizards and help files, and they can be quite confusing if you don't have them straight.

Fully qualified domain name

A *fully qualified domain name* (FQDN) is the entire path leading to a network object. For example, `servera` (Server A) is located in the `west` domain in a tree named `corp.com`. Server A's FQDN is

```
servera.west.corp.com
```

Similarly, a printer located in the same tree and domain might look like this:

```
prt1.west.corp.com
```

Using the FQDN, you can always identify the exact location of an object in the DNS namespace. Looking at the Server A example above, you know that `servera` is a member of the `west.corp.com` domain because of its FQDN. However, an FQDN doesn't necessarily include all the information that shows where the object is located in Active Directory.

Distinguished name

Every object in an AD forest has a distinguished name. A *distinguished name* (DN) is an X.500-based naming convention and is how objects are found in the directory by using LDAP. Distinguished names use some very odd abbreviations including:

DC	domain component
OU	organizational unit
CN	common name

These abbreviations are combined in a specific order, from left to right, to describe the exact path leading to an object. The common name of the specific object is listed first, followed by organizational units (if they exist), and then the domain component names.

If you apply distinguished names to the examples I use for FQDNs in the preceding section, here are the results:

```
CN=ServerA,DC=west,DC=corp,DC=com
```

```
CN=Prt1,OU=Printers,DC=west,DC=corp,DC=com
```

TIP

You might note that the Prt1 distinguished name above doesn't completely match the Prt1 FQDN. That's because although Prt1 is in the west.corp.com domain, it happens to be in the Printers organizational unit with the west.corp.com AD domain. So, although the FQDN of an object matches to where the object is located in the DNS namespace, the DN completely describes where the object is within the Active Directory namespace.

User principal name

Every user object in Active Directory has a user principal name. A *user principal name* (UPN) is the name usually recognized as an e-mail address. A UPN consists of the user's logon name and the domain name where the user object is located.

For example, user JoeB located in the domain xyz.com has the user principal name

```
JoeB@xyz.com
```

Similarly, user John Doe (logon name JohnD) located in domain corp.com has the user principal name

```
JohnD@corp.com
```

UPNs are useful particularly in multiple domain forest environments. Typically, when a user logs on, she has to specify her user ID and the domain where her user ID (that is, user object) is located as well as her password. If she knows her UPN, she can log on by typing the UPN without having to supply the domain name separately. One less piece of information your users need at login time!

NetBIOS name

To ensure backwards compatibility with Windows NT and older applications, some AD objects (primarily domains, computers, users, and printers) have NetBIOS names associated with them. The NetBIOS names enable older clients and applications to refer to these objects. Although you can use any valid NetBIOS character in these names, I strongly suggest that you use the same NetBIOS name for an object that you use in Active Directory. This practice eliminates confusion.

Planning the Active Directory Namespace

Before you can begin deploying Active Directory, it's essential that you invest some time in carefully planning the namespace that your Active Directory forest, or forests, will occupy. This is important because you're using DNS for your naming system in Active Directory. In the following sections, I cover the decisions that you must make and the implications of those decisions on the rest of your AD design.

Understanding domain naming

You can't choose your AD domain names haphazardly. The domain names you choose have far-reaching implications for the scope of your AD implementation within a company and for how that forest interacts with other networks, particularly the Internet.

When naming any of your domains, you should observe the standard DNS naming restrictions. This means following the characters defined in RFC 1123 that are supported by DNS. The RFC 1123 character set includes these characters:

- A–Z
- a–z
- 0–9
- - (hyphen)

You should follow these design principles when selecting the names of your AD domains:

- **Choose a name that's not likely to change.** Because renaming a domain is a complicated and risky process, don't choose a name that might change frequently. If you're considering naming domains after functional areas within your company, the company's previous reorganizations should give you a good feel for what names are static and not static in the company. Typically, choosing the company name as the forest root is a good idea. However, with corporate mergers these days, even that choice can be risky. One popular choice in forest root naming is to use a private name that is generic so you never have to worry about changing the name. An example of this would be to name your forest root domain `root.local`. With this kind of name, you're unlikely ever to need to worry about changing the name even if your company renames itself.

  WARNING!

  Originally, Microsoft didn't provide a way to rename a domain within an Active Directory forest. Starting with Windows Server 2003, Microsoft has provided a domain-renaming tool (RENDOM). However, the use of this tool is not something to take lightly as renaming a domain is essentially restructuring your AD forest, which is not a simple task. So although you have the ability to rename a domain, try to avoid the need to do this by planning your domain names ahead of time.

- **Register the name of your domain with an authorized domain registar.** If you deploy Active Directory, you have the option of deploying some or all the servers directly on the Internet. This can be required if you implement Active Directory Federation Services for example (see Chapter 9). Because the Internet and Active Directory both use DNS for name resolutions, you must be sure that the name of your root domain doesn't conflict with an existing registered name. Even if you company doesn't have an Internet presence (which is hard to imagine), you should still plan Active Directory as if you do have to interoperate with the Internet by registering a domain name. There a large number of authorized domain registrars. You can find a list of these companies at:

  ```
  www.icann.org/registrars/accredited-list.html
  ```

- **Avoid using a domain name more than once within your infrastructure.** If you have the same domain namespace in two different networks (for example, the Internet and your intranet), it might be difficult to determine whether the device you're referring to in this namespace is on the Internet or is internal. Managing this (like doing a split DNS design) requires a lot of planning and, if you are not careful, could cause a lot of confusion and ambiguity for your users and your administrators.

Understanding OU naming

You use organizational units to group objects into separate containers — especially those objects you want to administrate as a set. Although your users can view this OU structure, you're primarily setting it up for the administrators. You're limited by few constraints in naming your OUs. Just follow a consistent strategy and name the OU with a name that's representative of why you created the OU in the first place.

Understanding computer naming

A computer (whether a domain controller, member server, or desktop) is an object in Active Directory. Therefore, it has a name associated with it. In fact, a computer has two types of names associated with it: a DNS name and (for backwards compatibility) a NetBIOS name.

The DNS name, typically, is the computer's host name appended to the front of the computer's AD domain name. An example of a DNS name is `computer1.steveco.net`. The NetBIOS name normally is just the computer's host name, such as `computer1`.

Observe the following guidelines when naming your computers:

- Keep your computer's host name in Active Directory the same as its NetBIOS name to eliminate confusion.
- Limit your computer's host name to 15 characters because NetBIOS names are limited to 15 characters.
- Develop a naming standard for computer host names to guarantee uniqueness.
- Create separate naming formats for clients and servers. The administrator and user can then easily recognize whether a name refers to a client or server.
- Choose a server name that's represents the network services it provides. Your network is easier to use when a user or administrator can determine from the server's name the type of service the server provides. However, this naming strategy requires that you do some planning to determine the types of services your server will provide.
- Use the RFC 1123 character set because these computer host names are registered in DNS.

Understanding user naming

I discuss in earlier sections in this chapter that users have a UPN name and a NetBIOS user logon name as attributes of the user object. You should set up a naming standard for your user IDs that defines the format of the UPN and NetBIOS name. What you're primarily concerned with is the part of the ID that is used for the UPN prefix (the part of the UPN name on the left side of the @ symbol) and the NetBIOS name of the ID. Users can use either of these attributes to refer to their ID at the Netlogon window when logging on to Active Directory.

Follow these guidelines to develop your user-naming standard:

- **Keep the UPN prefix and the NetBIOS name the same.** When you create a new user, you have the opportunity to use different names for these attributes. Resist the temptation to do so. By using different names, you introduce uncertainty and confusion on the user's part because they could use either item to log on to the network.

- **Use a format that's both meaningful and user-friendly.** Because this username is the primary way of referring to a user ID, you want to choose a format that refers to the user. The user, who can then easily remember the ID, and the administrator, who can determine the user from looking at the ID, both benefit.
- **Use a unique name with the Active Directory forest.** An individual AD domain requires that user IDs have unique names. However, at a forest level, you can have users with the same UPN prefix and/or NetBIOS name.

If you don't use a unique name, you might encounter a problem when you have to move a user to another domain. Without unique IDs, you increase the probability of a naming conflict between the IDs you're moving and an existing ID in the target domain. Using unique UPN prefixes and NetBIOS names within the forest guarantees that user ID migrations don't force you to rename the UPN prefix or the NetBIOS name.

Make a user's e-mail address his UPN. If your users already have an RFC 822-formatted Internet mail address — that is, `<user>@<domain>` — you eliminate confusion by using the same name for the UPN. Using the same name simplifies life for the users because it's one less thing to remember.

REMEMBER

A UPN must be unique in the forest. A NetBIOS user logon name doesn't have to be unique.

What's New in Windows Server 2008 DNS?

How has Microsoft improved their DNS software in Windows Server 2008? Mostly, the DNS software has remained unchanged, but a few minor differences exist that you should be aware of.

Support for IPv6

IPv6 is the latest version of the Internet Protocol used for the network routing and address assignment technology on the Internet and most likely in your internal network. IPv6 provides support for a different IP addressing scheme that allows for a much larger address space than what's available in IPv4. Because DNS supports the A records that map DNS host names to an IP address, changes have been made to Microsoft's DNS product that support the IPv6 address scheme.

Support for read-only domain controllers

I say in the earlier "Storing and replicating DNS information" section that if a zone is AD-integrated that is a writable copy of the zone. Well, that's not completely true. I cover read-only domain controllers in Chapter 6, but for now, you need to understand that if you have an AD-integrated zone on a read-only domain controller (RODC), you cannot make changes to that zone on that DC. It's read-only. Therefore, the DNS software had to be updated to understand how to handle this type of read-only AD-integrated zone. Now, when an RODC running DNS receives a dynamic registration request, that request is forwarded to a DNS server that can accept the request (that is, it has a writable copy of the zone). After the registration completes, an immediate replication of the zone is executed back to the RODC that originally received the dynamic registration request.

Background loading of zone data

One problem in earlier versions of the DNS software was that when an AD-integrated zone was very large and therefore took a long time to load, the DNS server wouldn't accept queries for that zone until the load finished. This created additional problems when domain controller reboots were done as the length of service interrupt would be extended. Microsoft updated the DNS software so that the zone loads are now done as a separate thread in the background so that DNS can respond to queries sooner instead of waiting for a zone load to complete.

GlobalNames zone

Before Windows 2000 and Active Directory, Microsoft's directory service (NTDS) utilized the Windows Internet Naming Server (WINS) as a name resolution service. WINS allowed you to create a flat namespace for the resolution of NetBIOS names only. Originally, when Active Directory was released in 2000, it was envisioned that organizations would be able to phase out the use of the flat namespace that WINS provided in favor of DNS's hierarchical namespace. That hasn't proven to be the case though. Cases exist still where having a flat namespace can be useful; especially when you don't know what the AD forest structure is. To help organizations finally get rid of WINS, Microsoft is providing support for a new zone called GlobalNames. In this zone, you can register every device in your forest so that you end up creating a separate flat namespace that can be queried without the need to know which domain a device is in.

Chapter 5

Creating a Logical Structure

In This Chapter

- Creating trees and forests
- Defining domain boundaries
- Determining your structural model
- Designing an OU hierarchy
- Delegating administrative privileges

Before you begin implementing an Active Directory structure on your network, take the time to draw it on paper first. Stare at it, rearrange it, poke holes in it, and generally abuse it. That's right! Cause all the damage you can while the design is on paper.

In this chapter, I show you how to design and organize an Active Directory tree or forest. The decisions that you make in the design phase of your AD structure have a huge impact on your work life. These decisions affect the complexity and cost of managing your network for a long time. Make sure that you create a tree or forest that makes your job easier — not harder!

Planting a Tree or a Forest?

Your first decision in designing an AD structure is deciding whether you need one tree or multiple trees. The easiest way to determine how many trees to design is to consult your DNS namespace. All objects in an Active Directory tree must share the namespace, as I discus in Chapter 4.

Within Active Directory, each object has an X.500-based distinguished name. (See Chapter 4 for more information on distinguished names.) This distinguished name creates a path from the object to the root domain at the top of the tree. If all the objects in your planned tree can extend from the same root domain name, you can create a single tree. Figure 5-1 shows a corporation with a single namespace.

To put it simply, you create a forest only if you need to use more than one namespace. If you require more than one namespace because you require more than one naming structure, you need to plan an additional tree for each namespace. Figure 5-2 shows a corporate structure with two parallel divisions. Each division has different name requirements, which means that you must plan a separate tree for each division.

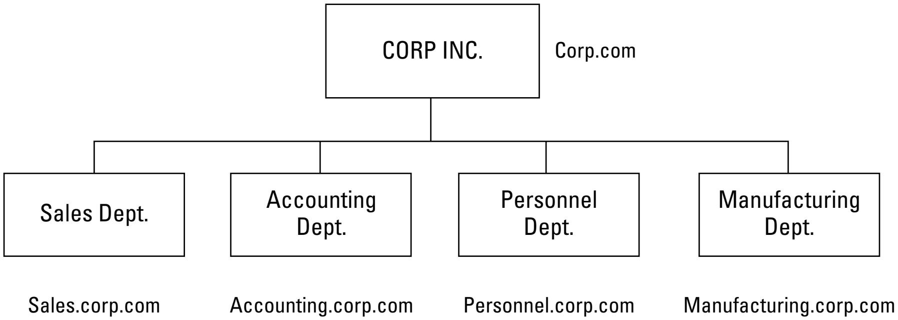


Figure 5-1: A name space contained within a single tree.

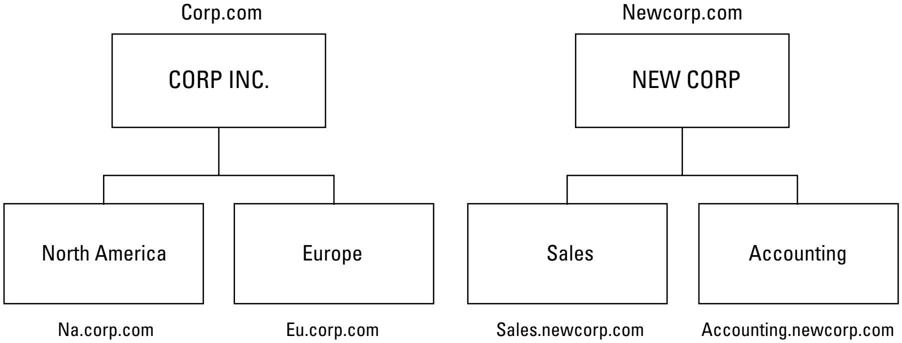


Figure 5-2: Two namespaces require two separate trees in the same forest.

Think of the differences between a tree and a forest in the following way:

- A *tree* is a logical grouping of domains within the same namespace.
- A *forest* is a logical grouping of one or more trees that a transitive trust relationship joins. Each tree in a forest has a distinct namespace.

Domains within a forest share the following common characteristics:

- **The same schema:** Every domain controller (DC) in the same forest contains a copy of the same Active Directory schema. A schema change affects all domains in the same forest. You cannot share a schema between forests.
- **The same configuration partition:** Like the schema, every DC in the same forest shares the same copy of the configuration partition.
- **Automatic trust of each other:** All domains in the forest are connected with two-way transitive Kerberos trusts. These trusts are set up automatically

and require no additional configuration. Therefore, the administrator can easily set up security groups containing users in different domains in the same forest.

- **A common Enterprise Administrators group:** The first domain in the forest is referred to as the forest *root domain*. Within the forest root domain, a security group called Enterprise Admins is created as a part of installing the first DC in the forest root domain. In Active Directory, you need to remember the following points regarding the Enterprise Admins group:
    - By default, the Enterprise Admins group is the only group allowed to make forestwide changes, such as the addition or removal of other domains to the forest.
    - When other domains are added to the forest, the Enterprise Admins group in the forest root domain is granted full control over that domain.
    - The Enterprise Admins group is added to the Administrators group on all domain controllers in the forest.
- **The same global catalog (GC):** Each DC that is designated as a global catalog server in the forest shares the identical copy of the GC.

Defining Domains: If One Isn't Enough

A domain is the cornerstone that you lay whenever you create trees and forests. Regardless of whether you design a tree or a forest, the starting point is always the *root domain*. The root domain is the first domain that you create in your AD structure, and it sits at the top of your diagram.

The root domain of your tree, similar to any other domain, is a grouping of resources built on the following components:

- Domain controllers
- Security policies (see Chapter 14 for more information)

For many small and medium-sized companies, a single root domain with a structured OU (organizational unit) model, shown in Figure 5-3, provides sufficient flexibility for an AD tree. If this is your situation, congratulations! You can move right along to the upcoming "Organizing with OUs: Containers for Your Trees" section. Life is good.

REMEMBER

One new thing in Windows Server 2008 AD DS is the ability to create different password policies within the same domain. Although in the past it was impossible to have multiple password policies for users in the same domain, in 2008, you can do this. However, just because you can do something doesn't necessarily mean that you should. I discuss more about this in Chapter 14.

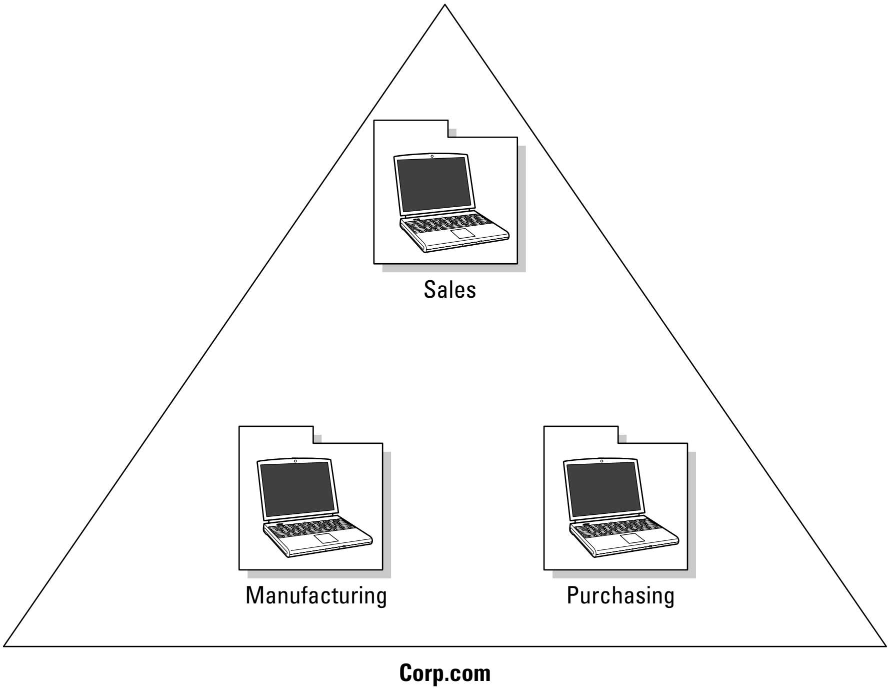


Figure 5-3: A single root domain with a structured OU model.

However, larger companies, companies with complex organization charts, and companies with multiple sites often find that a single domain isn't suitable. If you suspect that your organization falls into one of these three situations, read on.

Less is more!

A tree can consist of a single domain, and this configuration is highly desirable! Whenever possible, you want to limit your design to a single domain that you can organize and administer through OUs.

However, reality seldom follows best practice. (There's always a catch, isn't there?) More times than not, a single domain just isn't possible. After you specify a root domain, consider the following justifications for creating additional (child) domains:

- Your organization uses slow WAN-link connections, and you need to limit replication traffic across those links.
- Your organization has varying security needs that you can't accommodate within a single domain.
- Your organization has distinct political or organizational factions that require separate *administrative boundaries;* that is, different groups of administrators control specific domains.

- Your organization is very large, and additional domains provide for future growth and ease administrative burdens.
- Your organization spans international boundaries, and multiple domains enable you to separate corporate resources according to those boundaries.

TIP

Unfortunately, I have found that the most common reason for a corporation to require additional domains is to accommodate politically polarized groups who refuse to share the same sandbox. That statement might not sound quite so harsh if you consider the substantial expense that accommodating such turf wars involves.

Microsoft recommends that you keep your AD tree shallow. Remember that, within domains, you also have an OU hierarchy. (For more information about the OU hierarchy, see the upcoming "Organizing with OUs: Containers for Your Trees" section.) Between the tree hierarchy and the OU hierarchy, you might introduce significant complexity if you design too many levels. Each added domain or OU level decreases performance on your network. After you progress beyond five levels in domains or OUs, you begin to see a significant decrease in system performance.

REMEMBER

A good rule is to limit the depth of domains in a tree to a maximum of three. Two is preferable. One is optimal.

Recognizing the divine order of things

The key to designing an efficient Active Directory is to base the structure of the tree on the structure of your company. Start with a copy of your current corporate organization chart. Identify the root domain by determining the highest-level organizational group on the chart. The name of the root domain must always match the first level of the namespace. If you determine that you have separate namespaces and need more than one tree, work through designing one tree at a time.

WARNING!

Carefully consider the name of your forest root domain because changing its name later is a difficult process. Many times, companies use a generic name like `AD.LOCAL` for the forest root name so that the name doesn't have to be changed because of company name changes or other DNS namespace changes.

Use a triangle to represent a domain as you draw an Active Directory diagram. In your drawing, leave room under the root domain for the lines that represent the trust relationships. Figure 5-4 shows an AD diagram with lines representing the automatic trust relationships between the domains in the tree.

REMEMBER

In Active Directory, trusts (except for external trusts) are bidirectional and transitive. The trust relationship is passed along to all other domains connected by transitive trusts. A domain added beneath the `sales` or `manufacturing` domain in Figure 5-4, for example, automatically trusts the `corp` domain, which is the root domain.

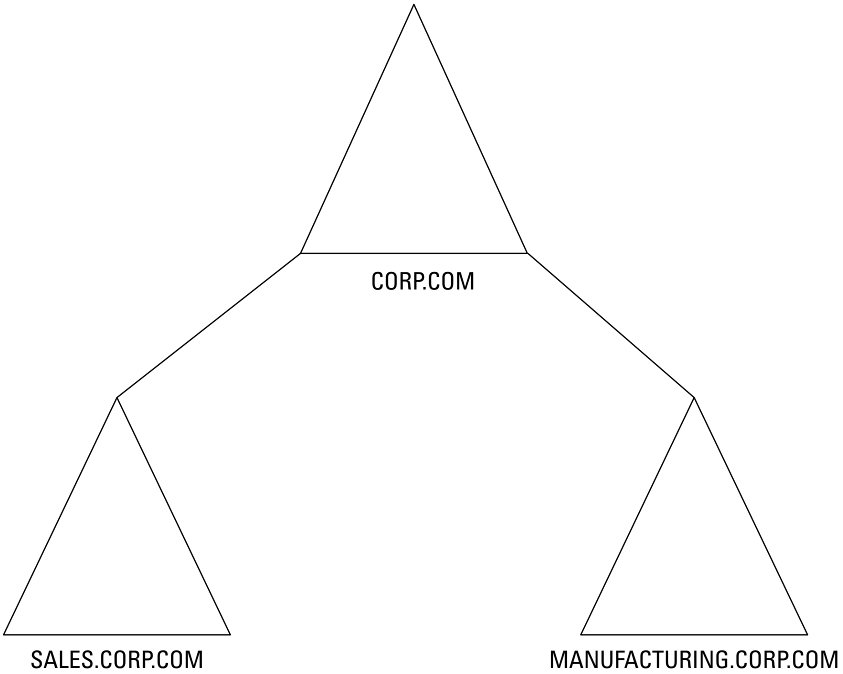


Figure 5-4: An AD diagram showing domains and trust relationships between domains.

If your company doesn't have an organization chart on which you can base your design, you must create your own structure. Most AD structures follow either a geographic or a functional model, as the following sections explain.

Geographic modeling

One popular method for designing an AD structure is to use a *geographic model*. If your organization is structured along international boundaries, this model is the one to use.

Look at the organization chart in Figure 5-5. This corporation structures itself along international boundaries. Administrative functions within one location function separately from those in other locations. You can efficiently manage this company by using a geographically modeled structure. The corresponding domain structure is shown in Figure 5-6.

Geographically bounded domains don't necessarily dictate a decentralized administrative model. Although decentralized administration is common in a geographic model, a centralized — even remote — IT department can still manage multiple domains.

Functional modeling

A *functional model* adapts to a variety of organization charts. Use it to group domains according to the following business models:

- Department
- Division
- Project

If you believe that a functional model is best for your organization, choose one of these three categories and define the child domains accordingly. Creating additional domains that reflect departments and divisions is quite common. Creating a domain based on a project, however, is less common because projects are seldom permanent. Domains should be stable. If you try to implement a project-based domain structure at a company where projects change frequently, you're creating an administrative nightmare.

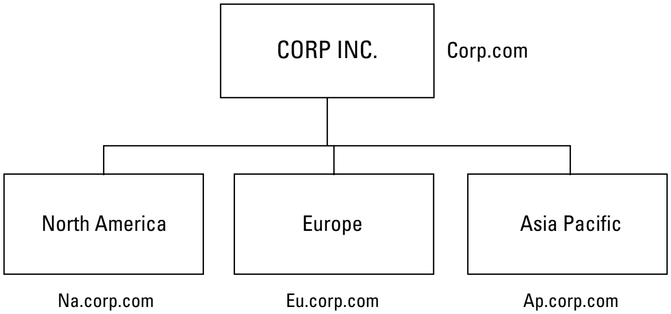


Figure 5-5: An organization chart displaying a geographic structure.

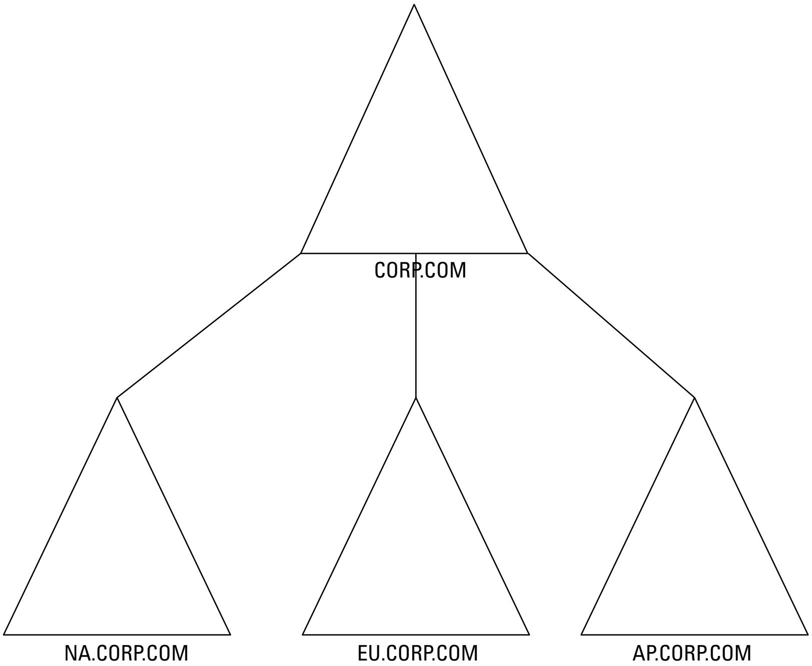


Figure 5-6: A domain tree displaying a geographic model.

Domain renaming

Starting with Windows Server 2003, Microsoft has a tool (RENDOM.EXE) that allows you to rename both the DNS and NetBIOS name of a domain. Because the DNS name of an AD domain also represents its position in the AD tree, renaming a domain can actually move the domain to a different location in a tree. RENDOM can be used to do the following:

- **Rename a domain without moving it:** Only the name of the domain changes and not its position in the AD tree (such as renaming `DomainA.Corp.Com` to `DomainB.Corp.Com`).
- **Rename a domain and move it within the same tree:** Not only does the domain name change, but also its position in the DNS namespace changes (such as renaming `DomainA.Finance.Corp.Com` to `DomainB.Corp.Com`).
- **Creating a new tree:** In this case, you might be creating a completely new namespace within the AD forest (such as renaming `DomainA.Corp.Com` to `DomainA.NewCorp.Com`).

The use of RENDOM is complicated, and you must carefully consider using it because incorrect usage can be extremely damaging to your AD forest. ***Note:*** Although you can use RENDOM to rename the forest root domain, you cannot designate another domain to become the forest root domain.

WARNING!

Don’t get in the habit of creating domains for resources that just don’t fit anywhere else. If you find yourself considering this approach, the resources in question are more appropriately suited to an OU.

In most cases, you find that your situation calls for either a geographic or a division-based (functional) model. Both of these models tend toward stability. You find out in the following section that these models are also suitable when you’re defining OUs.

The multiple forests model

Up to this point, everything you’ve seen has been within a single forest. However, when are multiple forests necessary? Deploying multiple forests is definitely the least attractive design. In this model, shown in Figure 5-7, objects are split across multiple Active Directory forests. Because multiple forests are involved, the global catalog, configuration partition, and schema aren’t shared. Because the GCs aren’t shared, users can’t search for objects in the other forest. Each forest also has its own forest root domain and, therefore, separate Enterprise Admins groups.

WARNING!

Because of these facts, the multiple forests model can be extremely difficult and costly to administrate. Select this model only after careful consideration.

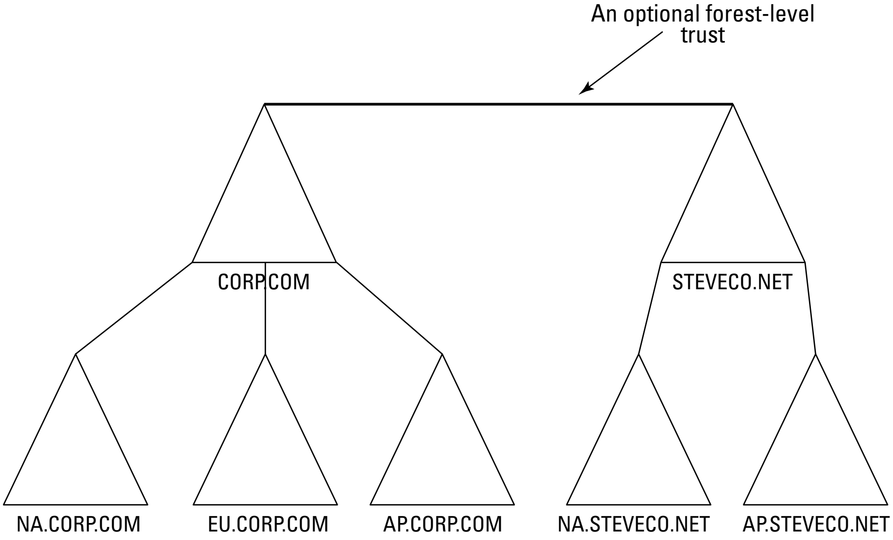


Figure 5-7: The multiple forests model showing a forest-level trust.

In the multiple forests model, you have the option of manually establishing transitive Kerberos forest-level trusts. These are similar to the trusts within a forest (but at a forest level) and are established between the forest root domains in each forest. Without these trusts in place, users in each forest only have the ability to access resources in their own forest.

WARNING!

A forest-level trust is transitive between the domains in each of the two forests that the trust connects. Forest trusts are not transitive at a forest level. So if you have a forest-level trust between Forest A and Forest B and another forest trust between B and C, the domains in Forest A will not implicitly trust the domains in Forest C.

Organizing with OUs: Containers for Your Trees

An *organizational unit* — or *OU* — is a logical container that you use to arrange groups of objects for convenient administration and access. You contain OUs within a domain. They can't span multiple domains nor can they contain objects from other domains.

OUs can contain the following items:

- Users
- Groups

- Printers
- Computers
- Network file shares
- Nested OUs

Generally, OUs are an efficient way to organize corporate resources because, like domains, you can arrange them hierarchically and they can accommodate various organizational models. An OU can easily assume the role of a resource domain but without the expense of additional hardware.

However, you can have too much of a good thing! Make sure that you minimize the depth of your OU structure. The deeper the overall AD structure, the more performance degradation you experience. Try to limit the OU structure to a maximum of three layers.

Creating a structure

Just as you examine your organization's policies and business model before you define domains, you should do the same before defining an OU structure. Although you can vary the OU hierarchy from domain to domain, doing so isn't a good idea. Ideally, you should make the OU hierarchy consistent and easy to understand. Varying the structure from domain to domain is certain to generate many help desk calls while users try to locate resources.

The following list suggests some OU models that you want to consider for your AD structure:

- Administrative
- Cost-center
- Project
- Division/department
- Geographic
- Object

The models that most people commonly adopt are administrative, division/department, and object. The object model, shown in Figure 5-8, is very appealing. Think how easily you can manage changes to large numbers of similar objects! Cost-center and project-based OU models tend to be less stable. Projects and cost centers come and go; when they do, the administrative burden increases because the administrator must add and remove OUs.

Planning for delegating administration

In contrast to domains, which set administrative boundaries, OUs provide opportunities for distributed administrative authority. At the OU level, you can specify an administrator's rights to create new user or group accounts, to create or modify specific objects, and to grant access permissions to container objects. You can even control whether users can see an OU!

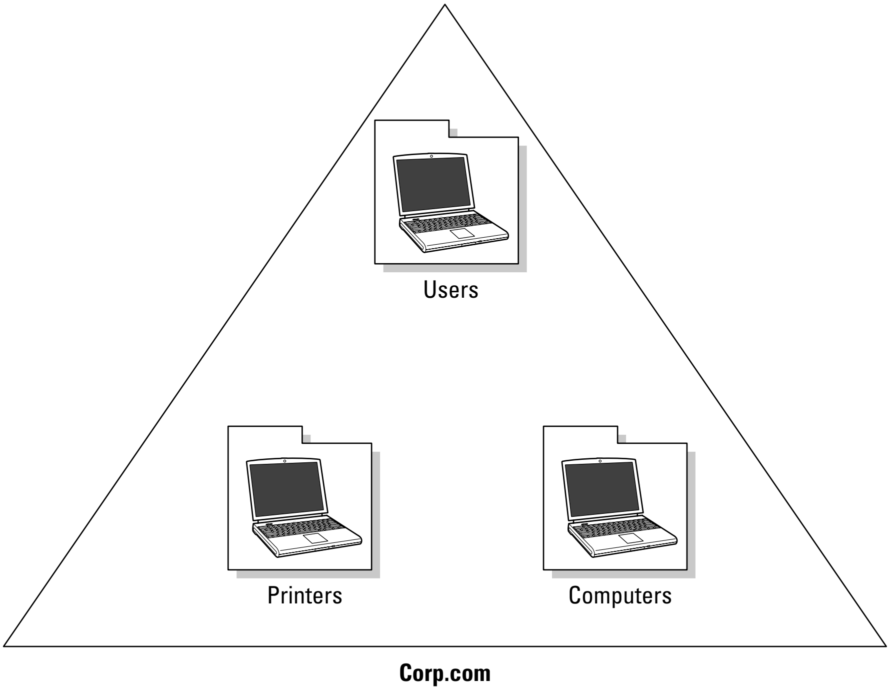


Figure 5-8: An example of an object-based OU model.

If your organization already has a recognizable administrative model, you can map the appropriate administrative roles to the OU structure. However, in many cases, administrative roles within an organization aren't clearly defined. Instead, they evolve over time, and you have no discernable model to follow. In such a case, answer the following questions while you create each new OU to help you plan how to delegate the appropriate level of authority:

- How are people going to use the OU?
- Who is going to administer the OU?
- What level of rights does the OU administrator require?

In Chapters 11 and 14, I get into the specifics of managing users and groups and of security. For now, during the planning phase, your concern is more with planning where and how you'll delegate administrative privileges than with managing Active Directory objects.

Ideally, the administrative model is the most important factor in determining how your logical structure takes shape. For each OU, ask, "Who is going to administer this OU?" Sometimes, you have highly specialized engineering teams who prefer to administer their users and resources. Place their resources in an OU or OU structure and then assign the appropriate administrator to the role. Frequently, you see administrative privileges assigned according to geographic location. For example, administrators in the Asia Pacific region might have administrative control over resources in their region of the world, whereas U.S.-based administrators have control over North American resources.

Consider the skill level and job role of each member of your administrative team when you decide who will have administrative authority over each OU. A help desk could be given password reset privileges over an OU containing users; highly skilled system administrators could be assigned to the schema administrator and enterprise administrator roles; and a department supervisor could be given administrative authority over a specific share. Be sure to assign privileges that are appropriate to the role. (See Chapter 11 for details on how to assign privileges.)

Planning your logical AD structure requires such attention to detail. Planning comes before implementing! Be sure that you complete the following steps before you create domains and OUs:

1. Using the DNS namespace, identify and name the root domain.
2. Determine whether a tree or a forest is appropriate for your organization.
3. Determine whether you need additional domains.
4. Consult your company's organization chart to decide which domain model is best for your needs and whether you need additional child domains.
5. Analyze the business models and processes in your organization to determine which OU model is best for your needs.
6. Determine who is to administer each OU.
7. Decide what administrative privileges OU administrators require.
8. Create a diagram for your logical AD structure that shows the domains and OUs required for your organization. For assistance in this process, review the figures I provide in this chapter.

Chapter 6

Getting Physical

In This Chapter

- Physical components of AD
- Designing an AD site topology
- Read-only domain controllers

The topics I discuss in Chapter 5 deal with the logical Active Directory structure. The logical structure attempts to match the AD design to an organization's business model and processes as well as to your administrative model. In this chapter, I discuss the physical aspects of Active Directory.

If you've sketched a logical structure design (as I suggest in Chapter 5), you've taken the first steps in matching your proposed AD design to your organization's business model. Now you have to deal with the physical structure. The physical design of Active Directory takes into account the network environment within which Active Directory is being deployed so that Active Directory performs from the user's perspective as well as from the AD replication perspective.

In this chapter, I tell you how to map your existing network infrastructure and how to adapt your logical AD structure to suit the network. I also walk you through planning domain controllers, sites, and site links.

The Physical Side of Active Directory

Active Directory is a directory service of objects. Some of these objects represent abstract items, such as domains and OUs; however, a set of objects in Active Directory corresponds to the directory service's physical environment. Having a physical side of Active Directory to design and configure provides some intriguing possibilities. No longer are you limited to engineering the physical network to handle the network traffic that a directory service generates — now you can design and control the traffic to meet the network topology and bandwidth limits. (I discuss gathering information about the network in Chapter 3.) For example, even in a hub/spoke topology network where some or all of the network links that make up the star are heavily used, you can still implement a single domain and control how Active Directory utilizes these links. Even though the logical and physical sides are separated, you still need to understand how the logical and physical sides match up because they can still affect each other in some circumstances. Fortunately, in most cases, each side of Active Directory is unconstrained by the other.

Before I discuss how to design the physical portion of Active Directory, here's the lowdown on aspects that Active Directory's physical design affects:

- **Directory replication:** One of the primary areas influencing the design of the physical side of Active Directory is the management of the traffic generated by AD replication. With a proper design, replication traffic can be controlled and managed so that the impact of the additional load created by Active Directory is minimized.
- **Localizing authentication:** When users log into an AD domain or access a resource, such as a file share, the user must get authentication services from Active Directory. The physical design of Active Directory affects traffic related to the user authentication and controls where the user gets those services by directing the user's request for authentication to a domain controller (DC) that is local (in network terms) to the user's desktop. This can help speed up authentication and keep slower network links from being used for authentication when it isn't necessary to do so.
- **Security:** Domain controllers contain data so sensitive that if it fell into the wrong hands it could seriously compromise your AD environment. That is why securing DCs is so important. One part of doing the physical design is determining where DCs are placed. In the past, placing DCs at smaller, less secure locations required a balance between localizing AD services for those users and risking the security of your Active Directory. Fortunately, Microsoft has a great solution for this in Windows Server 2008 that I discuss in just a moment!
- **Directory-aware application:** Active Directory and other dependent applications have evolved; therefore, a tighter integration has emerged. Now, the physical design of Active Directory can influence how you deploy these applications, and the manner in which you wish to deploy an AD-aware application affects the structure of your AD physical design. One great example is Microsoft Exchange Server 2007. Active Directory provides the directory service for Exchange, which includes providing the *Global Address List* — the list from which Exchange users

> pull e-mail addresses. In Exchange 2007, the physical design of Active Directory affects which Exchange servers other Exchange servers can directly communicate with. So, if you know that you're deploying AD-aware applications, it's important that you understand the implications that application has on your AD physical design (and vice versa).

Active Directory Physical Components

The process of creating an Active Directory physical design is not that much different from creating a logical design. You don't design in a vacuum. Instead, you look at the business and technical environment and at the company's needs, and you develop a topology designed to meet those needs. Before I discuss how to create a physical design for Active Directory, here are the components that you design with:

- Domain controllers
- Global catalog servers
- Sites
- Site links

Domain controllers and global catalog servers

I discuss domain controllers in the previous chapters, but to reiterate, a *domain controller* (DC) is a Windows server that provides directory and authentication services within an Active Directory Domain Services environment. Domain controllers in AD DS can be assigned either through the Initial Configuration Wizard or through the Server Manager tool in Windows Server 2008.

Domain controllers optionally can be *global catalog servers* as well. A *global catalog* (GC) stores a listing of every object within the AD forest. Normally, when a user logs in, a global catalog server must be accessed. It's important that you designate enough of these domain controllers as global catalog server so that users can always reach a global catalog.

One of the enhancements that Active Directory has included since Windows Server 2003 is the ability to cache a user's universal group memberships. Normally, this information is available only on a global catalog server, which is one of the reasons why a GC must be found during the login process. However, when universal group caching is enabled at a site, this information can temporarily be stored on other, non-GC domain controllers, making the necessity of contacting a GC less significant. Chapter 7 looks at how to enable this caching.

Active Directory sites

An *Active Directory site* is an AD object that represents a grouping of one or more TCP/IP subnets that are well connected and have remote procedure call (RPC) connectivity available between them. Domain controllers are placed within a site as a part of the DC promotion process. You can also manually move domain controllers between sites if necessary.

TIP

Network links that don't support RPCs typically include analog dialup links and ISDN connections. However, most network links, both LAN and WAN, do support RPCs these days.

AD client computers (both desktop/laptop computers and non-DC servers), on the other hand, are dynamically associated with a site at boot time. The domain controller, as a part of the client boot process, compares the client's IP subnet with the AD site definitions and determines which site the client is a member of.

The AD client computers will always attempt to get their AD services from DCs and GCs that are within the same site if possible before searching for them in other sites. This way AD requests are localized within the site as much as possible.

Subnets

TCP/IP-based networks are broken up into individual subnets connected by routers. Each of An IP subnet address and a subnet mask defines each of these subnets. Every computer in a TCP/IP-routed network (both servers and desktops) must be located in a particular TCP/IP subnet.

REMEMBER

Although I wish I could spend a whole chapter on what subnets and subnet masks are, there just isn't enough room in the book. So check out *TCP/IP For Dummies* by Candice Leiden (Wiley) to find more.

When you deploy Active Directory, it's necessary to create subnet objects for each TCP/IP subnet where all the AD domain controllers will be installed and where the subnet objects for all the clients of Active Directory will be located. These subnet objects then are associated with the site object. A subnet object can only be associated with a single site within Active

Directory. Associating subnets with a site allows the domain controllers and AD client to determine which site they're in. They do this by comparing their current subnet with the corresponding subnet object and then finding which site the subnet object is associated with.

Site links

Site links represent the Active Directory replication paths between sites. These paths are manually defined so that the designer has control over which network links the replication traffic occurs on. These site links also control how clients are directed to domain controllers when there's no DC in the client's local site. Each site link has the following attributes:

- **Connected sites:** A site link is defined by the sites to which it connects. A site link can connect two or more sites together.
- **Network transport:** Site links support replication communication over IP-based RPCs or with the Simple Mail Transport Protocol (SMTP). You normally want to use RPC whenever possible, but you can use SMTP when the sites you're linking don't support RPC.
- **Cost:** Each site link has a cost associated with it. Costs are used to assign preferences to links that determine which link should be followed when multiple link paths are available between sites. The cost represents what it "costs" to use this site link relative to the other site links and affects replication traffic as well as how users are assigned a domain controller. Links with lower cost values have preference over links with higher cost values. Cost values range from 1–32,767; the default being 100.
- **Frequency:** The frequency value defines how often a replication occurs when using this site link (the replication latency). You can configure the time between replications from a minimum of 15 minutes to a maximum of 10,080 minutes (one week). The default frequency is 180 minutes.
- **Schedule:** The schedule dictates when this link is active and available for replication between the sites. The schedule can also control which days of the week the link is available. Normally, the schedule is set so that the link is available 24 hours a day, but you can set up different schedules on a per-day-of-the-week basis.

By creating a site link, you enable two or more sites to be connected and to share the same site link attributes (transport, cost, frequency, and schedule). By default, site links create transitive connectivity between sites. If you create a site link between sites A and B and another site link between sites B and C, an automatic connection (known as a *site link bridge*) is created between sites A and C, as shown in Figure 6-1.

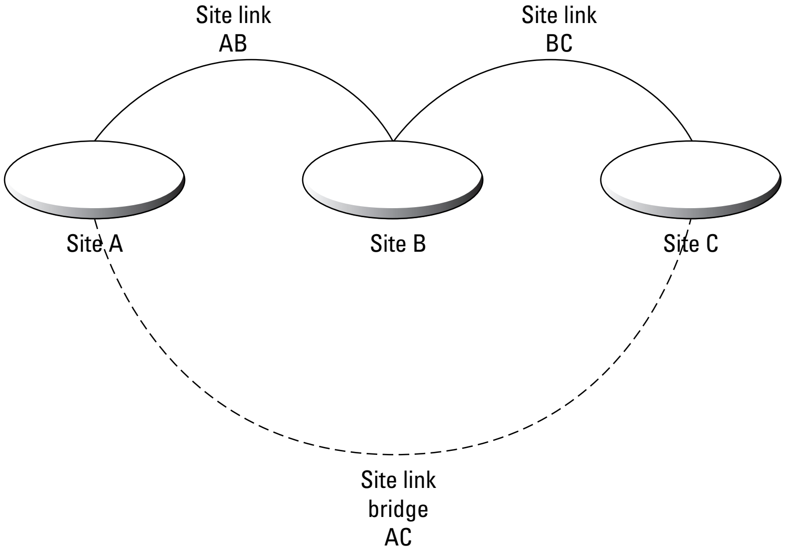


Figure 6-1: Three sites connected with two site links and one site link bridge.

Designing a Site Topology

After you've created a representative diagram of the network, you can start planning your AD sites and site links. Although the logical structure that you create in Chapter 5 helps users locate resources easily, it doesn't consider the network. By defining sites based on the network infrastructure, you can segregate authentication and replication traffic so that it doesn't traverse WAN links. This configuration leads to better response time for user logons and helps optimize costly WAN utilization. Your first step in planning your site topology is deciding where the domain controllers and global catalog servers need to be placed within the physical network.

Placing domain controllers

The location of domain controllers within the network is primarily determined by the location of the users within the network. This allows you to ensure that the requests for AD services go to a local DC and won't have to traverse WAN links. These services consist of both reading and writing to the directory store as well as the authentication services that Active Directory provides. Because authentication and directory access are core AD services that you provide to users, the placement of DCs can greatly affect the user's perspective of Active Directory's reliability and performance.

Consider the following factors in determining domain controller placement:

- **Sites with 50 or more users can justify a local domain controller.** Microsoft promotes this guideline within its documentation. Theidea is that a branch office with 50 or more users generates enough demand to benefit from a local domain controller. However, this does have to be weighed against the speed of the WAN connection to that office so that if you have a fast connection, you more easily use a remote domain controller than if you have a slow connection.
- **The closer a domain controller is to a group of users, the more you localize the network traffic relating to authentication and reading/writing to the directory.** This is especially true if the users at this location are accessing local resources.
- **For larger sites, make sure that multiple domain controllers are available for users in that site.** The increased authentication and directory-access traffic generated by users of that site can be distributed across domain controllers.
- **Deploy remote domain controllers only if your administrative model supports doing so.** If you have a centralized AD administration staff and you're considering deploying remote domain controllers, make sure that you can remotely administer these domain controllers. Using Terminal Services for remote desktop access to the DC and using the other AD administration tools allows you to perform most tasks remotely, but some things — especially hardware issues — can be done only by someone local. So make sure that you have someone local to the DC who can perform administrative tasks when required.
- **Consider where AD-dependent applications are running.** Many server applications that utilize Active Directory may require that DCs be located on the same subnet as the application server because of the demands that the application places on Active Directory. Microsoft Exchange Server is one good example.
- **The security of the physical computer running the domain controller role is critical.** There's enough data on any individual domain controller that, if compromised (that is, if a hacker gets access to it), the security of the entire AD forest is at risk. If the site doesn't have a secure data center to house the domain controller, you might have to consider not placing a DC on that site.
- **The administrator of a domain controller is normally the administrator of the domain.** When a server becomes a domain controller, it no longer has its own local security database. So if your administrative model requires that a site's administrators have administrative control over all local DCs, then those administrators typically will have to be granted administrative authority not over just the DC but over all DCs in the domain and all objects in that domain.

Later in this chapter, I look at a new deployment option for domain controllers in Windows Server 2008 called Read-only Domain Controllers (RODC). RODCs are designed to address the issues in these last two bullets.

TIP

Using the above guidelines, you should examine every location and data center that you have and consider whether that location should have local domain controllers. No matter the location of users in the physical network, you need to determine where those users get their domain controller services. Users get these services from either a local or remote domain controller. Place domain controllers as near to the users as you can, but make sure that the placement makes sense.

TIP

Every AD domain needs at least two domain controllers installed to provide redundancy. Then if one domain controller crashes, you still have at least one operating DC in the domain.

Placing global catalog servers

After you've determined where you're placing your domain controllers, the next step is to decide which of these DCs will be designated as global catalog (GC) servers. The general rule is that you need at least one GC per location because of the clients' need to contact a GC at logon time. However, if you have a multiple domain AD forest at a smaller location, keep in mind that turning a DC into a global catalog server is going to increase the amount of replication traffic to that DC. As I mention earlier, you do have the option of enabling universal group caching for that location, so you don't have to deploy a local GC when network bandwidth is a concern.

Placing operations masters

Although domain controllers are usually peers with each other in reading and writing directory information, certain roles cannot be distributed across all the DCs. Rather, they must be designated to be on only one DC in either the domain or forest. These roles in Active Directory are known as o*perations master* roles. Although the placement of operations masters isn't as critical as the placement of DCs and GCs, you still want to carefully analyze the situation and make sure that your placement makes sense. There are five operations master roles (also known as *Flexible SingleMaster Operations* or *FSMOs*):

- **Schema master:** I mention the schema master in Chapter 1, but to reiterate, the domain controller that's designated as the schema master is the only domain controller where changes to the schema can be made. There's only one schema master role holder DC in an AD forest.

- **Domain naming master:** When a new domain is added to an AD forest, it's necessary to verify that there isn't already an existing domain with that name in the forest. The DC holding this role is contacted when adding a domain to a forest to verify that the domain doesn't already exist. There is only one domain naming master DC in a forest.
- **PDC emulator:** In Windows NT (before Windows Server 2000 and Active Directory) the NT Directory Services (NTDS) included a domain model in which a primary domain controller (PDC) incorporated all changes, and backup domain controllers (BDC) formed a read-only copy of the PDC to provide additional instances of the directory for performance and redundancy. To provide a migration path from NTDS to Active Directory, an AD domain controller in each AD domain had to be designated to hold the PDC emulator role. Computers still running older NTDS-compatible operating systems (such as Windows NT) need to be able to identify a domain controller providing the PDC role. So, by designating one of the AD DCs to be the PDC emulator you allow for those older computers to work with AD. The PDC emulator also assists with the password change process by providing a single guaranteed DC that always has the current password of all users in the domain. Imagine a scenario in which you update your password on one domain controller but then attempt to authenticate to a different domain controller before your password change has synchronized. In that situation, your attempt would fail because the passwords would not match. Active Directory enables a high priority synchronization of password changes to the PDC emulator. That way, when a DC fails on a authentication, it checks with the PDC emulator to see if it has a more up-to-date password.

  Last, the PDC emulator acts as the time synchronization master for the domain (the PDC emulator for the forest root domain is the time synchronization master for the forest). Active Directory clients must synchronize their clocks with the domain controllers so that Kerberos works properly and the PDC emulator provides this time source for the clients so that this synchronization will occur.
- **RID master:** Every object that can have rights assigned to it (known as a *security principal*) has a Security Identifier (SID). This SID identifies the object (like a user or group) to the Windows security subsystem. SIDs are composed of a domain SID and a relative identifier (RID). The domain SID is the same for all security principals in the same domain. The RID is the unique part of the overall SID of each security principal. Because a security principal can be created on any domain controller, each DC needs a way of creating unique SIDs so that two DCs don't hand out the same SID to two different objects. This is accomplished by creating pools of RIDs on each domain controller. The RID master is the DC that's responsible for handing out these RID pools. Each domain has an RID master.
- **Infrastructure master:** The infrastructure master in each domain is responsible for controlling the user-to-group mappings when the users are in multiple domains.

The primary guideline to follow is to place forest-level roles (schema master and domain naming master) on a DC that's near the AD administrators. Additionally, the domain naming master DC should be placed on a global catalog server. The PDC emulator for each domain should be placed in a well-connected site so that other DCs can reach it. This is also the guideline to follow for the RID master; therefore, in most cases you can simply place the PDC emulator and RID master on the same DC. With the infrastructure master, if the forest has multiple domains, you shouldn't place this role on a global catalog server because this role's function is reconciling user-to-group memberships. If the forest has only one domain, then you can place this role on a GC if you wish.

TIP

You might be concerned with the fact that there can only be one instance of each operations master. The question is, "What happens if a DC holding one of these roles fails?" Never fear. The good thing is that users don't depend on these roles being available for day-to-day operations. If a DC holding one of these roles fails and you cannot restore it, you have tools available to move the role to another DC. The important thing is to monitor your AD environment so that you know if a DC holding one of these roles has failed so you can take corrective action.

Defining Active Directory sites

When the domain controllers, global catalog servers, and operations masters are placed, you're ready to define your AD sites. An AD site is a group of one or more TCP/IP subnets that are well connected and have RPC connectivity between each other. *Well connected* usually means network links are at LAN speeds (10 Mbps) or faster.

Each of the DCs you place must be located within an AD site. Therefore, the first step is to figure out the subnets on which you placed your DCs. Then for each of those subnets, determine which other subnets are connected to the subnet you located within the local LAN. This grouping of subnets represents a single AD site. If you have DCs at multiple locations separated by a WAN, this process should yield you multiple AD sites.

The second part of this process is to look at the subnets where all the AD client computers reside. These clients are both non-DC Windows servers that are members of the forest and all the user desktops. In some cases, these subnets are already part of the AD sites you previously defined for the DCs. But for AD sites that don't have a DC within them, the first step is to reevaluate those locations and, following the DC placement guidelines above, make sure that these locations still don't need a local DC. After doing that, then you need to decide whether this location should be a separate site. This isn't an exact science. If this non-DC location is connected to another location that has domain controllers, you can just add the non-DC location's subnets to that location with a DC, which effectively collapses the two locations into a single site. If this non-DC location is connected to multiple locations, then you can create a site for that location and define how the clients at that location find a DC by the site link topology (discussed in the next section).

Figure 6-2 illustrates this mapping of networks to AD sites. In this figure you have two LANs separated by a WAN link. Within each LAN you have a well connected network that has 10 Mbps or better network speeds, but between the LANs the WAN connection is somewhat slower. So you can create an AD site boundary around each LAN (Site A and Site B).

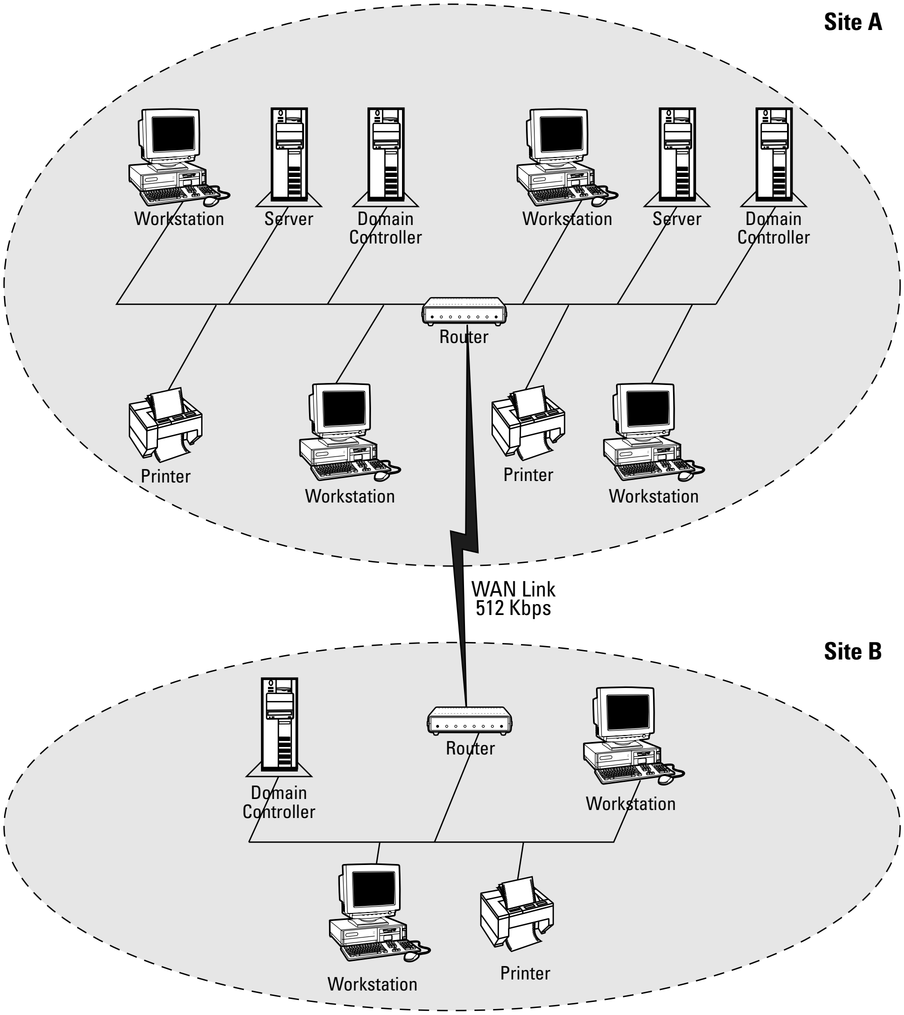


Figure 6-2: Site boundaries compared with the physical network.

You might have a situation where you have WAN connections that are actually faster than 10 Mbps, such as an OC-3 connection, for example. Does that mean you can span your AD sites across this link? If the connection is always available and isn't heavily used, you can span an AD site across this link. However, keep in mind that the AD replication traffic within a site is more frequent than between sites. Therefore, if you're billed by the amount of data placed on this connection, you might not want to spend the money necessary to span an AD site across this link.

Creating Active Directory site links

After you create the sites, you create the site links that provide the replication and authentication paths between the sites. Deciding which sites to connect usually isn't a difficult task. In most cases, you simply mirror the physical network structure so that the sites connect in the same topology as the network (see Figure 6-3).

Remember that a site link can connect more than just two sites. In many cases, companies have deployed a hub-spoke network topology. In this model, each remote location on the end of a spoke is a site and the central hub is a site. With this site structure, you can use a single site link that connects all the sites (assuming that the available bandwidths and transports are the same between the hub and each location). These two structures are shown in Figure 6-4.

What if the network links require different transports and various available bandwidths? That brings me to the more complex part of creating this topology: the configuration of the site link attributes. Remember that each site link has a transport, cost, frequency, and schedule attribute that can be configured. The attributes have to be matched to the restrictions that the network links present and still satisfy the company's requirements regarding performance.

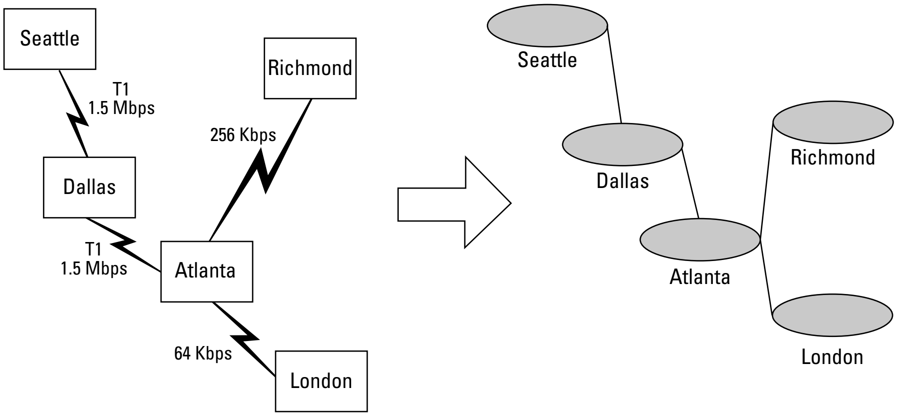


Figure 6-3: Transforming the physical network to the site topology.

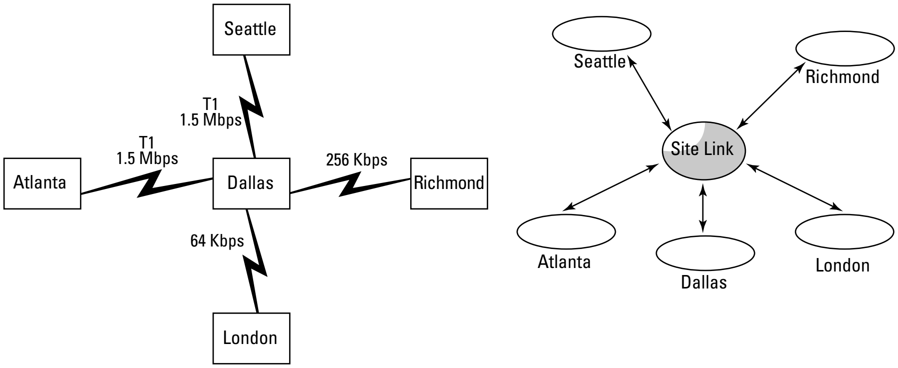


Figure 6-4: Transforming a hub-spoke network to a site topology.

For the transport, in almost all cases, you should be using RPCs (also known within the AD tools as *IP transport*). Only use the alternative, SMTP, when RPCs aren't supported on the network link(s) between the sites (this is a relatively rare situation).

You should set the costs of the links relative to the network link speeds that the site link uses. This shouldn't be the only factor that you consider when defining the site link costs. Reliability of the link, the amount of available bandwidth, and the actual monetary cost of using the network link should also figure into your calculations. If you group the link speed, available bandwidth, and link reliability into an abstract measurement of the network link's quality and compare this with the monetary cost of using this link, you can plot this site link cost on a graph (see Figure 6-5). When the link quality is low and the monetary cost of using the link is high, the site link cost is high. However, when the network quality is high and the monetary cost is low, the site link cost is low as well.

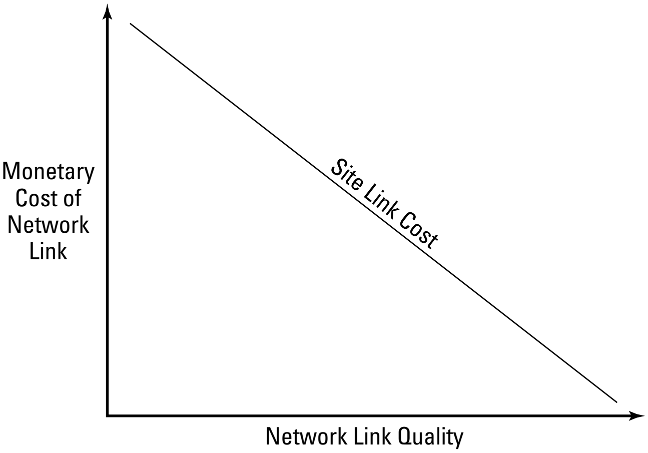


Figure 6-5: The effects of network link quality and monetary costs on site link costs.

Use frequency and schedule settings so that directory replication occurs only when you want it to. You need to measure the company's requirement for timely directory updates against the available network bandwidth. Be sure to consider the difference between what the frequency and schedule settings control. For example, if the network loads are relatively light, you should be able to schedule replication to run all the time. If you have a medium loaded network but still need to replicate information on a regular basis, you can set the frequency time to a higher value than the default 15 minutes but still keep the schedule set for 24x7. But, if the network load is heavy and traffic patterns indicate that loads are lighter at certain times of the day, you might want to schedule replication to occur only during those lighter times.

TIP

I want to remind you about the automatic creation of site link bridges. If you have a network topology in which the IP network isn't fully routed, you should turn off the site link bridging so that bridges don't get created and then attempt to establish replication paths between DCs that can't communicate with each other.

Read-Only Domain Controllers

Now is a good time to discuss a new domain controller deployment option that you have available in Windows Server 2008: the ability to create Read-only Domain Controllers (RODCs). This type of domain controller is designed to address some of the typical issues that arise in deploying a DC in a branch office environment. Earlier in the chapter, I mention that physically securing the DC is critical because the data on a DC can be used to compromise the security integrity of the entire AD forest. A normal domain controller contains a software key used in the creation of Kerberos tickets that can be used to access resources within the forest. If that key got into the wrong hands, by someone stealing a DC from an unsecured location, it could be used to gain unauthorized access to files, Web sites, e-mail, and so on.

TECHNICAL STUFF

This software key is used to create Ticket GrantingTickets (TGTs) that are given to users upon a successful logon to Active Directory. With a TGT, a user can then obtain a Kerberos session key that allows the user to access a particular resource like a file share. For more on Kerberos, see Chapter 14.

RODCs are designed to prevent this scenario of this key falling into the wrong hands as well as several others, including:

- **Administrative separation:** It's possible to create a server-level administrator for an RODC instead of giving the local administrator Domain Admin access, which is what's normally done with a writable DC.
- **Reduced replication:** Because the DC is read-only, it doesn't have as much replication traffic to deal with. This can be valuable to a branch office site that doesn't have a lot of network bandwidth back to the rest of the company.

When you create an RODC you create a DC that accepts only incoming AD replication (as well as incoming DFSR replication, which I talk about in Chapter 14). This incoming replication is the only way that updates can be written to an RODC. A local administrator cannot make changes to an RODC's directory store nor can LDAP writes be done to this server. To address the security issue with the Kerberos key, each RODC has a unique key rather than the common key that each of the writable DCs in each domain has. Another security issue that RODCs address is the compromise of user passwords. User passwords aren't normally stored on an RODC by default. By doing this, if a hacker compromises the RODC, none of the user passwords is available to that hacker.

REMEMBER

Okay, so maybe a read-only DC isn't really read-only if changes can be written to it via AD replication. The point is that, other than normal AD replication, there's no way to make direct changes to the RODC's copy of the directory. Maybe a better name would have been *Read-only except for AD replication Domain Controllers (ROEADRDC)*, but I guess someone found *read-only DC* simpler.

RODCs appear to be a great solution for branch offices. They are, but RODCs aren't without their prerequisites, limitations, and additional administrative issues, which I cover next.

RODC prerequisites and limitations

Planning and infrastructure requirements must be carefully considered before you begin deploying RODCs in your AD environment. RODCs are a new option available only in Windows Server 2008. Therefore, if you have an existing AD infrastructure deployed on Windows 2000 Server or Windows Server 2003, you have to address the following issues first:

- **A Windows Server 2008 PDC emulator role holder:** RODCs are supported only in a domain in which the PDC emulator operation role holder is a Windows Server 2008 domain controller. By observing this, you resolve an issue with the fact that an RODC cannot advertise itself as a time-sync source for clients. If your current PDC emulator is running an earlier OS version, you need to either upgrade that DC or move the role to a Windows Server 2008 DC.
- **A Windows Server 2008 DC that is the replication partner of the RODC:** The RODC must get its AD directory updates from a DC that understands RODCs. Normally, a DC that replicates with another DC is expected not only to send updates to the partner DC but also to receive updates from the partner DC. An RODC can only receive updates from a writable DC. Because it's an RODC, it never sends updates back, and the replication partner of an RODC needs to understand this. Only Windows Server 2008 DCs have this ability.

- **A Windows Server 2003 native mode forest or higher:** RODCs are supported only in an Active Directory that at least has all the DCs running Windows Server 2003 and has the forest-level setting at 2003 native mode. Note, though, that some RODC features aren't available until you upgrade all the DCs to Windows Server 2008 and set the forest level to 2008 native mode.
- **Must run ADPREP/RODCPREP:** So that DNS replication is properly supported in a forest with RODCs, you must run the `ADPREP/RODCPREP` command against the AD forest. This is only a requirement, though, when you're deploying in a preexisting forest with DCs running anything earlier than Windows Server 2008. I talk more about running DNS on an RODC in the next section.

RODCs simply cannot do some things because of their read-only nature. An RODC cannot be an operations role holder. Additionally, you can't design your AD site topology in such a way as to depend on the RODC to replicate information to other sites. Normally, when you deploy an RODC, it's placed at the periphery or endpoints of your network. That is, if your AD site topology (as well as the network topology) is a hub-spoke design, you place RODC only at the ends of the spokes (see Figure 6-6). By doing this, you ensure that other DCs aren't dependent on getting replication updates from the RODC.

Normally, you place RODC in a branch office by itself but having other DCs in the same site as the RODC is also supported.

WARNING!

I wish that within the confines of this book I could cover every issue related to RODCs, but I simply can't. If you're going to consider deploying RODCs, make sure that you thoroughly research the subject before doing so. Microsoft has a wealth of information about RODCs in the Windows Server 2008 TechCenter. The URL for this Web site can be found in Chapter 17.

Running DNS on an RODC

If you plan to run DNS on an RODC, you need to understand how dynamic registration works on this type of server. Because an RODC is read-only, if the DNS zone is AD-integrated, the RODC doesn't have a way of directly writing the resource records for the requesting client to the zone because it's in Active Directory. Instead, an RODC sends the client a DNS referral pointing to another DNS server that isn't an RODC. The client can then perform the dynamic registration with that DNS server. Most likely, this writable DNS server is outside of the branch office where the client and RODC are located. Additionally, the RODC DNS server attempts to pull the updated DNS records from the writable DNS server so that the local copy of the zone at the branch office is updated. The result of all this is additional network traffic related to DNS.

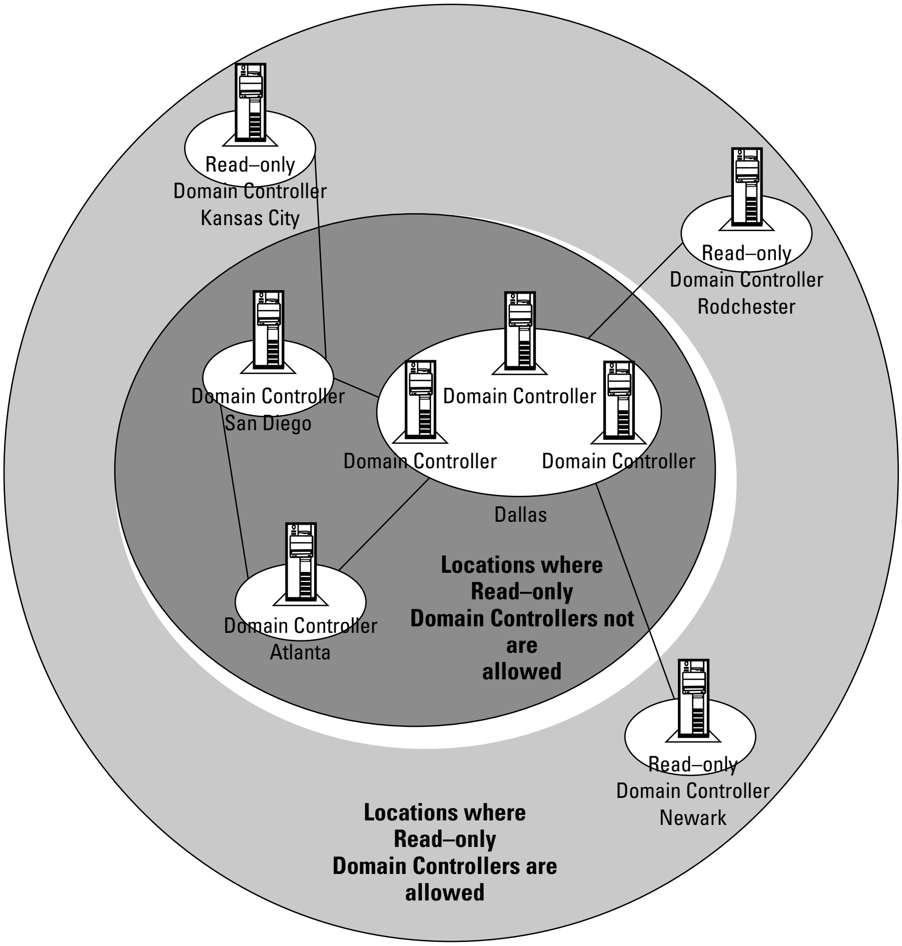


Figure 6-6: RODC placement in an AD site topology.

RODC administrative separation

One of the great features of running an RODC is that you can administratively separate (or delegate) the support of the RODC from the rest of the DCs in the domain. Normally, on a writable DC, if you need to administrate a DC, you must be a member of the Domain Admins group within the domain. However, placing an administrator in the Domain Admins group makes him an administrator not just of that particular DC but also of all the DCs in the domain. If you want to deploy a DC at a branch office and have it locally administrated, RODCs provide a way for that local administrator to support that RODC without the need to include them in the Domain Admins group.

You can delegate this administration two ways. First, this can be done at the time the RODC is created by using the `DCPROMO` command (I discuss this in Chapter 7). You can also create delegated administrators. To create a separate administrator for an individual RODC, you must use the `DSMGMT.EXE` command on the RODC, as follows:

1. **Open a command prompt (if not running a Windows Server 2008 Server Core server).**
2. **Type** DSMGMT **and then press Enter.**
3. **At the DSMGMT prompt, type** `Local Roles` **and then press Enter.**
4. **At the Local Roles prompt, type** Add <domain name>\<user name> **administrators, where <domain name> is the NetBIOS name of the domain where the user to administrate the RODC is located and <user name> is the logon ID of the new RODC administrator.**

RODC credential caching

As a security measure, RODCs normally don't contain the passwords for the users to prevent those passwords from being compromised if the RODC is stolen or broken into. There's a downside, however, of not storing these passwords locally — an RODC can't authenticate a user when she first logs into Active Directory. For a user to log in successfully, then, the branch office WAN connection must be working so that a writable DC can be contacted for login.

If your motivation for deploying an RODC isn't primarily security and you want to ensure that branch office users can log in when the WAN is unavailable, you can enable caching of passwords to the RODC. This is *credential caching*. Windows Server 2008 supports a password replication policy where administrators can control what passwords can be replicated to the RODC. This policy consists of two security groups: Allowed RODC Password Replication Group and Denied RODC Password Replication Group. By default, the Allowed group is empty but the Denied group contains a number of security groups (including Domain Admins and Schema Admins) so that the members of these groups never have their passwords replicated to a RODC.

The credential caching process for an RODC works as follows (see Figure 6-7):

1. A branch office user logs into an RODC.
2. The RODC determines that it doesn't have locally available credentials for the user and forwards the request to a writable DC.
3. The writable DC successfully authenticates the user and gives the user their Kerberos Ticket Granting Ticket (TGT) so that the user can access resources.
4. The RODC sees that the user has logged in and requests a copy of the hash of the user's credentials from the writable DC.

5. The writable DC receives the request and checks to see whether the user is a member of the Denied RODC Password Replication Group. If so, the request is denied.

6. If the user isn't in the Denied group, then the writable DC checks to see whether the user is in the Allowed RODC Password Replication Group. If the user is in the Allowed group, the credentials are sent to the RODC via normal AD replication and the credentials are cached. If the user isn't in the Allowed group, the credential request is denied.

7. The password is replicated from the writable DC to the RODC.

TIP

When you create an RODC with the `DCPROMO` command, you have an opportunity to modify the password replication policy as well.

The great thing about how this caching works on a particular DC is that the credentials are cached only if the user actually logs into that RODC. Therefore, if this RODC is compromised, only the passwords of users that have actually logged into this RODC are available. Also, if your RODC is stolen, the administrator can easily reset the cached credentials by deleting the RODC by using the AD Users and Computers tool. When you attempt to delete an RODC in this manner, you're prompted to reset all the passwords of the users whose credentials are cached on that DC. (See Figure 6-8.)

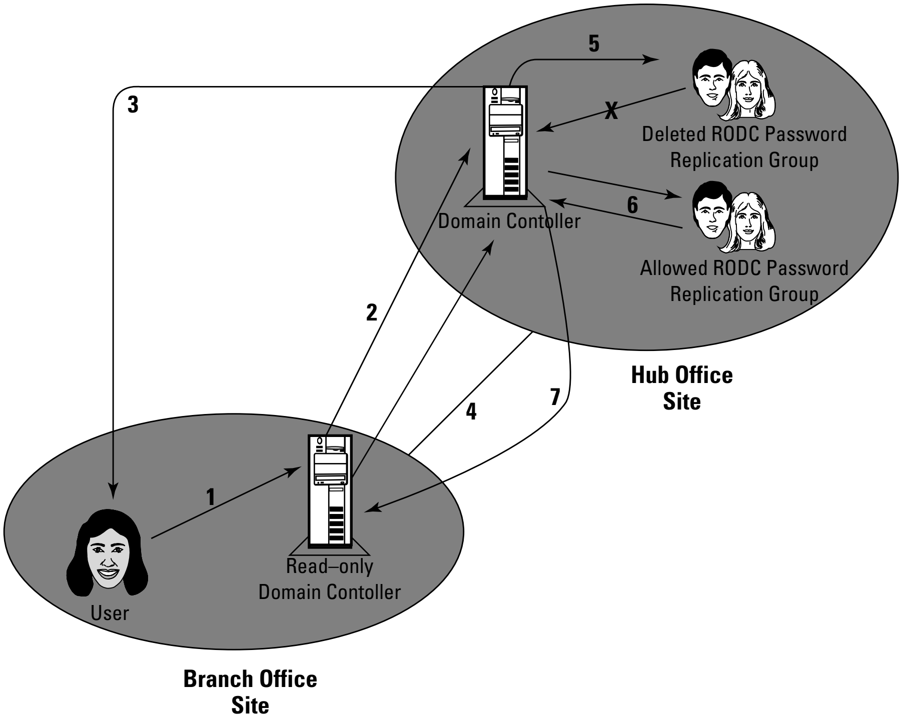
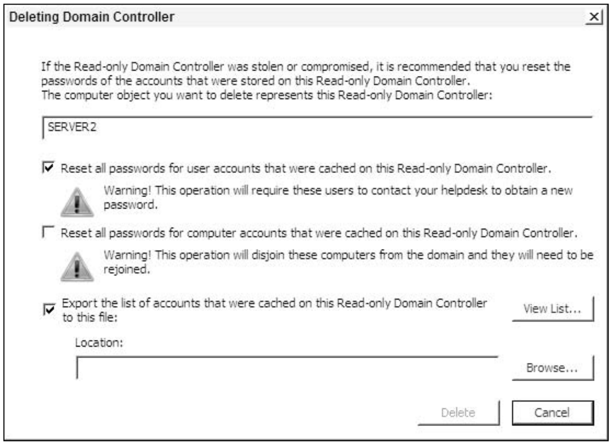

5
3
X
Deleted RODC Password Replication Group
6
Domain Contoller
Allowed RODC Password Replication Group
2
Hub Office Site
4
7
1
User
Read–only Domain Contoller
Branch Office Site

Figure 6-7: Credential caching on an RODC.



Deleting Domain Controller

If the Read-only Domain Controller was stolen or compromised, it is recommended that you reset the passwords of the accounts that were stored on this Read-only Domain Controller.
The computer object you want to delete represents this Read-only Domain Controller:

SERVER2

☑ Reset all passwords for user accounts that were cached on this Read-only Domain Controller.

Warning! This operation will require these users to contact your helpdesk to obtain a new password.

□ Reset all passwords for computer accounts that were cached on this Read-only Domain Controller.

Warning! This operation will disjoin these computers from the domain and they will need to be rejoined.

☑ Export the list of accounts that were cached on this Read-only Domain Controller to this file:

Location:

Figure 6-8: Resetting cached credentials on an RODC.

To be technically accurate, user passwords are never actually stored in Active Directory (on either writable or read-only DCs). Instead, a hash of the user password is stored. A *password hash* is a mathematically computed number that's generated by using the user's password in such a way that you cannot easily reverse compute the password by using the hash number. So when I speak of caching credentials, I really mean that the password hashes, not the actual passwords, are cached.

Chapter 7

Ready to Deploy!

In This Chapter

- Windows Server 2008 requirements
- Introducing Windows Server 2008 Server Core
- Attended and unattended DCPROMO
- Miscellaneous AD DS deployment tasks
- RODC installation

Earlier chapters of this book all deal with theory. It's time to put this theory to practice and start actually deploying Active Directory Domain Services in your environment. Here I cover how you can actually deploy AD DS on your servers (which means creating domain controllers) both in an interactive way as well as in an automated fashion. Then I look at some other topics related to deploying AD, including the creation of read-only domain controllers.

Installing Windows Server 2008

Of course, before you can begin the deployment of Active Directory, you need to install the operating system, which, in this case, is Windows Server 2008. The first step is to make sure that the hardware you plan to use as domain controllers is capable of running the OS. Table 7-1 lists both the minimum and recommended hardware levels for Windows Server 2008.

Table 7-1: Windows Server 2008 Hardware Requirements

Hardware Component	*Minimum*	*Recommended*
Processor	1 GHz 32-bit CPU	>=2 GHz 32- or 64-bit CPU
Memory	512MB of RAM	>=1GB of RAM
Hard Disk Space	8GB	>=40GB for Full Server Install; >=10GB for Core Server Install

If you're purchasing new servers to be domain controllers, you shouldn't have any difficulty in meeting or exceeding the recommended guidelines. If you're reusing existing servers, then just make sure that you're at least compliant with the minimum hardware levels. The good thing is that AD DS does not require a large number of CPUs or a massive amount of memory as it is not a computationally intensive application.

In addition to ensuring that you have the right hardware, you need to be aware of the various editions that Windows Server 2008 comes in. Although at the time of this writing Windows Server 2008 hasn't been released, it appears that five editions of Windows Server 2008 are going to be available. Table 7-2 lists the editions, the maximum number of supported processors, the maximum amount of memory (both in the 32-bit and 64-bit versions) and whether the edition supports AD DS as a server role (that is, can it be a domain controller?).

Table 7-2 Windows Server 2008 Editions

Edition	*Description*	*Max. # of CPUs*	*Memory (32 bit/64 bit)*	*Can be a DC?*
Web Server Edition	This edition is designed to act as an OS for Web servers only.	4 CPUs	4GB/32GB	No
Standard Edition	The primary edition of the OS that is used on most servers.	4 CPUs	4GB/32GB	Yes
Enterprise Edition	This edition is designed for servers running larger server applications (such as SQL databases) and supports clustering.	8 CPUs	64GB/2TB	Yes
Datacenter Edition	Datacenter is designed to support large enterprise-level applications and virtualization on a very high level.	32 CPUs (64 CPUs on 64-bit systems)	64GB/2TB	Yes
Itanium Edition	Edition for Itanium-based servers running large business applications.	64 CPUs	2TB	No

Server virtualization

Near the end of the Windows Server 2008 development cycle, Microsoft introduced a new feature called Hyper-V. *Hyper-V* is a server virtualization technology that allows you to run multiple virtual machines on the same physical server hardware. The advantages to virtualization include server consolidation and increased business continuity. Therefore, you might want to consider running your domain controllers as virtual machines. Virtualization is a great technology that can be used to set up labs and proof of concept environments. In Windows Server 2003 with MS Virtual Server 2003R2, Microsoft provides support for running domain controllers and I assume that this support will be extended to Windows Server 2008 with Hyper-V. Nevertheless, the vast majority of companies haven't readily embraced virtualization of domain controllers for various reasons, including concerns with the reliability and workload management. This attitude might change with Hyper-V, but we'll have to wait and see. Hyper-V will be available shortly after the release of Windows Server 2008.

When you are deploying AD domain controllers, you most likely are going to use the Standard Edition version of Windows Server 2008 because Active Directory really doesn't benefit from the additional CPUs, memory, and features (such as clustering) that the Enterprise Edition provides.

You should also consider whether you're going to run the 32-bit or 64-bit versions of the OS. Each of the editions that support AD DS comes in both 32-bit and 64-bit flavors. In running a 64-bit OS domain controller, you can realize some performance gains because you can cache more of the directory into the DC's memory than you can with the 32-bit OS option. However, these gains become noticeable only if you are running a large directory supporting 100,000 users or more. Therefore, if you're trying to decide between 32-bit and 64-bit options, the size of your directory can drive your decision. There are additional factors that can influence whether or not you go to a 64-bit OS including:

- Not all hardware devices have 64-bit drivers available for them just yet. If you have a device in your server like a network card or disk array controller that doesn't have 64-bit drivers available, you will have to stay with the 32-bit drivers and therefore the 32-bit OS.
- Examine any additional server applications that will be running on the domain controller. If any of these applications are designed for a 32-bit OS, verify that they can run in a 64-bit OS.

To Core or Not to Core

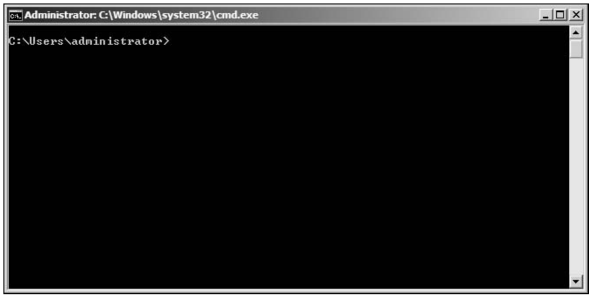
One of the cool things that Microsoft did with Windows Server 2008 was to provide two ways of installing the OS:

- A full installation that includes all the software to make the GUI-windowed desktop environment work with all the available server roles and features.
- A new Server Core installation that includes only the minimum amount of software and services required to make the server function. The Server Core installation provides you the capability to administrate the server with only a command line interface.

Server Core is intended to provide a way of deploying the server OS with a minimal number of software components, which yields two major benefits:

- The less software on the server, the fewer patches necessary for that software.
- With the minimum of services running on the server, you dramatically reduce the exposure points a hacker can exploit. Therefore, a Server Core installation is a more secure way of deploying a server.

The Server Core user interface is shown in Figure 7-1.



```
C:\Users\administrator>
```

Figure 7-1: The Windows Server 2008 Server Core user interface.

In addition to not having the GUI administration tools available, Server Core only supports a subset of the available server roles including:

- AD Domain Services
- AD Lightweight Directory Services
- DNS
- DHCP
- File Server
- Windows Server virtualization (Hyper-V)
- Windows Media Services
- Print Management

Even though you can't run any of the Microsoft Management Console (MMC) tools directly on a Server Core server, most of these tools allow you to administrate a remote server. Therefore, you can administrate a Server Core server with GUI tools as long as you're not running the tool on the Server Core server. But this will not completely eliminate the need to become more proficient with the command line tools as some administration will still have to be performed on the Server Core server itself.

Creating Server Core domain controllers (which support the AD DS role) is a great way to deploy because a Server Core installation is inherently more secure and requires less patching. The downside is that although you can run many of the MMC consoles remotely, you still have to be a lot more familiar with the command line tools than you probably are used to. But don't worry — I'm going to show you how to deploy a Server Core DC later in this chapter.

Appendix A contains a list of the more common command line tools that you need to know if you're going to utilize Server Core installations.

Deploying AD DS on a Full Server

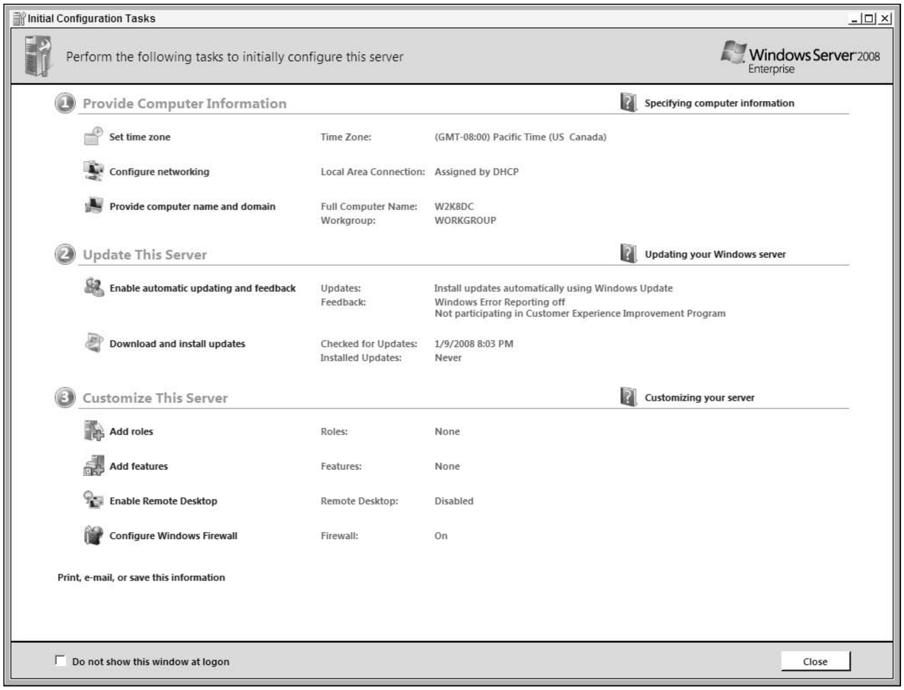
Setting up your first domain controller on a Windows Server 2008 server actually isn't all that complicated, especially if you've done your planning. (You did do your planning ahead of time, right?). In the following sections, I assume that you've already installed the OS successfully on the server. I show you the steps to take afterward to configure the server as the first domain controller in the AD forest. I cover both the fully attended process of setting up the domain controller and the silent, unattended domain controller creation. But, before I do that, here's a look at two tools that make your job easier.

Initial Configuration Tasks Wizard and the Server Manager console

With Windows Server 2008, Microsoft continues its great tradition of providing wizard applications to assist with software installation and configuration tasks. When you log in to a Windows Server 2008 server for the first time, you're presented with the Initial Configuration Tasks Wizard (see Figure 7-2).

From this wizard you can configure the following:

- **Set Time Zone:** It's critical that you change the time zone to the correct one for where this server is. Domain controllers are required to be in time synchronization with each other so that Kerberos works properly.



Initial Configuration Tasks

Perform the following tasks to initially configure this server

Windows Server 2008 Enterprise

1 Provide Computer Information — Specifying computer information

Set time zone	Time Zone:	(GMT-08:00) Pacific Time (US Canada)
Configure networking	Local Area Connection:	Assigned by DHCP
Provide computer name and domain	Full Computer Name: Workgroup:	W2K8DC WORKGROUP

2 Update This Server — Updating your Windows server

Enable automatic updating and feedback	Updates: Feedback:	Install updates automatically using Windows Update Windows Error Reporting off Not participating in Customer Experience Improvement Program
Download and install updates	Checked for Updates: Installed Updates:	1/9/2008 8:03 PM Never

3 Customize This Server — Customizing your server

Add roles	Roles:	None
Add features	Features:	None
Enable Remote Desktop	Remote Desktop:	Disabled
Configure Windows Firewall	Firewall:	On

Print, e-mail, or save this information

□ Do not show this window at logon

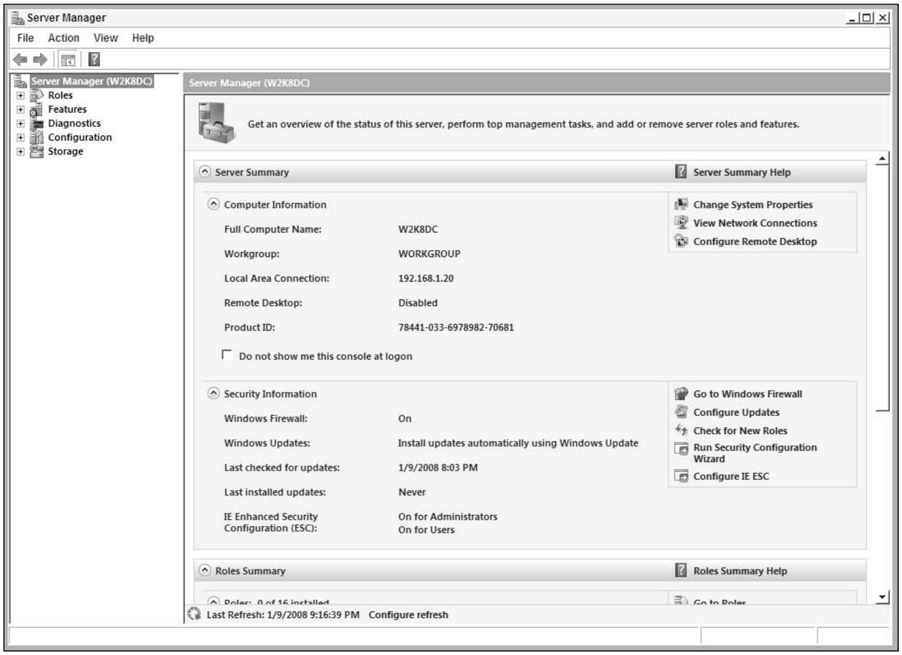
Figure 7-2: The Initial Configuration Tasks Wizard.

- **Configure Networking:** As shown in Figure 7-2, the server is initially configured to get its TCP/IP address from DHCP. You don't want to use DHCP for domain controllers because there's no guarantee that the IP address of the domain controller will have a nonchanging, or *static,* address. This configuration can be a problem particularly when the domain controller is to be a DNS server as well. So make sure you change this setting so that you assign the server a static IP address.
- **Provide Computer Name and Domain:** If you didn't already specify the name that you want the domain controller to have, make sure that you do it before you turn this server into a domain controller. Although you can rename a domain controller, it's better if you can avoid having to do so. Again, if you've done your planning, this shouldn't be an issue. You don't have to worry about the domain; this will be configured when you turn this server into a DC.
- **Enable Automatic Updating and Feedback:** This feature is a more subjective area. On one hand, you need to make sure that you're keeping your DC as secure as possible, and one way to do that is by making sure you're up-to-date on your patches. On the other hand, you might want to be more conservative in this area by not allowing automatic updates and patching the server only when you initiate it.
- **Download and Install Updates:** Selecting this item kicks off the Windows Update service that evaluates your server and gets any available

software updates. Before you turn this server into a DC, it is probably a good idea to make sure that you've applied all the available patches and updates.

- **Add Roles:** From here, you can select and install the various available server roles onto the server, including Active Directory Domain Services. But, I actually do this from Server Manager, which I show you next.
- **Add Features:** Features are different from roles in that they're secondary functions that support the roles. To deploy a DC, you typically won't have to worry about installing any of the features.
- **Enable Remote Desktop:** Selecting this customization allows you to configure the server to allow remote users to connect to this domain controller via Terminal Services. Use this only when you can't always be at the server console to administrate the server.
- **Configure Windows Firewall:** You typically won't have to worry about configuring the firewall service when creating a domain controller. You want to leave this service on for security reasons, too.

After you've done your initial configuration and any required reboots, you're ready to create your domain controller. New for Windows Server 2008 is an all-in-one tool called Server Manager. (See Figure 7-3.) This console was designed as a one-stop place for administrating the server. From this tool, you can install and administrate any of the roles and features on the server as well as manage the server.



Get an overview of the status of this server, perform top management tasks, and add or remove server roles and features.

Computer Information

Full Computer Name:	W2K8DC
Workgroup:	WORKGROUP
Local Area Connection:	192.168.1.20
Remote Desktop:	Disabled
Product ID:	78441-033-6978982-70681

□ Do not show me this console at logon

Security Information

Windows Firewall:	On
Windows Updates:	Install updates automatically using Windows Update
Last checked for updates:	1/9/2008 8:03 PM
Last installed updates:	Never
IE Enhanced Security Configuration (ESC):	On for Administrators On for Users

Roles: 0 of 16 installed

Last Refresh: 1/9/2008 9:16:39 PM

Figure 7-3: The Server Manager tool.

Attended domain controller installation

Here's a look at how to perform a manual, nonautomated (that is, *attended*) installation of a domain controller. (This process is also referred to as a *promotion.*) First, you must install the AD DS server role onto the server. This process is done via Server Manager, as follows:

1. **From the Start Menu in the Administrative Tools group, select Server Manager.**
2. **Select the Roles container and then click the Add Roles link.**

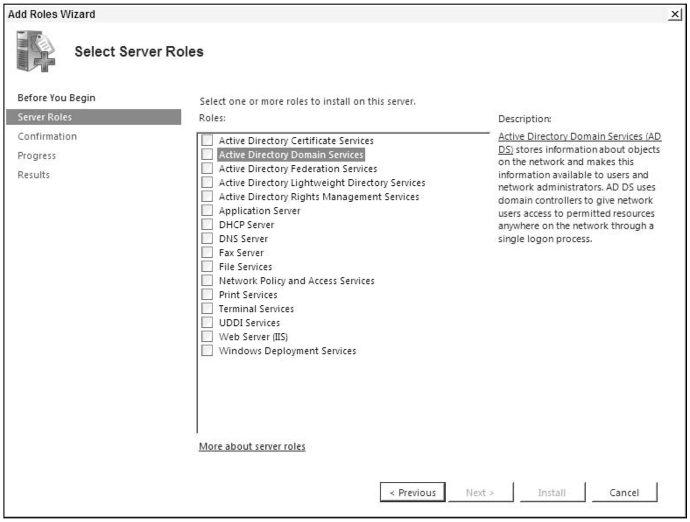
   This opens the Add Roles Wizard.
3. **Click Next.**

   You're presented with the list of available server roles (see Figure 7-4).
4. **Select the Active Directory Domain Services role and click Next.**

   You're presented with some information and links to the Help files on AD DS.
5. **Click Next.**

   TIP

   Don't make the mistake of underestimating or simply not using the information in the Help files. Microsoft has really done a great job with the content in Help, which also contains links to the Microsoft TechNet Web site that contains the latest available technical information.
6. **Select Install to install the server role. Click Close when the role installation is complete.**



Select one or more roles to install on this server.

Roles:

- ☐ Active Directory Certificate Services
- ☐ Active Directory Domain Services
- ☐ Active Directory Federation Services
- ☐ Active Directory Lightweight Directory Services
- ☐ Active Directory Rights Management Services
- ☐ Application Server
- ☐ DHCP Server
- ☐ DNS Server
- ☐ Fax Server
- ☐ File Services
- ☐ Network Policy and Access Services
- ☐ Print Services
- ☐ Terminal Services
- ☐ UDDI Services
- ☐ Web Server (IIS)
- ☐ Windows Deployment Services

More about server roles

Description:

Active Directory Domain Services (AD DS) stores information about objects on the network and makes this information available to users and network administrators. AD DS uses domain controllers to give network users access to permitted resources anywhere on the network through a single logon process.

Figure 7-4: The list of available server roles in the Add Roles Wizard.

At this point, you might think that you've created your domain controller, but all you've done is added the AD DS role to the server, which only installs three of the consoles for administrating Active Directory. Now that you've done this, you're ready to create the domain controller. The following steps run you through the installation of the first domain controller in a new AD forest:

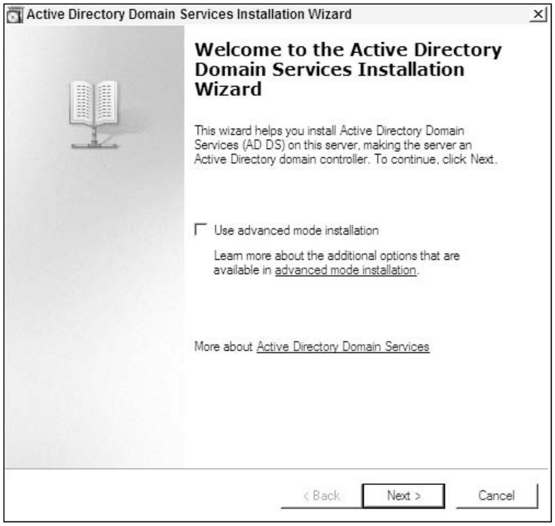
1. **Run the DCPROMO command to invoke the Active Directory Domain Services Installation Wizard. (See Figure 7-5.)**

   You can run this a couple of ways:

   - Run DCPROMO from the Server Manager by selecting the Active Directory Domain Services role and then scrolling in the right pane to the DCPROMO link and selecting it.
   - From the Start menu, select Run, type **DCPROMO**, and then click OK.

2. **For this example, you should run the wizard in advanced mode. So select the Use Advanced Mode Installation option and click Next.**

   An advisement about security compatibilities between Windows Server 2008 and Windows NT appears, saying that Windows Server 2008 by default doesn't enable secure channel encryption that's compatible with how Windows NT works. Because you're installing a new forest, you don't have to worry about this, but pay attention to this warning when you're adding a new DC into an existing forest that includes Windows NT servers.



Active Directory Domain Services Installation Wizard

Welcome to the Active Directory Domain Services Installation Wizard

This wizard helps you install Active Directory Domain Services (AD DS) on this server, making the server an Active Directory domain controller. To continue, click Next.

☐ Use advanced mode installation

Learn more about the additional options that are available in advanced mode installation.

More about Active Directory Domain Services

Figure 7-5: The Active Directory Domain Services Installation wizard.

3. **Click Next.**

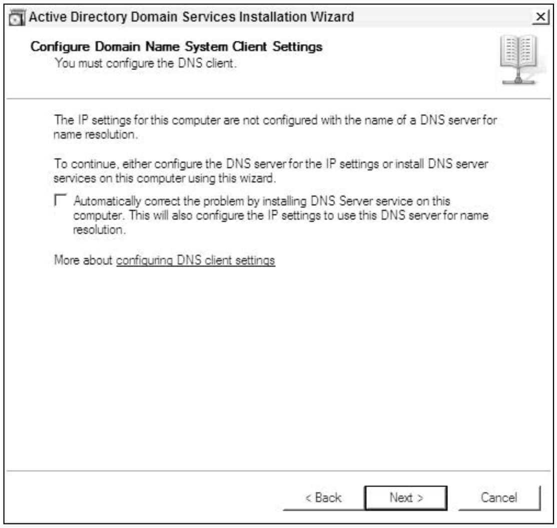
   If you haven't configured this server to use a DNS server (I recommend this configuration), the next screen in the wizard warns you of this. It recommends that you either specify a DNS server IP address on the server's network adapter configuration or have the wizard automatically install DNS when the server is promoted to a domain controller. The latter is my recommendation so that you can take advantage of AD-integrated zones, secure updates, and all the other wonderful DNS features that I discuss in Chapter 4.

4. **Select the check box enabling the DNS software installation and click Next. (See Figure 7-6.)**

   The next couple of steps in the wizard inquire about the forest, tree, and domain for which this server is going to be a domain controller. In this case, you're installing the first DC in a new forest, which means you're also creating a new domain at the same time. Because this is going to be the first domain in the forest, this domain is the forest root domain.

5. **Select the Create a New Domain in a New Forest option and click Next. (See Figure 7-7.)**

   You 're prompted for the Fully Qualified Domain Name (FQDN) for the new domain. If you've done your planning, you should already know what the name of this domain will be in DNS. ***Note:*** If you aren't installing DNS along with the domain controller promotion, the DNS server that the server is using must be able to resolve the FQDN that you enter here.



Active Directory Domain Services Installation Wizard

Configure Domain Name System Client Settings

You must configure the DNS client.

The IP settings for this computer are not configured with the name of a DNS server for name resolution.

To continue, either configure the DNS server for the IP settings or install DNS server services on this computer using this wizard.

□ Automatically correct the problem by installing DNS Server service on this computer. This will also configure the IP settings to use this DNS server for name resolution.

More about configuring DNS client settings

Figure 7-6: The prompt to install DNS in the AD DS Installation Wizard.

6. **Type the name of the domain and click Next.**

   A check is performed to verify that an existing AD domain with this name doesn't already exist.

   Next, you're prompted for the NetBIOS name that will be used in this domain. Again, if you've done your planning, you should have this available.

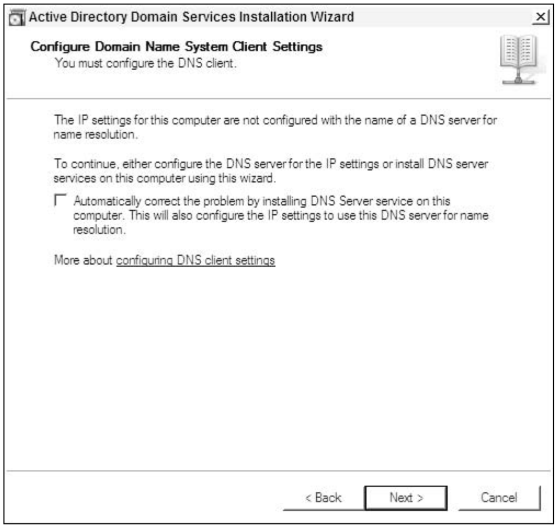
7. **Type the name and click Next.**

   Another check is made to determine whether this name is already in use.

   In the next screen, provide the forest functional level for your new forest: Windows 2000, Windows 2003, or Windows 2008. If you need to have Windows 2000 or 2003 domain controllers in the forest, then you can select the corresponding functional level. But if you're using only Windows Server 2008 and want to take advantage of all the new AD DS features available in 2008, you should select the Windows Server 2008 forest functional level. Doing this defaults the domain functional level to Windows Server 2008 as well.

8. **Select the forest functional level for your new forest. (See Figure 7-8.)**

   Next, you're prompted for some additional options for enabling DNS (which you're already doing): Whether the new domain controller should be a global catalog server (which it should because it's the first DC in the forest) and whether the DC should be an RODC (which it can't be because it's the first DC in the forest).



Configure Domain Name System Client Settings
You must configure the DNS client.

The IP settings for this computer are not configured with the name of a DNS server for name resolution.

To continue, either configure the DNS server for the IP settings or install DNS server services on this computer using this wizard.

□ Automatically correct the problem by installing DNS Server service on this computer. This will also configure the IP settings to use this DNS server for name resolution.

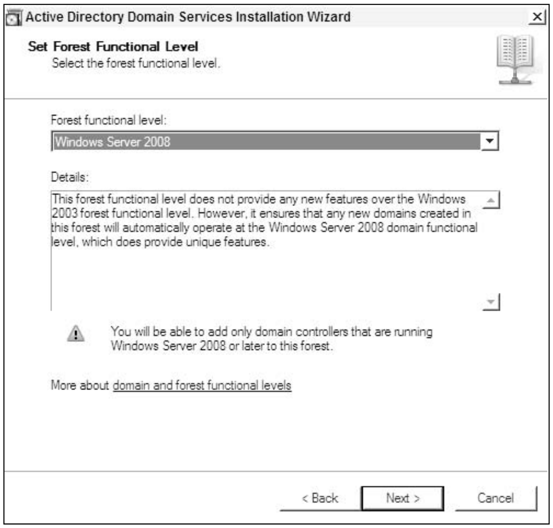
More about configuring DNS client settings

Figure 7-7: Selecting the Create a New Domain in a Forest option.

Forest and domain functional levels

While the Windows Server operating system has evolved from Windows 2000 Server to Windows Server 2008, new features and abilities have been added to Active Directory. Because many of these features require that all the DCs in a forest or domain be running the same version of the OS, it's necessary to have a way of restricting these new abilities until admins can get all the DCs running on the same OS. Within an AD forest, there's a forest functional level setting, and within each domain of the forest, there's a domain functional level setting. Each of these settings can be set to either *mixed mode* (the DCs are running different OSs in the domain or forest) or *native mode* (the DCs are all running the same OS). The forest/domain functional level also includes the OS version. For example, you could have a forest or domain that's in a Windows Server 2008 mixed mode; that is, DCs in the forest are running Windows Server 2008 as well as Windows Server 2003 or Windows 2000 Server. Until you're running in native mode (in both the forest and domains in the forest) on a particular OS, you typically can't take advantage of all the new AD features for that particular server OS release. For example, in the RODC discussion in Chapter 6, I say you can't have an RODC in a forest unless it's running in (at least) a Windows Server 2003 native mode; but unless you're in 2008 native mode, you can't take advantage of all the features of an RODC.

9. **Click Next. You're prompted that the DNS zone for the new forest doesn't exist because you're going to create a new DNS server. This is the expected response. Just click OK to continue.**



Active Directory Domain Services Installation Wizard

Set Forest Functional Level

Select the forest functional level.

Forest functional level:

Windows Server 2008

Details:

This forest functional level does not provide any new features over the Windows 2003 forest functional level. However, it ensures that any new domains created in this forest will automatically operate at the Windows Server 2008 domain functional level, which does provide unique features.

You will be able to add only domain controllers that are running Windows Server 2008 or later to this forest.

More about domain and forest functional levels

Figure 7-8: Selecting the forest functional level.

10. **You're asked where you want to store the directory service files and the SYSVOL directory on the domain controller. (See Figure 7-9.) This should be one of the items you planned upfront (I discuss DC file locations in Chapter 3). Type the correct locations and click Next.**

11. **Next, type a password to be used on this DC in directory service restore mode that is used in disaster recovery (I cover this mode in Chapter 16). Type the password into both fields and click Next.**

    Almost done! The wizard displays a summary of the options you've provided. You're also presented with an option to export the settings you've specified to an answer file (I talk about this in a second).

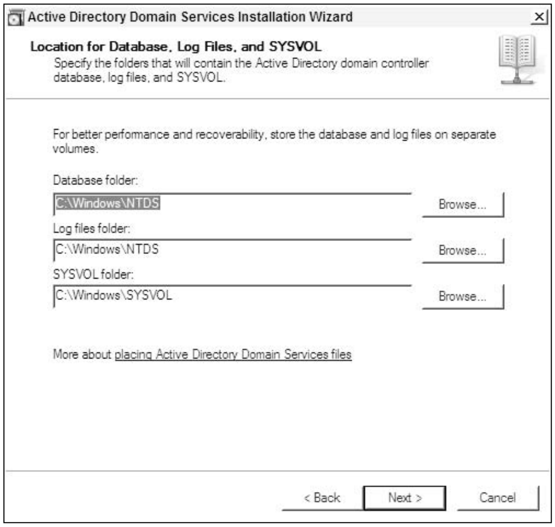
12. **Review the settings you've provided. If they're okay, click Next.**

    The wizard performs the required changes and software installation to turn your server into a domain controller. When the wizard finishes, you're prompted to reboot the server to complete the installation.

Unattended domain controller installation

Although it's great to have an easy to use wizard to promote your domain controllers, sometimes it's better not to have to answer all of its questions every time you create a new DC. Why?

- You might have to create a lot of new DCs in remote locations (where you cannot run the DCPROMO Wizard yourself) or over a remote desktop connection.



Active Directory Domain Services Installation Wizard

Location for Database, Log Files, and SYSVOL

Specify the folders that will contain the Active Directory domain controller database, log files, and SYSVOL.

For better performance and recoverability, store the database and log files on separate volumes.

Database folder: C:\Windows\NTDS

Log files folder: C:\Windows\NTDS

SYSVOL folder: C:\Windows\SYSVOL

More about placing Active Directory Domain Services files

Figure 7-9: Specifying where to put the directory service files and SYSVOL.

- You want to prevent any errors in the answers to the questions in the DCPROMO Wizard.
- Simply to save time in running the wizard.

Fortunately, you can create a text file that provides all the answers to the DCPROMO Wizard in advance. This file is an *answer file.* You can tell the DCPROMO program to use an answer file by running DCPROMO as follows:

```
DCPROMO /unattend:<answer file name>
```

Where `<answer file name>` is the filename of the answer file you're using.

You can create this file a couple of ways:

- You can create it by running DCPROMO interactively, as shown in the preceding section, and then when prompted at the end, clicking the Export Answers button to create the answer file. Keep in mind that you can perform these steps without actually promoting a DC by just running DCPROMO up to the point where you can export the answer file and then just canceling the wizard.

  If you create the answer file from the DCPROMO Wizard, you're still going to need to edit the answer file before using it. The Safe Mode administrator password option (referred to as the `SafeModeAdmin Password` key in the file) will need to be defined. If the key isn't defined, DCPROMO will error out, stating that the supplied password is not sufficient.

- You can simply create the answer file manually by using Notepad. This is a more challenging way to create the file but it allows you more flexibility in controlling how you want the DC promotion process to run.

To create the file with Notepad, follow these steps:

1. **Create a new text file with the Notepad application. Place the following statement at the beginning of the file:**

   ```
   [DCINSTALL]
   ```

2. **Add the necessary key value pairs to the file so that DCPROMO has sufficient information to promote the server successfully to a domain controller.**

   The format of the entries is

   ```
   <key>=<value>
   ```

   For example, if you're specifying the `ForestLevel` key be a value of 3, the entry is

   ```
   ForestLevel=3
   ```

Table 7-3 lists some of the commonly used keys and their values.

Table 7-3: DCPROMO Answer File Keys and Values

Key	*Value*
`ReplicaorNewDomain`	Set this key to `Replica` if this is an additional DC for an existing domain or to `Domain` if this is the first DC in a new domain. If you're installing an RODC, set this key to `ReadOnlyReplica`.
`NewDomain`	Set to `Forest` if creating a new forest, `Tree` if creating a domain in a new tree of the existing forest, and `Domain` if creating a new domain in an existing tree.
`NewDomainDNSName`	The FQDN of the domain the DC is going into.
`ChildName`	The FQDN of the child domain if creating a new child domain.
`ParentDomainDNSName`	The FQDN of the parent domain. Needed when creating a new child domain.
`ForestLevel`	The forest functional level: `0` = Windows 2000, `2` = Windows 2003, `3` = Windows 2008.
`DomainNetBIOSName`	The NetBIOS name of the new domain.
`DomainLevel`	The domain functional level: `0` = Windows 2000, `2` = Windows 2003, `3` = Windows 2008.
`InstallDNS`	Set to `Yes` to install DNS and `No` to not install DNS (default).
`ConfirmGC`	Set to `Yes` if the DC should be a GC; otherwise, `No`.
`CreateDNSDelegation`	`Set to Yes` if you're creating a new delegated DNS domain for the new AD domain along with installing DNS on the server. If delegated domain already exists, you can set this to `No`.
`DatabasePath`	Set to the file path on the server where the directory service database file should be stored.
`LogPath`	Set to the file path on the server where the directory service database log file should be stored.
`SYSVOLPath`	Set to the file path on the server where the SYSVOL directory should be stored.

(continued)

Table 7-3: *(continued)*

Key	*Value*
`SafeModeAdminPassword`	Set to the password you want to use for safe mode on the server.
`UserName`	The name of the administrative account with which DCPROMO should be executed. This account needs to have the correct permissions to perform the action specified in the answer file.
`Password`	The password for the account specified in the `UserName` key.
`UserDomain`	The domain of the account specified in the `UserName` key.
`ReplicationSourcePath`	Set to the directory where the previously created Install From Media (IFM) image is located.
`RebootOnSuccess`	`Yes`, if you want to have the server reboot after DCPROMO finishes completing the promotion process. Default is `No`.

This isn't all the available keys, though. For a complete list, go to the Command Reference for Windows Server 2008 at Microsoft's TechNet Web site (`http://technet.microsoft.com`).

Deploying AD DS on a Core Server

Using a Windows Server 2008 Core Server for a domain controller is a great way of providing AD DS because of the increased security and lower maintenance. Unfortunately, setting up that core server as a DC isn't quite as easy as a full server installation setup. The primary reason is that no windowed applications and wizards are available when you set up a core server. Therefore, you have to find out how to use the command line tools to get things configured.

Before you can run DCPROMO, you need to perform a couple of configuration changes to your core server. First, you need to configure a static IP address for the server. This process is done by using the `NETSH` command, as follows:

1. **Identify the ID number that the operating system has assigned to the network adapter you want the server to use on the network. You can do so by executing the following command to get a list of the network adapter IDs.**

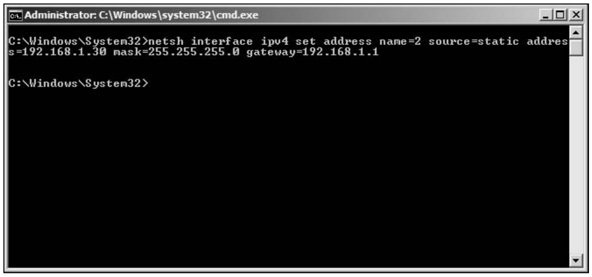
```
Netsh interface ipv4 show interfaces
```

This brings up a list of the network interfaces on the server. The Idx column shows the interface number.

2. **After you have the correct interface number, type the following command:**

```
Netsh interface ipv4 set address name="<Idx>"
        source=static address=<static IP>
        mask=<subnet mask> gateway=<gateway IP>
```

`<Idx>` is the network adapter interface number, `<static IP>` is the IP address you are assigning, `<subnet mask>` is the IP subnet mask for the network the server is on, and `<gateway IP>` is the IP address of the network gateway. See Figure 7-10.



```
C:\Windows\System32>netsh interface ipv4 set address name=2 source=static addres
s=192.168.1.30 mask=255.255.255.0 gateway=192.168.1.1


C:\Windows\System32>
```

Figure 7-10: Configuring a static IP address by using NETSH.

3. **Configure the IP address of the DNS server that this server should be using. This is done with NETSH as well, with the following command:**

```
Netsh interface ipv4 add dnsserver name="<Idx>"
        address=<DNS IP> index=1
```

`<Idx>` is the network adapter interface number and `<DNS IP>` is the IP address of the DNS server.

Although my examples assume that you're using IPv4, the `NETSH` command supports IPv6 as well.

Now you're ready to run DCPROMO. The downside here is that because you're using a core server, you can't run DCPROMO in interactive mode, answer the questions in the AD DS Installation Wizard, and promote the DC. However, you can run DCPROMO in the unattended mode. as I just covered. to promote the DC. The other alternative is to provide some command switches along with the `DCPROMO` command to control how the DC promotion is done. The same keys and values listed in Table 7-3 work as command switches as well, as shown below:

```
DCPROMO /<key>=<value> /<key>=<value> /<key>=<value> ...
```

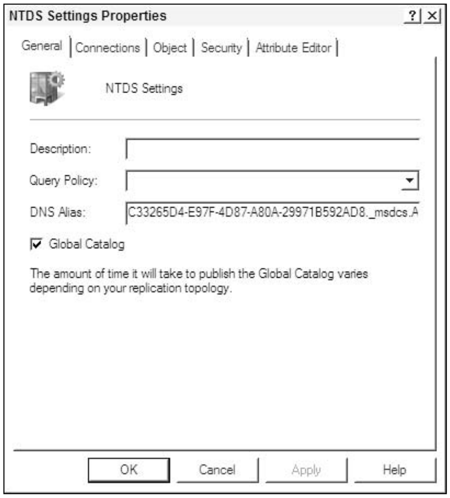
Note: A forward slash precedes each key and value set.

After the install

Now that you've installed your domain controller or domain controllers, you're done with the installation of AD DS and you don't have to do anything else, right? Wrong. You're just getting started. I have a list here of some of the other tasks that you need to tackle before you can truly say that your Active Directory is ready for prime time:

- **Sites and subnet creation:** In Chapter 12, I cover how to create the sites and subnets that you've already planned. At this point in your AD DS install, you can create those objects. Then you will need to move the domain controllers to the appropriate site. I strongly recommend that you perform the sites and subnet creation after you deploy the first DC in the forest. The advantage in doing this first is that each time you promote a new DC, the DC will be placed into the correct site automatically based on the DC's IP address. This can save you a lot of time and heartache.
- **Organizational Unit creation:** After you create each of the domains in your forest, you can go back to each domain and build out the OU structure as determined by the planning you did. (If you didn't do your planning, I hope you're starting to feel guilty now.) After the OU structure is complete, you can set up any necessary permissions delegation (see Chapter 14) and any group policies that are assigned to the OUs.
- **Global catalog creation:** Although you can designate your DC a global catalog server while running DCPROMO, you can also go back and tell a domain controller to become a GC. This is done in the AD Sites and Services console under the NTDS Settings properties of each domain controller. (See Figure 7-11.)
- **Operations master assignments:** After you create more than one domain controller in each domain of the forest, you can start thinking about where to assign the operations masters. Refer to Chapter 6 for recommendations on where to place these roles.
- **Group Policy creation and application:** I cover group policies in Chapter 14. At minimum, you want to edit the security policy portion of the Default Domain Policy on each domain in your forest so that you enable your established policy for passwords and account lockouts.

- **Universal group caching enablement:** If you have an Active Directory site where you're not placing a global catalog server, you should consider enabling universal group caching on the site. In the AD Sites and Services console, select the site on which you want to enable caching and then pull up the Properties panel of the site's Site Settings object (see Figure 7-12). After selecting the check box to enable universal group caching, specify the site that the domain controllers in the caching site should pull the universal group memberships from. Typically, this is the closest site that has a global catalog in it.



NTDS Settings

Description:

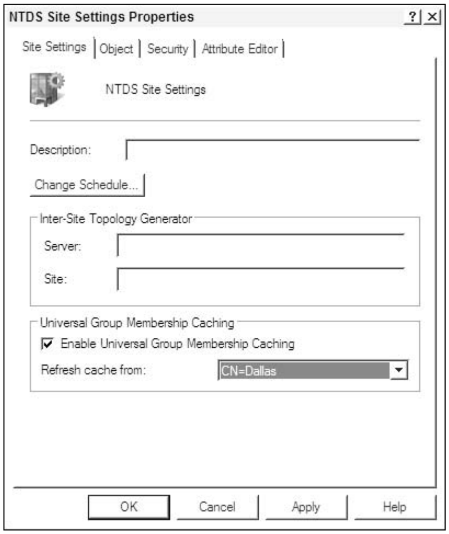
Query Policy:

DNS Alias: C33265D4-E97F-4D87-A80A-29971B592AD8._msdcs.A

☑ Global Catalog

The amount of time it will take to publish the Global Catalog varies depending on your replication topology.

Figure 7-11: Enabling a domain controller to be a global catalog server.



NTDS Site Settings

Description:

Inter-Site Topology Generator

Server:

Site:

Universal Group Membership Caching

☑ Enable Universal Group Membership Caching

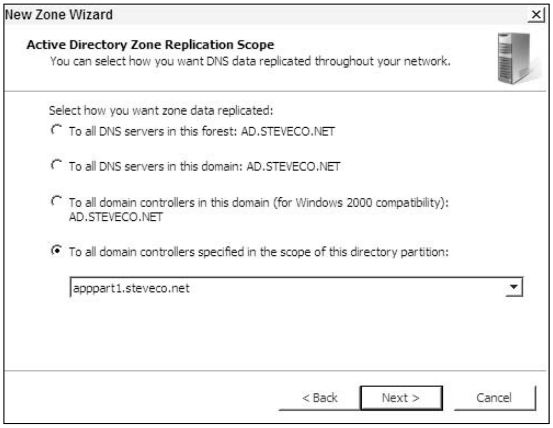
Refresh cache from: CN=Dallas

Figure 7-12: Enabling universal group caching on a site.

- **DNS application partition creation:** Depending on your DNS design, you might have a need to create an application partition in Active Directory to host a particular DNS zone. The first step is to create the application partition for DNS to use. Perform this step by using the DNSCMD utility from a command prompt window as follows:

  ```
  DNSCMD <servername> /createdirectorypartition <FQDN>
  ```

  `<servername>` is the DNS server the application partition is being created on, and `<FQDN>` is the name of the partition. After you create the partition, you can then create a new DNS zone. When you're prompted for the replication scope of the zone in the DNS console (see Figure 7-13), specify that the zone be stored in the application zone you have created.



New Zone Wizard

Active Directory Zone Replication Scope

You can select how you want DNS data replicated throughout your network.

Select how you want zone data replicated:

○ To all DNS servers in this forest: AD.STEVECO.NET

○ To all DNS servers in this domain: AD.STEVECO.NET

○ To all domain controllers in this domain (for Windows 2000 compatibility): AD.STEVECO.NET

● To all domain controllers specified in the scope of this directory partition:

apppart1.steveco.net

Figure 7-13: Using an application partition to store a DNS zone.

Miscellaneous Issues

Before I finish this chapter, I want to cover a few miscellaneous topics related to installing AD DS.

Installing AD DS from media

Typically, each time you add a new domain controller to a forest, the DC has to replicate with one or more existing domain controllers in the forest to obtain a copy of all the existing directory information. In a large directory environment, this replication can take a while to complete, creating a negative performance impact on the network and other domain controllers.

But, of course, there is another alternative. You can perform an installation of the directory data from supplied media (that is, from data copied from another domain controller) at the time that you promote the server to be a domain controller. The first step in this process is to create the media. This is done by using the `NTDSUTIL` command. Just follow these steps:

1. **On an existing domain controller that's in the same domain as the new DC you're about to bring up, open a command prompt window.**
2. **Run the NTDSUTIL utility by typing** NTDSUTIL **and pressing Enter.**
3. **Specify that you want to work with the NTDS instance by typing** ACTIVATE INSTANCE NTDS **at the NTDSUTIL prompt and pressing Enter.**

   You can have other instances of Active Directory on this server if you're running AD LDS, which I cover in Chapter 8.
4. **To enter the Install from Media (IFM) mode of NTDSUTIL, type** IFM **and press Enter.**
5. **At this point, you can create one of several types of media.**

   The media you create is dependent on whether you're going to use the media to deploy a normal domain controller or an RODC. Also, you have the option of including or not including a copy of the SYSVOL. Table 7-4 lists the four different media types and the command that you use to do so.

   Note: `<dir>` is the local directory on the DC where the copy of the media will be placed.
6. **After you type the appropriate command and press Enter, a snapshot backup of the data is created and placed into the directory you specify. After this process is complete, quit the NTDSUTIL utility.**

Table 7-4: Available IFM Installation Media Types

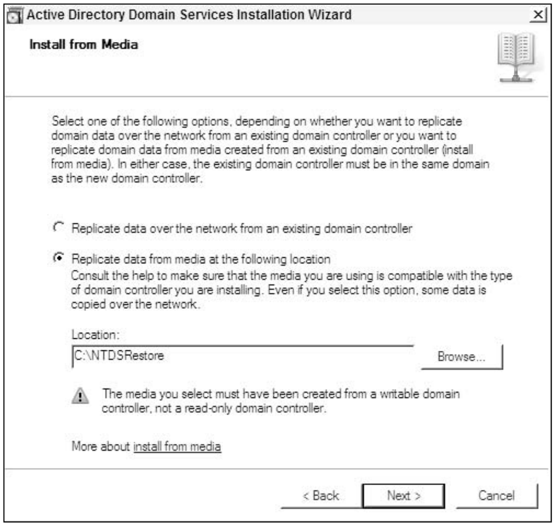
Type of Media	*Description*	*Command*
Full	This creates a full copy of the directory instance but no SYSVOL.	`Create Full` *<dir>*
RODC	This creates a copy of the directory designed for a read-only DC.	`Create RODC` *<dir>*

(continued)

Table 7-4: *(continued)*

Type of Media	*Description*	*Command*
RODC with SYSVOL	This creates a RODC copy of the directory with the SYSVOL directory.	`Create SYSVOL RODC` *<dir>*

Now you can install the new domain controller. You can use the media with an interactive DC promotion by using the Active Directory Domain Services Installation Wizard or you can specify the IFM media directory with the `ReplicationSourcePath` key in the answer file for an unattended installation. (See Figure 7-14.)



Install from Media

Select one of the following options, depending on whether you want to replicate domain data over the network from an existing domain controller or you want to replicate domain data from media created from an existing domain controller (install from media). In either case, the existing domain controller must be in the same domain as the new domain controller.

○ Replicate data over the network from an existing domain controller

● Replicate data from media at the following location
Consult the help to make sure that the media you are using is compatible with the type of domain controller you are installing. Even if you select this option, some data is copied over the network.

Location: C:\NTDSRestore

The media you select must have been created from a writable domain controller, not a read-only domain controller.

More about install from media

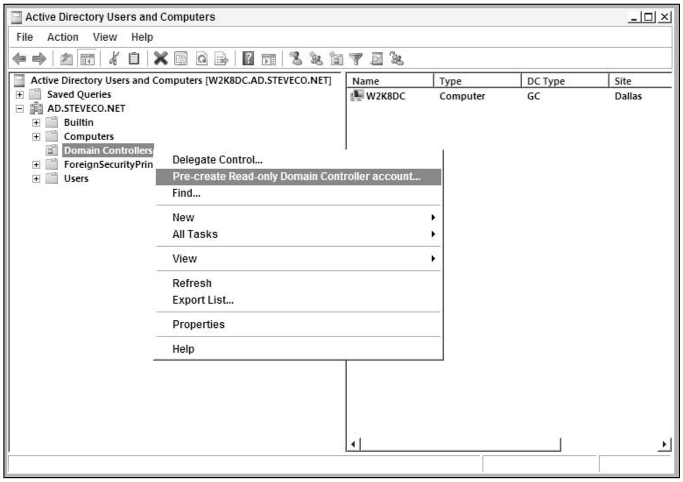
Figure 7-14: The Install from Media screen of the AD DS Installation Wizard.

Deploying an RODC

Installing a read-only domain controller isn't difficult. If you're promoting the RODC by using the AD DS Installation Wizard, the wizard will prompt you to specify this. On the other hand, if you're running an unattended promotion, you can specify the creation of an RODC by including the `ReadOnlyReplica` value for the `ReplicaorNewDomain` key in the answer file. However, you should be aware of the following items if you're going to deploy an RODC:

- If you're adding an RODC to an existing forest that was constructed by using Windows Server 2003, that forest will need to be at the Windows Server 2003 forest/domain native mode. Also, you will need to run the `ADPREP /RODCPREP` command against the forest before you can install an RODC.
- The first Windows Server 2008 domain controller in an existing Windows 2003 domain cannot be an RODC. In other words, there must be at least one writable Windows Server 2008 DC in a domain before you can create your first RODC in that domain.
- You can also install an RODC through a process of first staging the domain controller account and then installing the RODC later. Staging provides a way for a non-domain administrator to perform an RODC install, which is particularly useful when you're deploying an RODC in a remote location like a branch office where no AD administrators are present. The first step is to create the RODC account. ***Note:*** A domain administrator must do this part. You do this in the AD Users and Computers console by right-clicking the Domain Controllers OU and selecting the Pre-Create Read-Only Domain Controller Account option (see Figure 7-15). The AD DS Installation Wizard opens in a special mode for creating just the RODC computer account. When you're ready to promote the server at the branch office to be an RODC, run DCPROMO by using the `/UseExistingAccount:Attach` command switch.

As you can when creating a normal DC, you can use an answer file in staging an RODC as well. This might be preferable when a non-administrator is running the RODC promotion.



Name	Type	DC Type	Site
W2K8DC	Computer	GC	Dallas

Figure 7-15: Selecting the Pre-Create Domain Controller account option.

Part III

New Active Directory Features

The 5th Wave By Rich Tennant

"Sure, at first it sounded great – an intuitive network adapter that helps people write memos by finishing their thoughts for them."

In this part . . .

Active Directory includes a number of new components in Windows Server 2008. These components expand upon the directory services that AD DS provides to create a more complete solution for identity and access management. If your needs include providing single sign-on access to Web applications, controlling how users use documents, or simply providing a directory service without all the bells and whistles, then definitely take a look at the subjects covered in this section.

Chapter 8

AD LDS: Active Directory on a Diet

In This Chapter

- Scenarios for AD LDS usage
- AD LDS capabilities
- AD LDS Authentication and replication
- Deploying AD LDS

Sometimes, too much of a good thing isn't good at all. That's why enjoying food a little too much causes folks to go on diets. Active Directory Domain Services is sometimes guilty of being too much of a good thing as well. That's why Microsoft has created as part of the Windows Server 2008 Active Directory umbrella a standalone server application called Active Directory Lightweight Directory Services or AD LDS. In this chapter, I take a brief look at AD LDS, including in what ways AD LDS is useful, how it works, and what it's capable of providing.

The Need for a Lighter AD

AD Lightweight Directory Service is simply that: an LDAP-based directory service application that's "lighter" than Active Directory Domain Services because it doesn't include all the features that are in AD DS. So why would anyone want to use a directory services app that doesn't have all the features of AD DS? As I describe in Chapter 1, a directory service is really just a hierarchical information store to which data can be written and read. There are times when a directory service without a lot of extraneous functionality is exactly what you need, particularly when you need to make a directory service available on the Internet. So what isn't in AD LDS?

- **Kerberos authentication:** Although AD LDS provides LDAP-based authentication, authentication via Kerberos isn't a part of AD LDS. The reason is that Kerberos-based authentication is unnecessary in many of the scenarios in which you could use AD LDS.

- **Forests, Domains, DCs, and GCs:** Forget everything you've read so far about forests and domains: They simply don't exist in AD LDS.
- **Dependence on DNS:** AD DS dependence on DNS as a locator service doesn't exist in AD LDS. As such, there really is no dependence on SRV records and a DNS infrastructure when you deploy AD LDS.
- **Sites, Site Links, Subnets:** AD LDS does not utilize any concept of a site topology nor does it perform replication in the same manner as AD DS. Therefore, the need to define sites, site links, or subnets is not required to set up an AD LDS server.
- **Group Policies:** Although I haven't discussed group policies yet (see Chapter 14), know that AD LDS doesn't support them. Group policies can be used for enforcing security on users and computers as well as controlling a user's desktop experience and even for distributing software to a computer. Group policies are strictly a Microsoft technology for working with Microsoft-based computers and aren't required in a more generic directory service, such as AD LDS.

TIP

Active Directory Lightweight Directory Service isn't really a new application. This application was previously available in Windows Server 2003 but was known as Active Directory Application Mode, or ADAM.

So when would you want to use AD LDS rather than the full featured AD DS? The following sections provide some examples.

Microsoft Identity Lifecycle Manager

If an organization is storing user information in multiple directories, the management and synchronization of these user identities can be a complicated subject. Additionally, when organizations have users in multiple directories, the process of creating, modifying, and deleting these identities (also known as *provisioning*) is also more difficult. Microsoft has a product called Identity Lifecycle Manager (ILM) that is positioned to provide a solution for this complexity. Previously known as Microsoft Identity Integration Server (MIIS), ILM can provide both directory synchronization and identity provisioning services, as well as management of user-associated certificates (such as for a smart card). ***Note:*** If you need to synchronize multiple AD DS or AD LDS instances, you can still use IIFP as a free solution, but if you have other types of directories to work with, as well as AD, and complex provisioning scenarios, you might consider ILM as a solution.

AD LDS as a phone book

Directory services are a great way of providing information that can be frequently retrieved and searched on in a hierarchical way. In Chapter 1, I use a phone book as an analogy. Well, there's no reason that you can't create a directory service that's actually a phone book. Imagine that you need to make a searchable phone directory of your organization available on the Internet. (See Figure 8-1.) This isn't a difficult task, but it has security repercussions. If you've already deployed AD DS and you have the employees' phone numbers available in that directory, it might not be a good idea to expose your AD DS environment to the Internet for security reasons. Using AD LDS is a great alternative because it can be deployed separately from AD DS and it's designed to simply provide the information retrieval service that you need without the complications involved with Kerberos authentication and group policies.

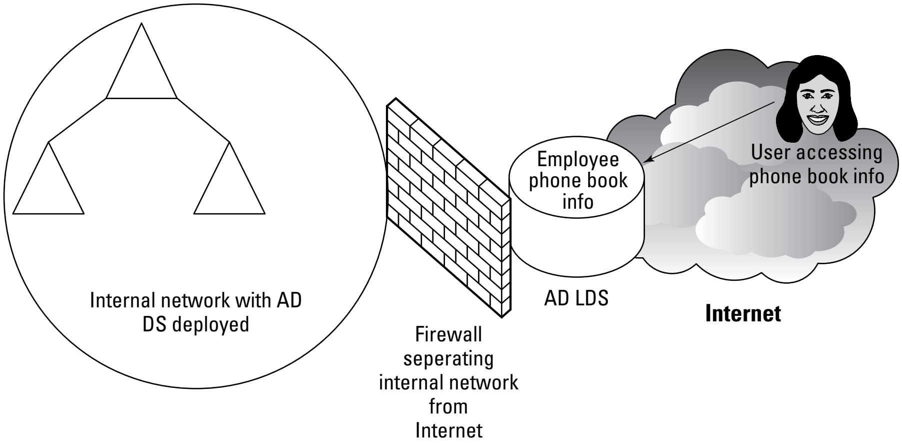

Internal network with AD DS deployed
Firewall seperating internal network from Internet
Employee phone book info
AD LDS
User accessing phone book info
Internet

Figure 8-1: Using AD LDS as a phone book service.

AD LDS as a consolidation store

Say you have multiple directory services established, including multiple AD DS forests, HR systems, and e-mail directories. So that you can create a single point in which information from all these directories is available, you can use AD LDS as a consolidation store. (See Figure 8-2.) Typically AD LDS is more appropriate for a consolidation store because you usually do not need the additional AD DS features like Kerberos authentication or GPO support. This requires a synchronization tool to pull the data from each of these source directories and then collect it into the AD LDS store. Microsoft provides a free tool — the Identity Integration Feature Pack, or IIFP — that can synchronize multiple AD DS and AD LDS instances and provide support for Exchange 2000/2003 directory information. Microsoft also has a more full-featured application called the Identity Lifecycle Manager (ILM) that also includes the functionality that IIFP provides. See the "Microsoft Identity Lifecycle Manager" sidebar for more information.

Microsoft has also included a command line tool called ADAMSYNC.EXE that can be used to perform a manual synchronization. This tool provides a quick way of copying data between AD LDS and AD DS without the additional overhead involved with setting up either IIFP or ILM.

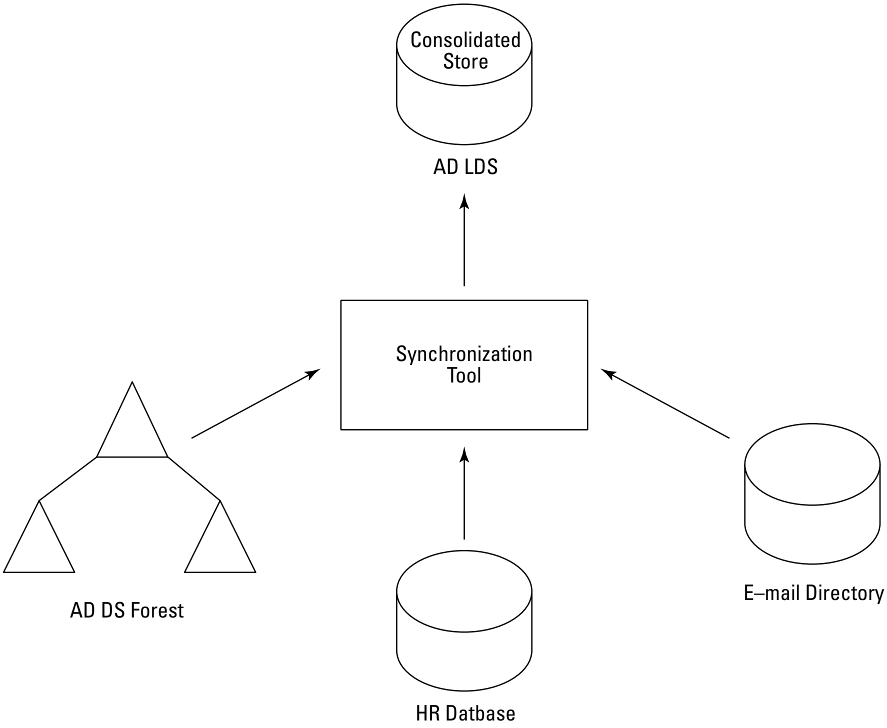


Figure 8-2: AD LDS as a consolidation store.

AD LDS as a Web authentication service

Companies these days are increasingly deploying directory-enabled Web applications in a perimeter network accessible from the Internet that can provide employees and business partners access to business applications and data. To secure these applications and data, you need to provide an authentication service for this solution. AD LDS can provide LDAP-based authentication so that users can authenticate themselves and their identities can be determined by using the information located in the directory store. The identity determines exactly what information and applications the user can access. AD DS can provide LDAP authentication as well but AD LDS is typically a better solution here as the other AD DS features like Kerberos, GPO support are not needed. (See Figure 8-3.)

TIP

I talk about Active Directory Federation Services in the next chapter, but keep in mind that you can use AD LDS as an authentication store for AD FS as well.

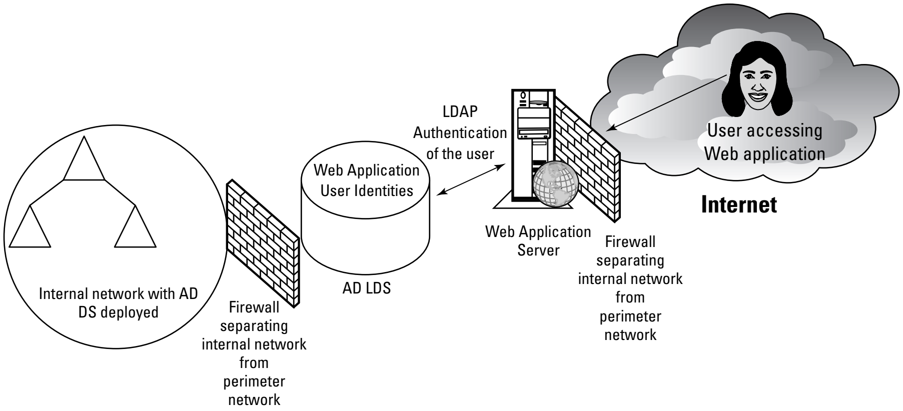

Internal network with AD DS deployed
Firewall separating internal network from perimeter network
Web Application User Identities
AD LDS
LDAP Authentication of the user
Web Application Server
Firewall separating internal network from perimeter network
User accessing Web application
Internet

Figure 8-3: Using AD LDS as a Web authentication service.

Working with AD LDS

Essentially, Active Directory Lightweight Directory Services isn't all that much different from the full-featured AD DS. AD LDS is an X.500-compatible directory service that supports LDAP as an access protocol. AD LDS is even built on the same software code that AD DS uses. After that, though, things get a bit different. A particular instantiation of AD LDS runs in what is called an *instance*. An instance has the following attributes:

- An instance runs as a separate server service (the name of the service is the name of the AD LDS instance). Therefore, you can run multiple instances of AD LDS on the same server and each instance will be independent of each other. (See Figure 8-4.)
- Each instance has at least two partitions on it, as follows:
  - *A schema partition:* The schema partition serves the exact same purpose that the schema partition provides in AD DS. It defines the objects and attributes that can exist within that instance of AD LDS.
  - *A configuration partition:* The configuration partition provides an area to define how replication works with this instance.

- In addition to the configuration and schema partitions, an instance can have zero or more application partitions within it. Each of these application partitions can be used by separate applications, thereby avoiding the need to set up separate instances for each application. It's within these application partitions that the users and groups needed by the application are created.
- So that multiple instances can be supported on a single server, each instance uses a unique set of network port numbers. The default ports for an instance are 389 and 636. The first port is for LDAP, and the second port is for LDAP over an encrypted SSL session (also known as LDAPS). If these port numbers are unavailable, the AD LDS Setup Wizard prompts you with available port numbers beginning at 50000.

TIP

Because each instance runs as a separate service and can use customized port numbers, you can run AD LDS on an AD DS domain controller.

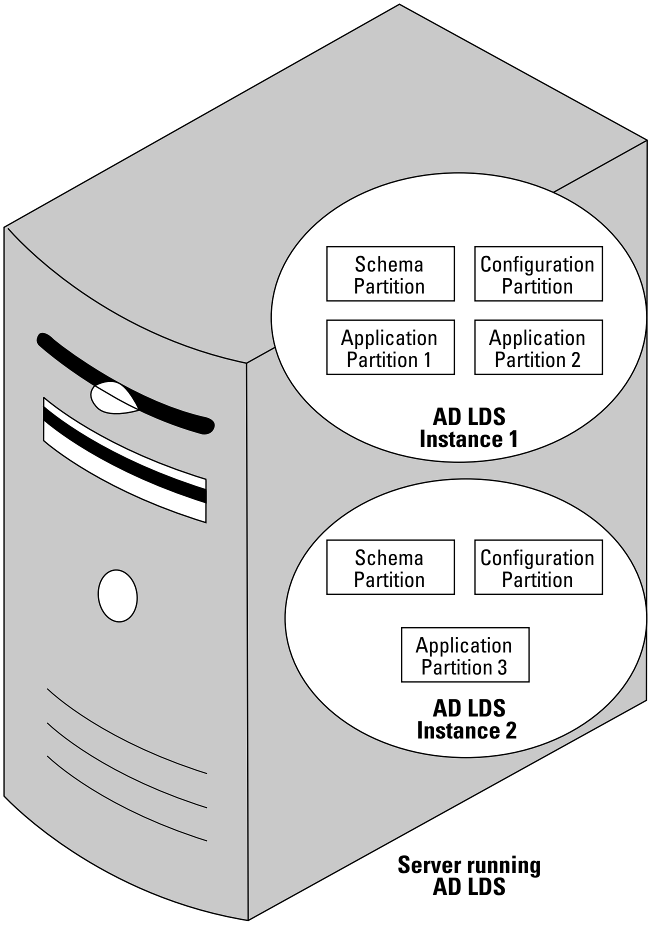


Figure 8-4: AD LDS instances on a server.

Each instance on a server has a separate set of associated files on the server. These files are located in the system root drive under `Program Files\Microsoft ADAM\<`*instancename*`>`. To back up these files you simply need to create a file backup of the files in this directory. To restore the files, simply stop the AD LDS instance service and restore the related files to the same directory.

Because users will want to access your AD LDS instance, you have to manage both users and groups within the instance. Several tools are available to you for managing the contents of the partitions in an AD LDS instance. The ADSIEdit.MSC and `LDP.EXE` tools that come with the Windows Server 2008 OS provide a way to connect to the instance, view each of the partitions, and create, modify, and delete directory objects including users and groups. You can use the `LDIFDE.EXE` command line tool on the instance for importing and exporting data as well.

Security and Replication with AD LDS

As with AD DS, users must authenticate to an AD LDS instance to be able to read the contents of the directory. Typically, this is done through the directory-enabled application that the user is attempting to access. This authentication is normally provided via LDAP authentication and is referred to as *binding* to the directory. When the user has successfully completed the LDAP bind, he can now access directory contents based on what has been authorized to the users.

TIP

AD LDS can integrate with AD DS and forward an authentication request to an existing forest for a domain controller to resolve. This is a *bind redirection*. By doing this, the user logging in to AD LDS also gains access to any resources granted to her AD DS user ID as well as her AD LDS account.

AD LDS also supports the replication of partitions. Like AD DS, the replication is *multimaster;* that is, you can make changes to any one copy of a partition and those changes will be replicated to all other copies. Because AD LDS uses the same base software as AD DS, the replication of data works in a similar manner. (See Chapter 12 for more on AD DS replication.) Each replica of an instance must include both the schema and the configuration partitions. You can then decide whether any application partitions within the instance should be replicated as well. This grouping of replicated instances that share a common schema and configuration partitions is called a *configuration set*. (See Figure 8-5.) With the ability to replicate your directory in a flexible manner, you can load balance the traffic directed to your instances, as well as provide fault tolerance, in case one of the instance replicas is unavailable. You can create new replicas of an instance with the AD LDS Setup Wizard by specifying in the wizard that you want to establish a replica instance. When you do this, you will be prompted to provide the server name and port number for the existing instance from which you want to replicate.

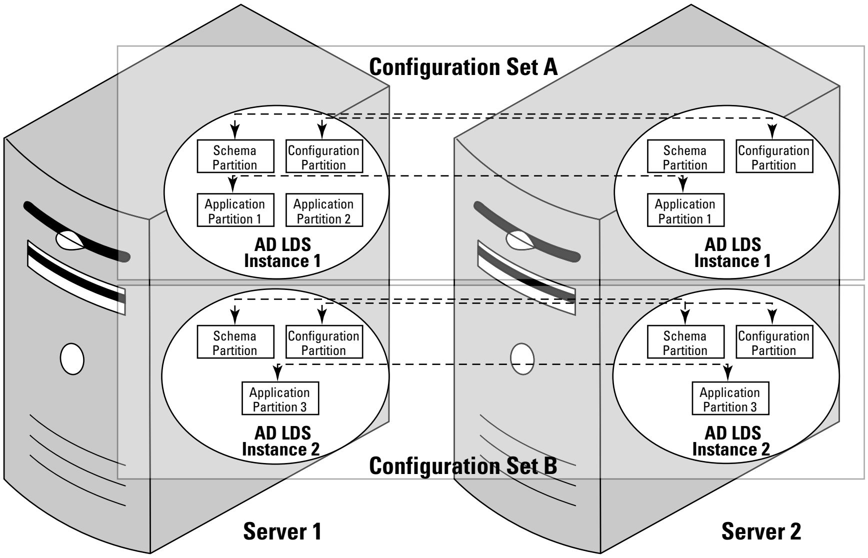


Figure 8-5: Replication of configuration sets in AD LDS.

In Figure 8-5, two servers are running AD LDS, and each server has two instances of AD LDS running. Instance 1 on each server is a member of the same configuration set, A, because the schema and configuration partitions are replicated between the two instances. Also, the administrator has decided to replicate Application Partition 1 between the two instances. Both of the Instance 2s are also members of their own separate configuration set, B, and all partitions (schema, configuration, and Application Partition 3) are replicated between the two instances.

You might also need to perform replication between AD LDS and AD DS. As I mention earlier in the chapter, you can use IIFP, ILM, or `ADAMSYNC.EXE` as a solution.

Deploying AD LDS

To install AD LDS on a server, open the Server Manager tool on any Windows Server 2008 server, click the Install Roles link, and then select Active Directory Lightweight Directory Services. After the AD LDS role has been installed on the server, you can then run the AD LDS Setup Wizard to create either a new instance or a replica of an existing instance.

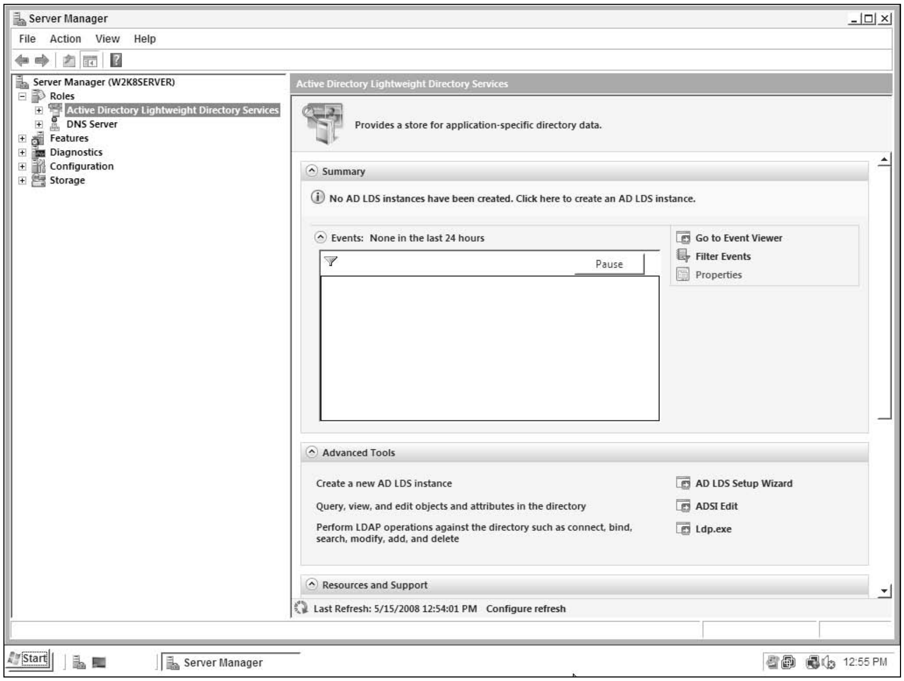
Follow these steps to create a new AD LDS instance on your server:

1. **Open the Server Manager tool from the Administrative Tools group.**
2. **Open the Roles container and select Active Directory Lightweight Directory Services.**

The AD LDS Role Manager opens. (See Figure 8-6.)

3. **From the AD LDS Role Manager, click the Click Here to Create an AD LDS Instance link.**

   The AD LDS Setup Wizard opens. (See Figure 8-7.)


Server Manager
File
Action
View
Help
Server Manager (W2K8SERVER)
Roles
Active Directory Lightweight Directory Services
DNS Server
Features
Diagnostics
Configuration
Storage
Active Directory Lightweight Directory Services

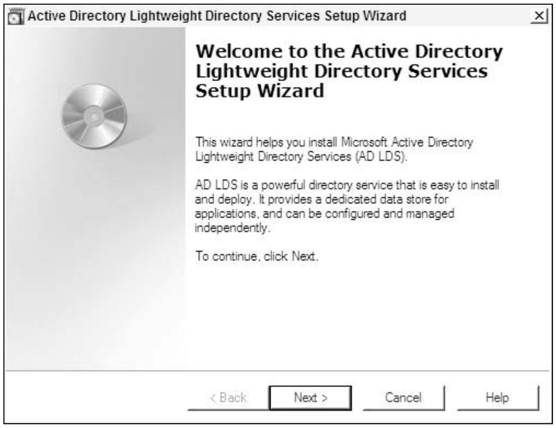
Provides a store for application-specific directory data.

No AD LDS instances have been created. Click here to create an AD LDS instance.

Create a new AD LDS instance	AD LDS Setup Wizard
Query, view, and edit objects and attributes in the directory	ADSI Edit
Perform LDAP operations against the directory such as connect, bind, search, modify, add, and delete	Ldp.exe

Last Refresh: 5/15/2008 12:54:01 PM Configure refresh

Figure 8-6: The Active Directory Lightweight Directory Services Role manager.



Welcome to the Active Directory Lightweight Directory Services Setup Wizard

This wizard helps you install Microsoft Active Directory Lightweight Directory Services (AD LDS).

AD LDS is a powerful directory service that is easy to install and deploy. It provides a dedicated data store for applications, and can be configured and managed independently.

To continue, click Next.

Figure 8-7: The Active Directory Lightweight Directory Services Setup Wizard.

4. **Click Next.**

   You can now select whether you want to create a new unique instance or a replica of an existing instance.

5. **Because you're creating a new instance, select that option and then click Next.**

6. **Type the name for the instance. Remember that this will be the name for the server service on the server that will host the instance. Figure 8-8 shows** LDSInstance1 **for the instance name. Click Next.**

   You're presented with the default LDAP and LDAPS ports that the instance will use. If you're not running this on a domain controller, you get the default ports of 389 and 636.

7. **Click Next.**

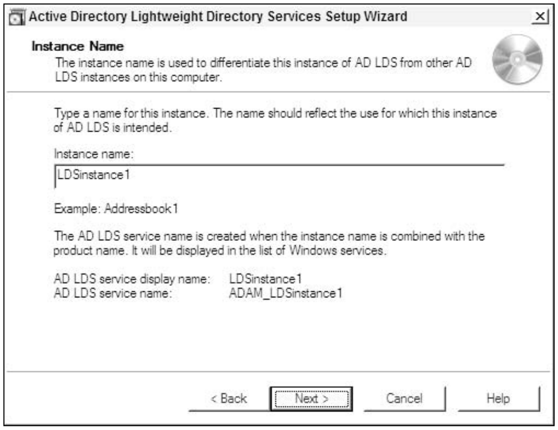
   You now have to decide whether you want to create an application partition for the instance. (If you don't do it now, you can create one later.) ***Note:*** For the naming of the application partition you need to use an LDAP-formatted distinguished name. (For a quick discussion on distinguished names, see Chapter 4.)

8. **In this example, I am using** CN=App1,DC=STEVECO,DC=NET**. (See Figure 8-9.) Click Next.**

   On the next screen, you're provided the default file directory paths in which the files for this instance will be located.

9. **Click Next.**

10. **Next, you must define the account under which the server service for this instance will be run. You need to carefully consider how you answer this question. Note that the recommendation is to not use the Network Service account; rather, create a separate user ID that has no administrative rights on the local server and use that ID for the instance's service account. Click Next.**


Active Directory Lightweight Directory Services Setup Wizard
Instance Name
< Back
Next >
Cancel
Help

The instance name is used to differentiate this instance of AD LDS from other AD LDS instances on this computer.

Type a name for this instance. The name should reflect the use for which this instance of AD LDS is intended.

Instance name:
LDSinstance1

Example: Addressbook1

The AD LDS service name is created when the instance name is combined with the product name. It will be displayed in the list of Windows services.

AD LDS service display name: LDSinstance1
AD LDS service name: ADAM_LDSinstance1

Figure 8-8: Typing the name of the AD LDS instance.

The wizard now asks you for the account to which you want to give initial administrative privileges.

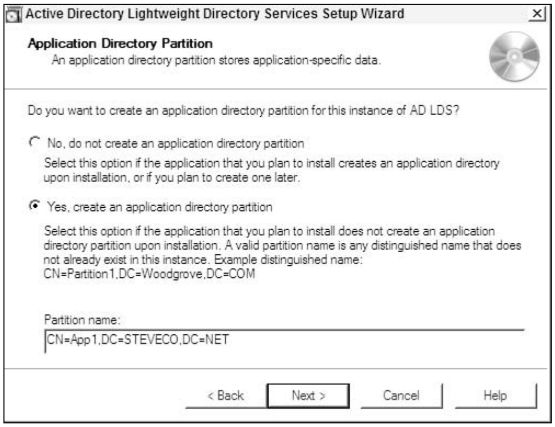
11. **You can either use the account under which you're running the wizard or specify a separate account. Make your selection and then click Next.**

    You're shown a list of optional LDIF (LDAP Delimited Interchange Format) files that can be imported into the application partition when the instance is created. Table 8-1 lists each of these LDIF files and describes why you might want to import them.

12. **For this example, import the file allowing InetOrgPerson objects to be created. Click Next.**

    You're presented with a list of the options you've selected in the wizard. (See Figure 8-10.)

13. **Click Next; the instance is created.**



Active Directory Lightweight Directory Services Setup Wizard

Application Directory Partition

An application directory partition stores application-specific data.

Do you want to create an application directory partition for this instance of AD LDS?

○ No, do not create an application directory partition

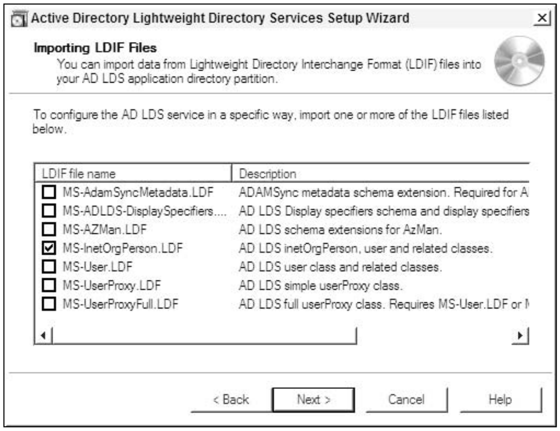
Select this option if the application that you plan to install creates an application directory upon installation, or if you plan to create one later.

● Yes, create an application directory partition

Select this option if the application that you plan to install does not create an application directory partition upon installation. A valid partition name is any distinguished name that does not already exist in this instance. Example distinguished name: CN=Partition1,DC=Woodgrove,DC=COM

Partition name: CN=App1,DC=STEVECO,DC=NET

Figure 8-9: Defining an application partition.



Active Directory Lightweight Directory Services Setup Wizard

Importing LDIF Files

You can import data from Lightweight Directory Interchange Format (LDIF) files into your AD LDS application directory partition.

To configure the AD LDS service in a specific way, import one or more of the LDIF files listed below.

LDIF file name	Description
□ MS-AdamSyncMetadata.LDF	ADAMSync metadata schema extension. Required for A
□ MS-ADLDS-DisplaySpecifiers....	AD LDS Display specifiers schema and display specifiers
□ MS-AZMan.LDF	AD LDS schema extensions for AzMan.
☑ MS-InetOrgPerson.LDF	AD LDS inetOrgPerson, user and related classes.
□ MS-User.LDF	AD LDS user class and related classes.
□ MS-UserProxy.LDF	AD LDS simple userProxy class.
□ MS-UserProxyFull.LDF	AD LDS full userProxy class. Requires MS-User.LDF or M

Figure 8-10: List of optional LDIF files to be imported.

Table 8-1: Optional LDIF Import Files

LDIF File Name	*Description*
`MS-AdamSyncMetadata.LDF`	This file allows for the instance schema to be extended to support the ADAMSYNC tool. If you want to sync the instance with an AD DS domain by using ADAMSYNC, you must import this file.
`MS-ADLDS-DisplaySpecifiers.LDF`	You can use the AD Sites and Services MMC console to administrate how the instance replicates with other replicas. You must import this file to use this console.
`MS-AZMan.LDF`	If you're going to use an application with your instance that supports the Windows Authorization Manager (AZMAN), you must import this file.
`MS-InetOrgPerson.LDF`	RFC2798 defines what a standard user object for an LDAP directory looks like. This helps directory-enabled applications use a standardized user object without having to know which directory service is being used. If your application needs to use an InetOrgPerson-type user object, you must import this file.
`MS-User.LDF`	If you want to create user objects but don't necessarily need to create InetOrgPerson-type user objects, you can import this file.
`MS-UserProxy.LDF`	Import this file if you need to support bind redirection to AD DS.
`MS-UserProxyFull.LDF`	You can import this file instead of MS-UserProxy.LDF which provides for more attributes on the user object. ***Note:*** You must also import either the MS-InetOrgPerson.LDF or MS-User.LDF file along with this file.

Chapter 9

Federating Active Directory

In This Chapter

- Why federate?
- Federation scenarios
- Web services to the rescue!
- Deploying Active Directory Federation Services

This chapter is about federations. Federations? To be honest, the first time I heard that Active Directory was going to support federations I started looking for starships and Vulcans. But, of course, this has nothing to do with a sci-fi TV show and a union of cooperative planets. Rather, I'm talking about how to establish partnerships between companies that want to share access to data and applications. More specifically, Microsoft has developed Active Directory Federation Services to allow "federated" relationships to be established between organizations so that users in one organization can authenticate and access Web applications in another organization without the need of establishing AD DS forest trusts.

So, in this chapter, I discuss what federations are and how Active Directory Federation Services works to provide those federated relationships.

Authentication Everywhere!

While the use of the Internet has evolved, so have the applications that you can place on the Internet. The Internet has provided for the sharing of data between organizations on a level that was never readily achievable before. As such, publishing Web-based software applications and services and then making those applications available to users that are external to your organization (like a partner company) is becoming more commonplace. However, because you need to secure these applications, you have the issue of determining how best to provide authentication services to these external users.

Imagine a scenario (see Figure 9-1) in which a user in Organization Awants to access an external Web application maintained by Organization B. The user in Organization A logs into Organization B's internal AD DS forest, but the Web application that Organization B provides does not use A's forest for authentication. Instead, the Web application uses Organization B's own separate AD DS forest for authentication services.

One solution would be to simply create an ID for the Organization A user in the Web application AD DS forest. However, this approach has many inherent problems including the following:

- You must create a process for the creation, modification, and deletion of these external user accounts. As difficult as creating and modifying accounts can be, the real problem here is the deletion of the external user accounts. Can you really depend on the other company to inform you when a user leaves the company?
- Each of these accounts will have passwords. This means that password resets and forgotten passwords will create an additional burden on your Web administrators.
- Because you are creating a separate external account, these users are going to have to remember another password beyond their normal day-to-day internal user account password. In addition to remembering the extra password, the users have to type it in every time.

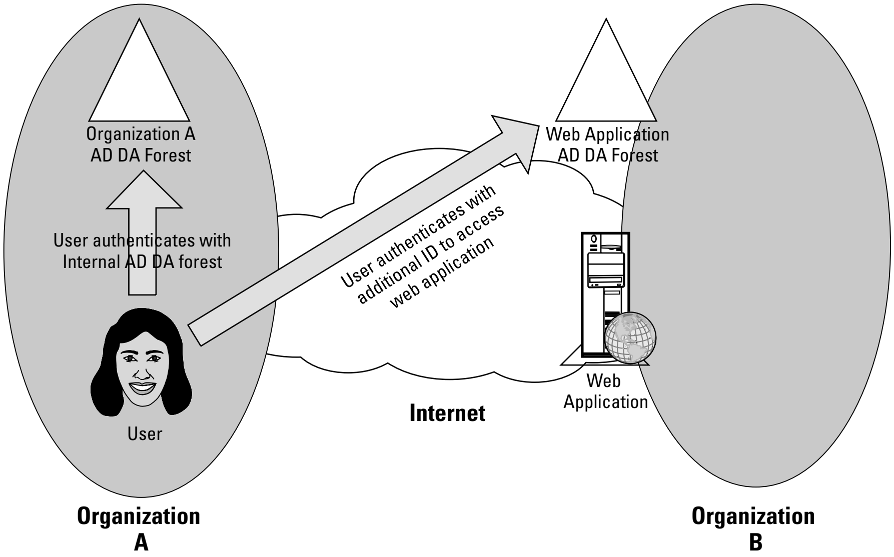


Figure 9-1: A user accessing an external Web application by using an ID in the Web application forest.

- Depending on the industry you're in, you might have to deal with the privacy issues that can arise from sharing information about employees between companies.

If Active Directory Domain Services is being used as the authentication service in both environments, another solution you might consider is establishing a forest trust across the Internet so that users in Organization A could be granted access Organization B's Web application (see Figure 9-2). However, this solution also has problems with it including that you have to expose your AD environment to the Internet and the maintenance of a forest trust relationship over the Internet.

The best solution would be to provide a technology that allows users to log in to their local user accounts, to access all the entitled resources in their environment, and to access the Web applications in any required external organization — all with a single authentication. That's where federations come in. A *federation* is a way to project an identity from a single logon in one environment to another trusting environment and to grant rights to that identity to access resources in the trusting environment. To put it more simply, federations allow you to access applications in other environments by using your current ID so that you don't have to reauthenticate.

TIP

The concept of being able to log in once to access all of your resources, no matter if they are all in the same environment or not, is known as Single Sign-on or SSO.

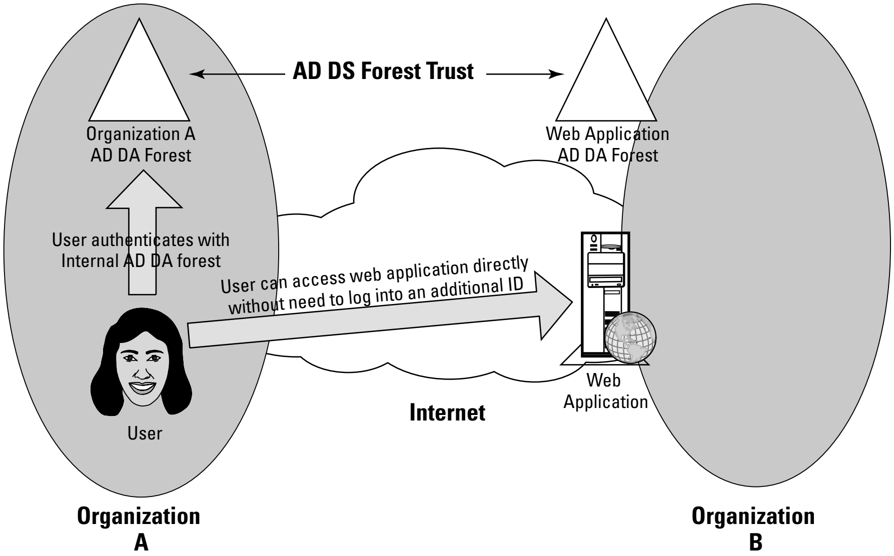


Figure 9-2: A user accessing an external Web application over a forest trust.

Identities, tokens, and claims

Before I go any farther, I need to cover a few federation-related concepts. The first one is the idea of an *identity*. Outside of a computing environment, your identity is a way of describing yourself, and this description is uniquely yours and no one else's. However, your identity can also be defined by an established shared authority. A common example is how a state government identifies you by issuing a driver's license or a birth certificate. In the digital world, too, you can have a digital identity, which is a way for computing systems to identify uniquely that you're you. When you log into Active Directory, your digital identity is established because you *successfully authenticated* (that is, you have the credentials no one else should have, so you must be you!).

That brings me to the next term, *token*. A token is an item issued by a recognized authority to an identity; the token is then used to prove the identity to another authority. In the real world, a driver's license is an example of a token issued by a recognized authority (a state government) that can be used to identify you to other authorities, say a bank, for example. A token is considered *secure* — no one else can use it. A driver's license is considered secure because it has your picture on it; therefore, others can't use it. A token in a computing environment works the same way. When you log in to Active Directory, you receive a unique token that can be presented to other servers to gain access to resources, such as a file or a printer.

REMEMBER

Within the AD DS Kerberos authentication system, you also receive a token used to gain access to resources. This token is a *TicketGranting Ticket,* which I cover in Chapter 14.

The last item is a *claim*. Claims are declarations made by a recognized authority about your identity. Staying with the driver's license analogy, your license contains certain pieces of information about you, such as your name, address, weight, any driving restrictions you have, and so on. These are claims. Other authorities, such as banks, trust driver's license claims because banks trust the authority that issued the license (that is, the state government). In much the same way, when a token is accepted as a part of an authentication process, computers examine claims from your token, such as your logon ID or the security groups of which you're a member. (See Figure 9-3.) Your access to the resources can then be determined on what these claims state about you.

All these items are utilized in a federation solution. Before I continue with federations, I look at one other concept: security token services.

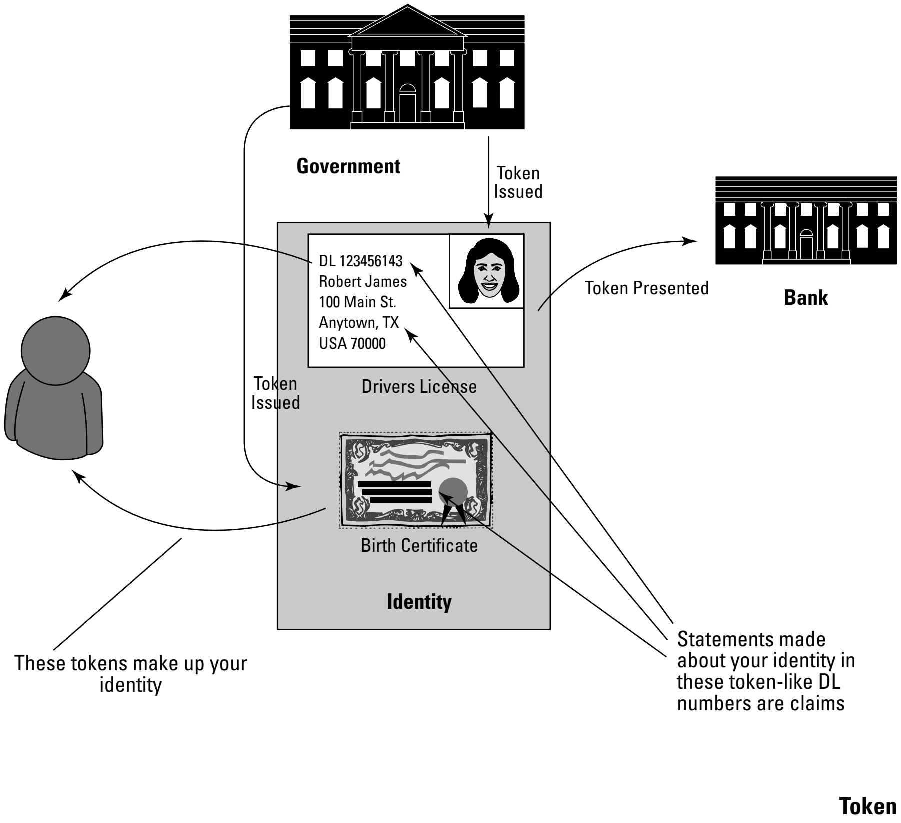


Figure 9-3: Identity, tokens, and claims.

Security token services

A *security token service* is a service that accepts a token from you and returns another token. In the real world, you 're very familiar with such services — you just don't call them that. One good example is when you purchase a movie ticket with a credit card. A credit card can be considered a token because it can be a part of your identity and a known authority (your bank) issues it. When you're ready to make a purchase, you present the card to the cashier. The cashier trusts the token to be authentic because scanning the card confirms the card's authenticity with a bank, and the theater trusts the bank. Then the cashier presents you with a new token (your movie ticket), which in turn allows you to enter the theater and view the movie. In this case, the cashier is the security token service. So the security token services also present you a token so that you can gain access to something that the original token did not authorize you to access.

Another good example is you going to the airport to get a plane ticket. At the ticket counter, you have to present a token — some sort of government-issued ID or passport — to get your ticket. The ID is the initial token; the plane ticket you receive is the token that's issued by the security token service (the ticket agent).

Federations

When you understand identities, tokens, claims, and security token services, you can look at what a federation is. A *federation* is a means for you to project your authenticated identity from one computing environment to another. With a federation, you can log into your home environment that stores your account and project your identity to any target or resource environment in which you want to access a resource (say, a Web application) without having to reauthenticate yourself. You can do this because of the trust relationship (that is, federation) between the two environments. The trust relationship is typically one-way from the resource environment that contains the Web application to the account environment that contains the user. (See Figure 9-4).

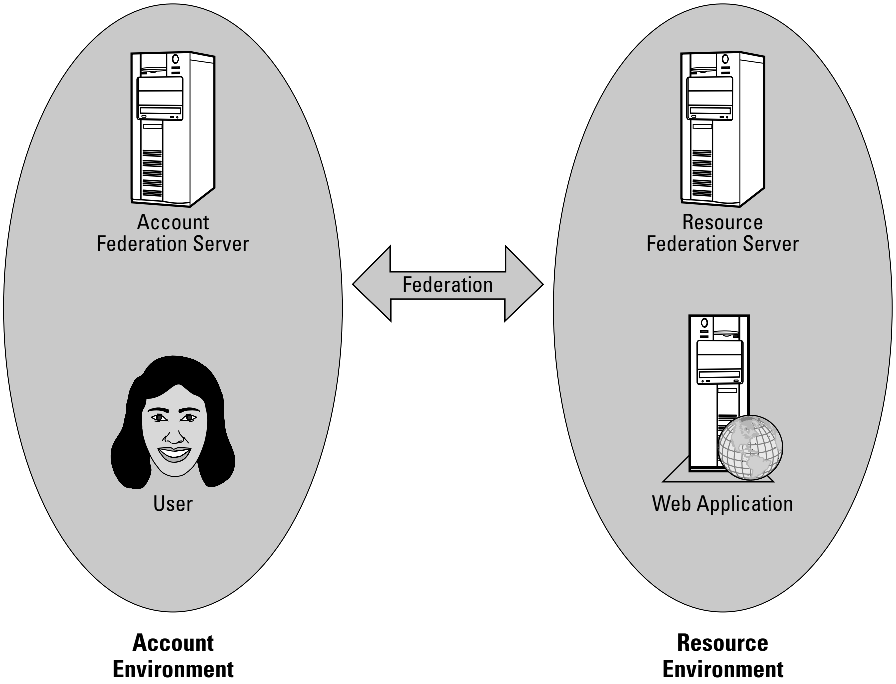


Figure 9-4: A federation.

Active Directory Federation Services (AD FS) is Microsoft's implementation of this federation concept. A server running AD FS acts as a security token service in support of the federation. But, what is AD FS specifically designed to provide? First, AD FS is designed around WS-* Specification; therefore, it's designed to be an SSO solution for providing users federated access to Web applications.

Take a look at how AD FS works. Figure 9-5 shows a federation authentication process between a client and a Web application along with the associated account and resource AD FS servers. The following steps explain how the client gets SSO access to the Web application.

1. The client attempts to access the Web application by using a Web browser. The application requires the user to be authenticated, but the only credentials the user has are from his environment (the account environment), which has no meaning to the Web application in the resource environment. At this point, the user doesn't have an AD FS authentication cookie (that is, a token); the user receives the token at the end of this process. However, if the cookie were available, the browser would provide it to the Web application and allow the user access without the need of logging in (this is Single Sign-on). But, because the user doesn't yet have the cookie and the Web application is enabled to support federated authentication, the client's browser is redirected to the resource AD FS server.

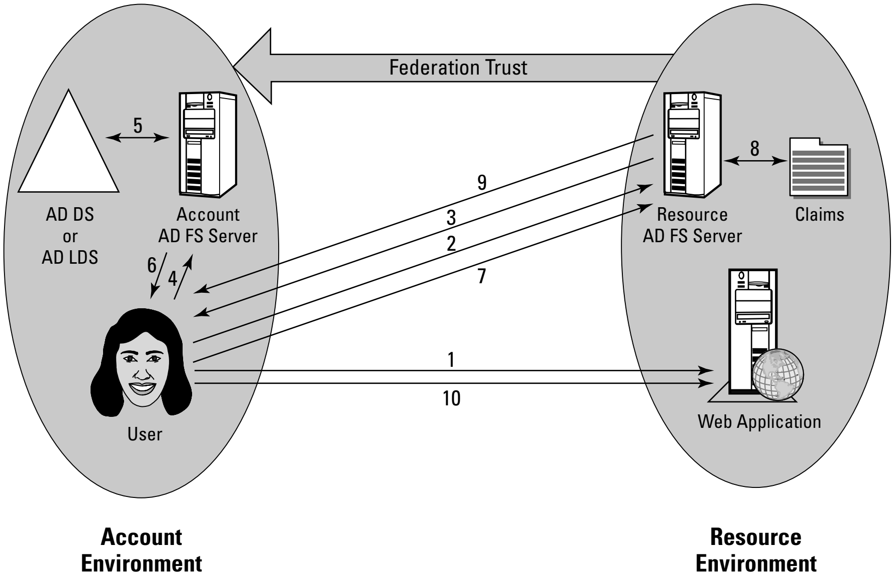


Figure 9-5: The AD FS authentication process.

2. When the browser is directed to the resource AD FS server, the server looks at its list of federation partners it trusts and determines that the user is in a trusted account environment. This is because the user has previously authenticated with the AD DS environment in the account environment.
3. The user's browser is instructed by the resource AD FS server to obtain a security token from the account AD FS server. The resource AD FS can do this because of the preexisting federation relationship that was established.
4. The user's browser presents the authentication token from the resource AD FS server to the account AD FS server.
5. At this point, the account AD FS server authenticates the user against the AD DS or AD LDS account store that the account AD FS is configured to use. When this authentication is successful, the account AD FS server pulls the associated claims about the user that the resource AD FS server is configured to examine to determine whether the user can be allowed access to the Web application. This information is packaged into a new security token.
6. The account AD FS server provides the user's browser with the new token and redirects the browser to the resource AD FS server. ***Note:*** This is where the account AD FS is acting as a security token service because it has taken the original token (the user's logon into AD DS) and has translated it into a new token to be recognized by the resource AD FS server.
7. The browser sends the token to the resource AD FS server to examine the claims that have been made about the user.
8. Based on the claims in the token (such as the user's UPN, e-mail address, security group membership, and so on) the resource AD FS server determines whether the user can be allowed access to the Web application. A policy defined within the resource AD FS states what valid claims a user must have to access the application. If the claims-check is successful, a new security token is generated that allows the user to access the Web application.
9. The token is sent to the browser where it's stored as an authentication cookie on the user's computer.
10. The token is then presented to the Web application where the user gains successful access to the application.

REMEMBER

The tokens exchanged in this process are not encrypted — they're signed to guarantee authenticity. But, to make sure that all the traffic is secure, the communications among the browser, AD FS servers, and the Web applications are encrypted by using Transport Layer Security/Secure Socket Layer (TLS/SSL), which is the technology that is commonly used to encrypt data that is transmitted over the Internet. I cover the certificates required for AD FS later in the chapter.

That's it! Simple right? Well, okay, it's not that simple, but I hope you get the idea. At the most simplistic level, federations are about providing SSO access to Web applications.

Federation Scenarios

In the following sections, I run through the three primary federation scenarios for which Microsoft has specifically positioned AD FS to provide a solution.

Web single sign-on scenario

In this situation, you have a number of Web applications you want to make available to Internet users. These users have to authenticate to the Web application, but you want that logon to occur only once, even if the user accesses multiple applications. This scenario is diagramed in Figure 9-6.

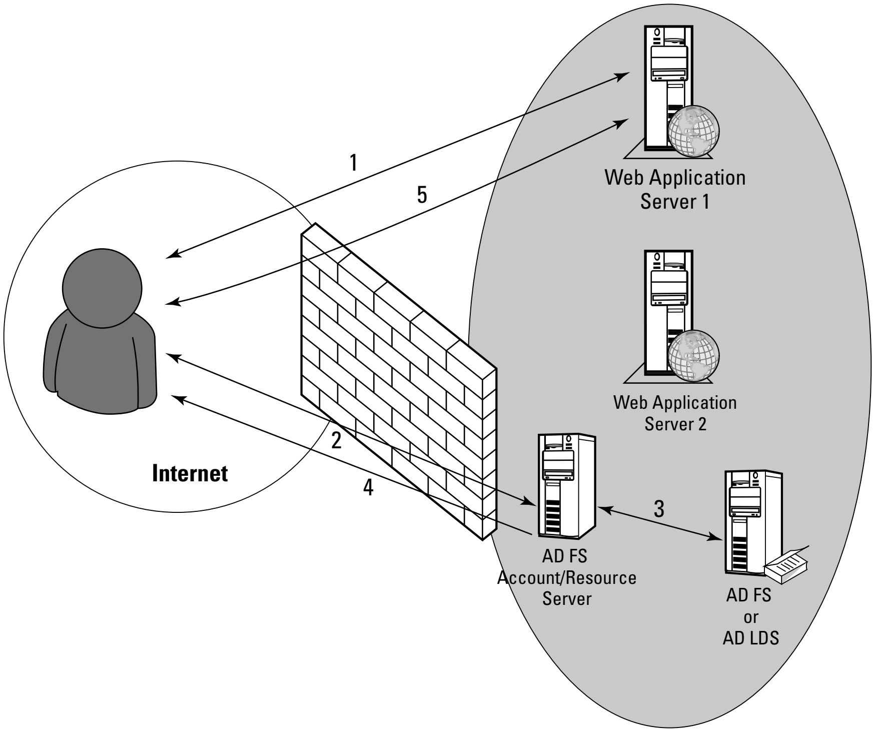


Figure 9-6: The Web single sign-on scenario.

Of the three scenarios I cover, this one is the simplest. You require only one AD FS server, which acts as both the account and resource federation server. The Web application server is enabled to support federations. The AD FS server also needs either access to an AD DS domain controller or to an AD LDS instance. Because this scenario deals with external users accessing an application, you might prefer using AD LDS for this solution because it allows you to separate those external user IDs from your internal AD DS directory. The AD FS, directory, and Web application servers are located in a perimeter network so that the external users can access these servers without having to expose any internal infrastructure.

When the external user accesses the application on Web Application Server 1 (arrow 1 in Figure 9-6), the federation-enabled application checks whether the user already has an authentication cookie that can enable access. In this case, the user doesn't have the cookie yet, so the application redirects her browser to the AD FS server. The browser requests to log in with the AD FS server (arrow 2), the user is presented with the logon screen for the AD FS server, and the user types in her credentials. When the AD FS server receives these credentials, they're sent to the account store (AD DS or AD LDS) for validation (arrow 3). The AD FS server then constructs a token (which includes the claims for the user), using the information provided by the account store. Then the token is sent to the user (arrow 4) and the token is stored on the user's computer as an authentication cookie. The browser is redirected back to the application on Server 1, but this time it provides the authentication cookie as a means of logging in to the application and is successful in accessing the application (arrow 5).

WARNING!

In showing the communication flow that occurs in these scenarios, I'm leaving out some of the steps to simplify the explanation. If you want more information, Microsoft's Windows Server 2008 TechCenter Web site provides details about what occurs in these communications. You can find the web site at:

```
http://technet.microsoft.com/windowsserver/2008
```

The benefit of deploying this federation scenario is that it provides a method of secure access to the application. When the user has the authentication cookie, it can be used to access other applications (on Web Application Server 2, for example) without the need for the user to log in a second time.

Federated Web SSO scenario

This scenario is probably the typical use of AD FS: to provide users in one company federated access and SSO services to Web applications provided by a partner company. This scenario is pretty much the same one I run through in Figure 9-5, but I cover it one more time. (See Figure 9-7.)

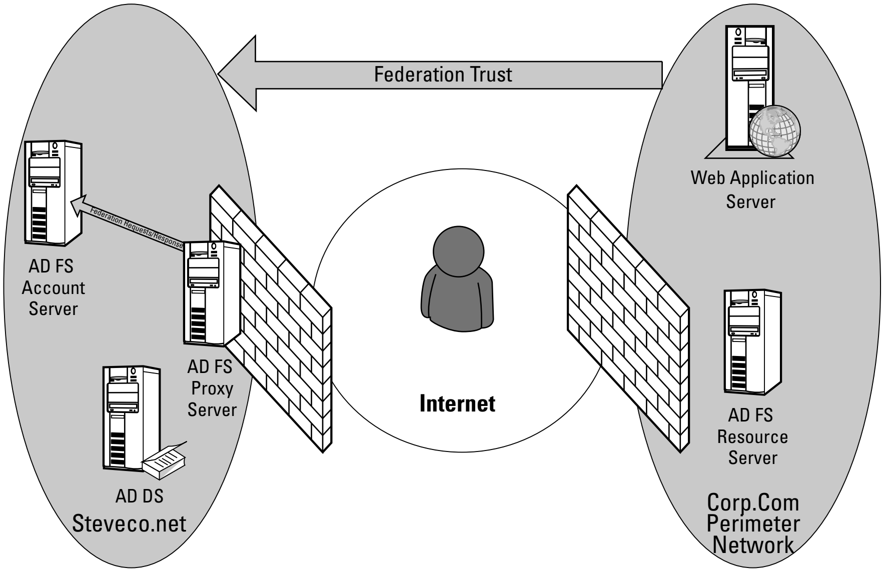


Figure 9-7: The Federated Web SSO scenario.

Figure 9-7 shows two companies (Steveco.Net and Corp.Com). A federation exists between these two companies with Corp.Com trusting Steveco.Net. This scenario requires that an AD FS server be deployed in both companies with Steveco.Net having the account federation server and Corp.Com having the resource federation server. AD DS is being used as the account store in Steveco.Net because internal users in Steveco need to access a federated Web application in Corp.Com.

An internal user in Steveco.Net attempts to access the Web application in the perimeter network of Corp.Com (arrow 1). Because the user doesn't have the authentication cookie yet, the user's browser is redirected to the resource AD FS server (arrow 2) that the Web application server is configured to use. The resource AD FS server determines which federation partner the user is a member of and redirects the user to the Steveco.Net account AD FS server. The account AD FS server obtains the user's logon credentials (arrow 3) that were obtained when the user initially logged in to the AD DS forest and validates them against the AD DS forest (arrow 4). The security token for the user is built and then sent to the user (arrow 5). The user's browser is directed to the resource AD FS server in Corp.Com, and the browser presents the token from the account AD FS server. The resource server then validates the token and examines its claims to determine whether the user can access the application. If the claims are valid, then the resource AD FS builds a new token that it sends to the user's browser (arrow 6). This token is stored locally as an authentication cookie. The new token is then sent to the Web application server, where it's validated, and user access is granted to the application (arrow 7).

One variation of this scenario is the user being a remote user on the Internet (see Figure 9-8). In this case, the servers remain where they are, but you have the option of deploying an AD FS *proxy server* — an AD FS server that can relay the federation requests and responses between the user on the Internet and the internal AD FS account server. A proxy server runs only a subset of the software that a normal AD FS federation server does, but by deploying an AD FS proxy, you can now allow external users to have the same level of federated SSO access to the Corp.Com applications as an internal user would have.

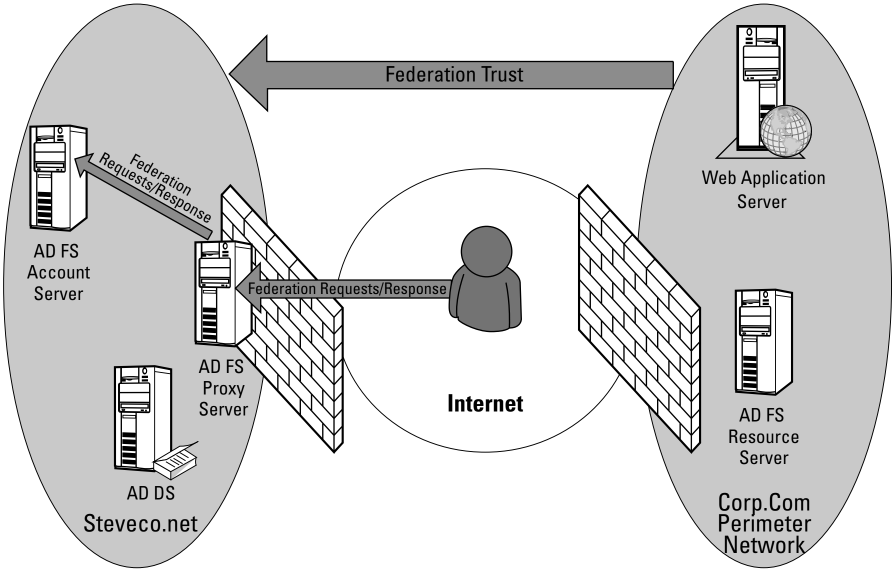


Figure 9-8: The Federated Web SSO scenario with an AD FS proxy server.

Federated Web SSO with forest trust scenario

The last scenario is similar to the first scenario in that only one company is involved. However, here, you're concerned with providing a traveling employee SSO access (via the Internet) to the Web applications as well as providing internal users the same level of access. (See Figure 9-9.)

The Web application in the perimeter network uses the AD DS forest in the perimeter network as a default authentication service (that is, Windows integrated authentication is enabled). A one-way forest trust is in place between the perimeter network forest and the intranet forest. Internal employees can access the Web application by using their internal AD account without the need of creating new IDs in the perimeter network forest. Additionally, internal employees can authenticate via Windows integrated authentication on the Web server. The external user authenticates via TLS/SSL or forms authentication. This example is still a Web application enabled for federated authentication; therefore, the federated token (that is, the authentication cookie) is still generated for either user to provide the SSO functionality to the other Web applications in the perimeter network.

The communication flow for this scenario follows the same process I describe in the other scenarios. But, keep in mind that when you need to provide Web applications to a company's internal and external employees, this scenario is the one to consider.

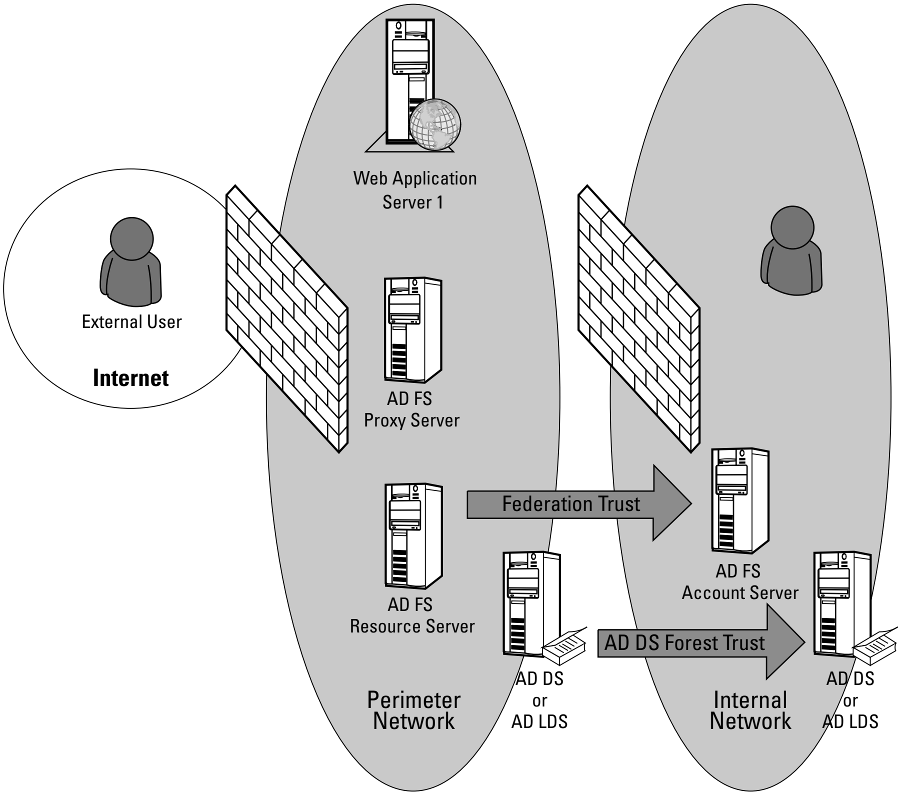
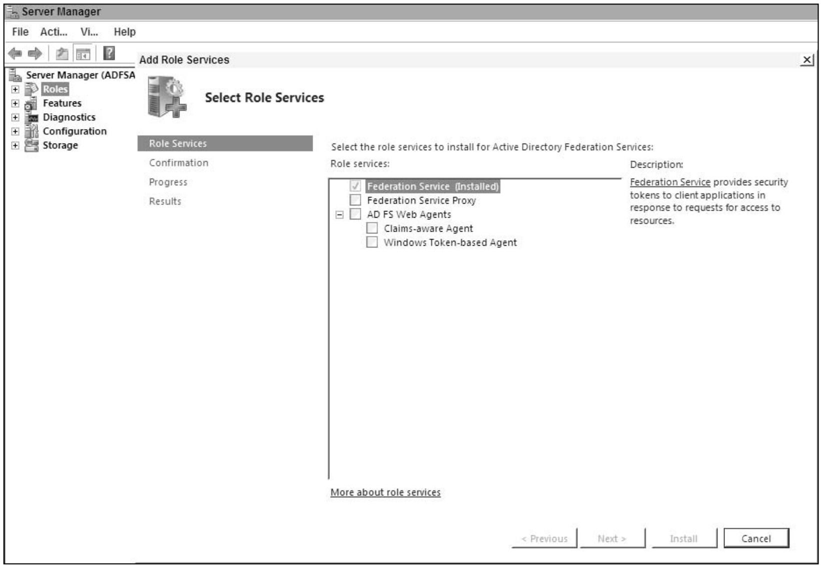


Figure 9-9: The Federated Web SSO with forest trust scenario.

Deploying Active Directory Federation Services

Active Directory Federation Services was introduced in the R2 update of Windows Server 2003. Windows Server 2008 offers several improvements in AD FD, including support for AD Rights Management Service (I cover this in the next chapter) and the ability to import and export trust policies which makes establishing the federated trust between an account and resource federation server simpler. AD FS is a role that can be installed on a server by using the Server Manager tool. When you select the AD FS role, you're prompted for the role service you want to install. These role services include the following:

- **Federation Service:** You select this role service when you're installing either an account or a resource federation server.
- **Federation Service Proxy:** This role service is for installing an AD FS proxy server.
- **AD FS Web Agent:** This role service must be installed on the Web application server that will be participating in the federation. You also have two options here: installing a claims-aware agent that is designed to support an application that can use claims-based authentication or to install an Windows Token-based agent that supports a Windows authentication-based Web application. (See Figure 9-10.)



Select the role services to install for Active Directory Federation Services:

Role services:

- ☑ Federation Service (Installed)
- □ Federation Service Proxy
- □ AD FS Web Agents
  - □ Claims-aware Agent
  - □ Windows Token-based Agent

Description:

Federation Service provides security tokens to client applications in response to requests for access to resources.

Figure 9-10: Selecting the various AD FS role services.

In addition to installing the AD FS role on the participating server, an AD FS server must also have the following software installed:

- Internet Information Server (IIS).
- ASP.NET 2.0.
- .NET Framework 2.0.
- The certificates to support the signing and encryption required by the federation service. This includes certificates that must be on the AD FS servers (both resource and account) as well as any of the Web application servers that are participating in the federation. These certificates need to be issued from a certificate authority that's trusted by all users and servers within the federated environment. This certificate authority can be from an external provider like Verisign or can be an internal one like AD CS can provide.

The configuration of AD FS is primarily done in the IIS console and the AD FS console (which is installed when you install the AD FS role). Setting up the federation trust and the federation-enabled Web application is rather complex and beyond what I cover in this book, but I want to cover the high-level steps.

1. **Create the federation trust policy on the account and resource federation servers.**

   This is done within the AD FS console. Typically, you create the trust policy on the account server and then export it to an XML file that can then be imported to the resource server. At this point, you also need to get the necessary certificates for the signing and encryption process.

2. **Define the organizational claims that will be used to determine whether the federated user is allowed to access the application.**

   In this step, you define what the user's claims must state in order for the user to be granted access to the Web application. As you might recall, a claim is a way of making statements about the user attempting to access the Web application, and access is granted or denied based on what these claims state. You can match up claims on either the user's UPN, e-mail address, or security group membership.

3. **The account stores for each federation service must be created and defined.**

   AD DS and AD LDS are support account stores for AD FS.

4. **Ensure that your Web application is configured properly to work within the federation.**

Of course, the detail within each of these steps will vary greatly depending on which federation scenario you deploy and your security requirements. Microsoft offers a lot more detail about these steps in the Windows Server 2008 TechCenter Web site.

Chapter 10

AD Certificate Services and Rights Management Services

In This Chapter

- Understanding a public key infrastructure
- Inside of AD Certificate Services
- Managing how information is used
- Inside of AD Rights Management Services

In this chapter, I take a quick look at the two remaining components of the Windows Server 2008 Active Directory umbrella: Active Directory Certificate Services and Active Directory Rights Management Services. Both of these services provide different aspects of an overall security solution: Certificate Services provides a solution for issuing and managing certificates to be used for data encryption or secure authentication, and Rights Management Services provides a solution for controlling the use of digital content. So let's get started!

Active Directory Certificate Services

Microsoft has been providing AD Certificate Services (AD CS) as a part of the Windows Server software since 2000. But, in 2008, the service has been rebranded as being a part of Active Directory and now includes a good number of enhancements that make its use even more attractive. But before I get into what is in AD CS, I want to cover what certificates are and what a Certificate Authority (CA) is.

What is public key infrastructure (PKI)?

Public key infrastructure (PKI) is a term that gets used a lot these days, particularly as we are forced to become more and more security conscious. PKI addresses several security needs, including the need to secure the transfer of information between two parties by providing for the encryption of the data. PKI also provides the means to ensure that data hasn't been altered during the transmission (through digital signatures) and that the data really came from the party that you're told it's from.

Let me define some terms upfront:

- **Keys:** These are pieces of data used in cryptographic algorithms to *encrypt* (seal) data, or verify the authenticity of an *entity* (user or computer) and that data hasn't been modified (known as *signing*). Keys can be either public or private. A *public key* is a key that is publically available, and a *private key* is known only to the entity that owns the key.
- **Symmetric Encryption:** A type of encryption in which the key used to encrypt the data is the same key used to decrypt the data. Note that because both the encrypting entity and the decrypting entity are using the same key, the key is a public one.
- **Asymmetric Encryption:** This scenario uses a public and private key pair that is associated with each other. With this type of encryption, one of the keys is used to encrypt the data in such a way that only the corresponding second key is capable of decrypting the information.
- **Digital Signature:** A digital signature provides a way of ensuring that a received piece of data hasn't been tampered with while it was transferred over the network, and that it actually came from the identified source. The signature is actually the *hash* of the data. A hash is a unique number that is mathematically created by using the data for which the digital signature is being created. If the calculated hash for a piece of data that's been transmitted is identical for the sender and receiver, then you know that the data has not been altered.
- **Certificate:** This is an electronic proof of identity. Certificates are normally issued to an entity as a way of allowing the entity to prove that they are authentic (that is, I am who I say I am). Certificates have a number of pieces of data within them and the data contained can vary based on what the certificate is going to be used for (encrypting e-mail, smartcard authentication, encrypting a Web site, and so on). The various pieces of data the certificate contains includes a public key, a digital signature that can be used to verify the certificate's authenticity and prove what certificate authority issued it, and a date/time stamp indicating how long the certificate is valid.
- **Certificate Authority (CA):** A CA is a recognized authority responsible for issuing certificates. Sometimes, this is referred to as a *certification authority*.

PKI can be considered the combined use of certificates, keys, and CAs to enable the signing and sealing of data. The primary thing that enables PKI for a company is the establishment of a trusted certificate authority. By creating a CA you can issue certificates to the users and computers.

However, just because a CA issues a certificate doesn't mean that other entities immediately trust that the certificate and its associated public key are authentic. The trusting of such a certificate occurs when that entity trusts the certificate of the certification authority that issued that certificate. (See Figure 10-1.)

CAs provide the following:

- **Certificates** to users and computers.
- **Verification** of the entity requesting the certificate (also known as *enrollment*).
- **A Certificate Revocation List (CRL)** that is used to mark previously issued certificates as revoked. This list is necessary so that when a certificate is compromised (or is issued to an entity that no longer needs the certificate), the CA can mark the certificate as revoked by adding it to the CRL. PKI clients go through a number of checks in validating a certificate, including checking the CRL of the issuing CA to verify that the certificate hasn't been revoked before using it. The CRL is typically a file that is available through HTTP, FTP, or LDAP.

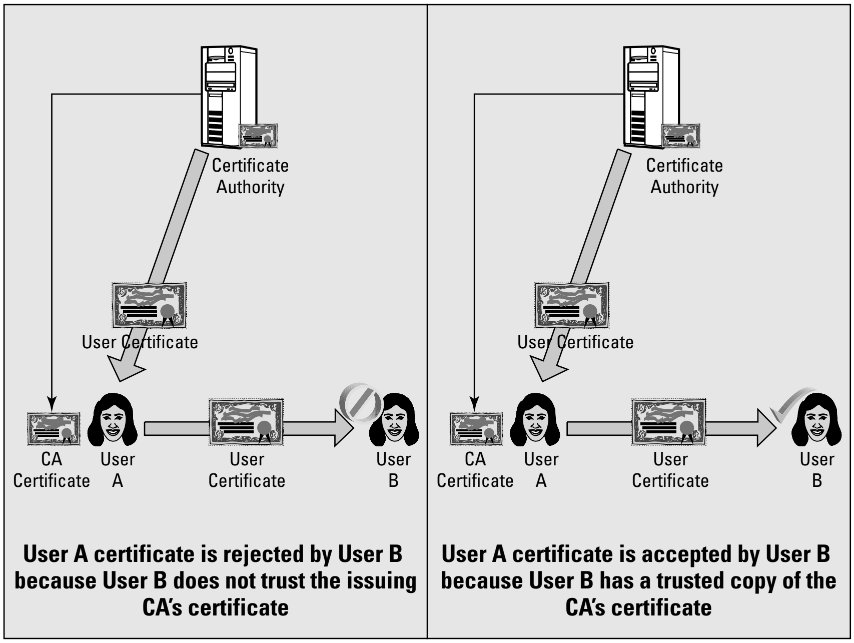
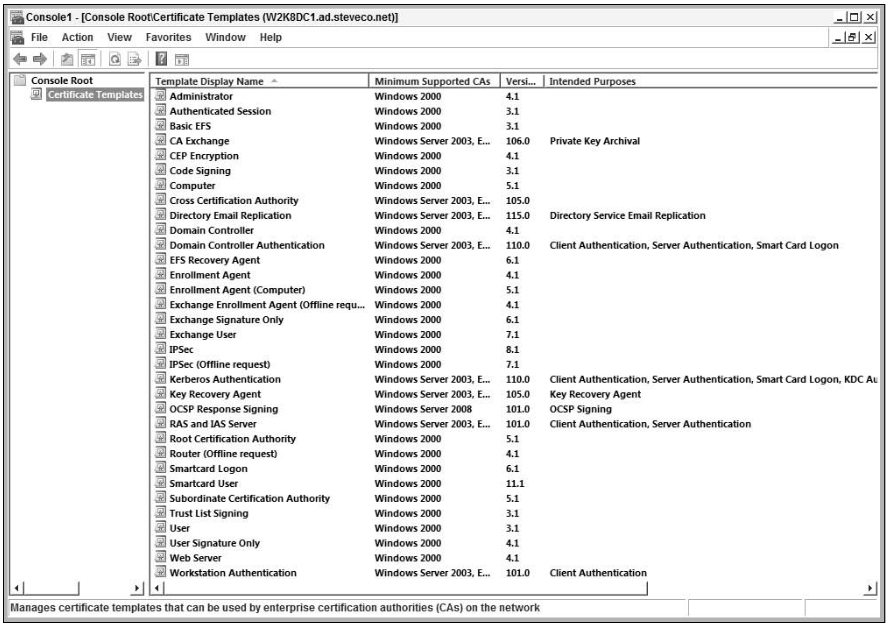

Certificate Authority
User Certificate
CA Certificate
User A
User Certificate
User B
User A certificate is rejected by User B because User B does not trust the issuing CA's certificate
Certificate Authority
User Certificate
CA Certificate
User A
User Certificate
User B
User A certificate is accepted by User B because User B has a trusted copy of the CA's certificate

Figure 10-1: How certificate authorities provide trusted certificates.

In Windows Server 2008 AD CS, there's a new, additional method for publishing the CRL known as the Online Responder Service. I cover the ORS later in this chapter.

The issue that has to be resolved is the location of the CA. You have the option of obtaining certificates from public CA servers on the Internet (for a cost of course), or you can establish your own CA infrastructure. This is what AD CS is really designed for. AD CS allows you to establish a CA (or multiple CAs) within your company so that you can issue certificates and keys for a variety of purposes. In fact, AD CS comes with a large number of certificate templates, each having a unique use. (See Figure 10-2.)


Console1 - [Console Root\Certificate Templates (W2K8DC1.ad.steveco.net)]
File
Action
View
Favorites
Window
Help
Console Root
Certificate Templates

Template Display Name	Minimum Supported CAs	Versi...	Intended Purposes
Administrator	Windows 2000	4.1	
Authenticated Session	Windows 2000	3.1	
Basic EFS	Windows 2000	3.1	
CA Exchange	Windows Server 2003, E...	106.0	Private Key Archival
CEP Encryption	Windows 2000	4.1	
Code Signing	Windows 2000	3.1	
Computer	Windows 2000	5.1	
Cross Certification Authority	Windows Server 2003, E...	105.0	
Directory Email Replication	Windows Server 2003, E...	115.0	Directory Service Email Replication
Domain Controller	Windows 2000	4.1	
Domain Controller Authentication	Windows Server 2003, E...	110.0	Client Authentication, Server Authentication, Smart Card Logon
EFS Recovery Agent	Windows 2000	6.1	
Enrollment Agent	Windows 2000	4.1	
Enrollment Agent (Computer)	Windows 2000	5.1	
Exchange Enrollment Agent (Offline requ...	Windows 2000	4.1	
Exchange Signature Only	Windows 2000	6.1	
Exchange User	Windows 2000	7.1	
IPSec	Windows 2000	8.1	
IPSec (Offline request)	Windows 2000	7.1	
Kerberos Authentication	Windows Server 2003, E...	110.0	Client Authentication, Server Authentication, Smart Card Logon, KDC Au
Key Recovery Agent	Windows Server 2003, E...	105.0	Key Recovery Agent
OCSP Response Signing	Windows Server 2008	101.0	OCSP Signing
RAS and IAS Server	Windows Server 2003, E...	101.0	Client Authentication, Server Authentication
Root Certification Authority	Windows 2000	5.1	
Router (Offline request)	Windows 2000	4.1	
Smartcard Logon	Windows 2000	6.1	
Smartcard User	Windows 2000	11.1	
Subordinate Certification Authority	Windows 2000	5.1	
Trust List Signing	Windows 2000	3.1	
User	Windows 2000	3.1	
User Signature Only	Windows 2000	4.1	
Web Server	Windows 2000	4.1	
Workstation Authentication	Windows Server 2003, E...	101.0	Client Authentication

Manages certificate templates that can be used by enterprise certification authorities (CAs) on the network

Figure 10-2: The list of certificate templates available in AD CS.

Inside AD Certificate Services

AD CS is available on all editions of Windows Server 2008 with the exception of the Web Server edition. However, if you're looking to deploy all the available features of AD CS, you should be deploying the role on either an Enterprise or Datacenter edition of Windows Server 2008. Table 10-1 shows what features of AD CS are available by server edition.

Table 10-1: AD Certificate Services Features by Windows Server 2008 Editions

Features	*Windows Server 2008 Standard Edition*	*Windows Server 2008 Enterprise/Datacenter Edition*
Act as a CA	*	*
Support for the Online Responder Service		*
Clients can request certificates through a Web interface		*
Network devices can enroll for certificates		*
Custom certificate templates		*

Like all other Windows Server 2008 server roles, AD CS can be installed through the Server Manager tool. By installing AD CS, you're specifying that the server is to become a certificate authority. You can deploy two types of CA: an Enterprise CA or a Standalone CA. The primary difference between these types is the level of default integration with AD DS. Table 10-2 shows the differences between these two types of CAs.

Table 10-2: Enterprise versus Standalone Certificate Authorities

Ability	*Enterprise CA*	*Standalone CA*
Must have access to AD DS forest	Yes	No
Uses Group Policy to publish its CA certificate	Yes	No (by default, but certificate can be distributed by GPO manually)
Automatically publish issued certificates and CRL in AD DS	Yes	No
Use AD DS authentication to identify a requestor during certificate enrollment	Yes	No
Support for certificate auto-enrollment	Yes	No
Support for customized certificate templates	Yes	No

From this table's material, you can see many advantages to using an Enterprise CA. So why would you ever use a standalone CA? One answer is for times when you need to establish a CA in a region of your network in which you don't have access to your AD DS forest (like in a DMZ). In cases similar to this, a standalone CA is a good solution.

CA hierarchies

When establishing PKI within a medium- to large-sized company, deploying only one CA has some inherent problems. First is the lack of fault tolerance. If the single CA goes down, no one can enroll for new certificates and the CRL might not be available (assuming that you deployed a standalone CA that doesn't publish the CRL to AD). Also, a single CA isn't a good solution when your company is geographically dispersed. Additionally, you might want to have different CAs dedicated to providing certain types of certificates. Although you can simply deploy multiple CAs, the problem is that all the entities using the certificates must trust all the CAs that issued the certificates. Therefore, you need to publish all the CA's certificates — and having to keep up with multiple CA certificates creates an administrative headache.

Fortunately, there's an easy solution. You can create hierarchies of certificate authorities. At the top of the hierarchy is the root CA. One or more levels of subordinate CAs can be underneath the root CA. What makes a CA subordinate to another is the fact that the higher level CA signed the lower level CA's certificate. Therefore, the question is who signs the root level CA if it has no superior. A root CA is the root of the hierarchy because it uses a self-signed certificate. (See Figure 10-3.)

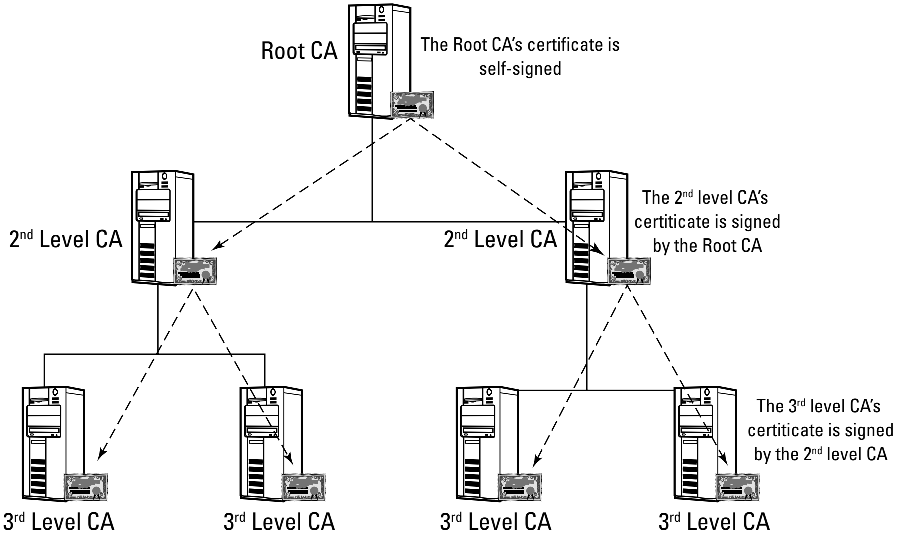

Root CA
The Root CA's certificate is self-signed
2nd Level CA
2nd Level CA
The 2nd level CA's certiticate is signed by the Root CA
The 3rd level CA's certificate is signed by the 2nd level CA
3rd Level CA
3rd Level CA
3rd Level CA
3rd Level CA

Figure 10-3: A Certificate Authority hierarchy.

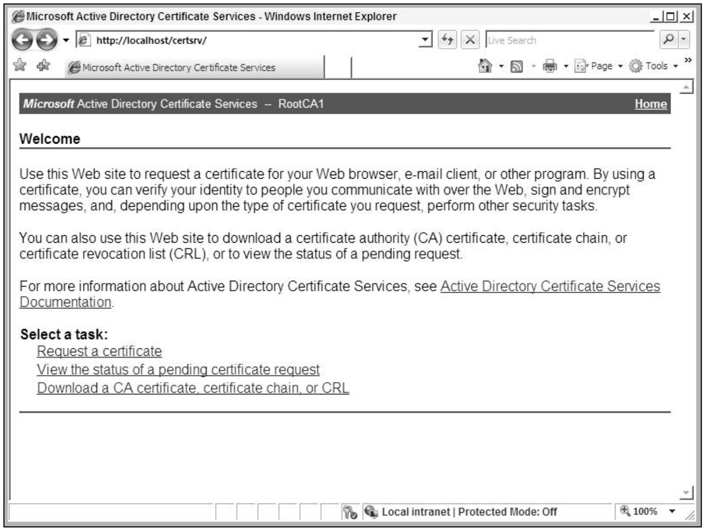
One of the great benefits of a CA hierarchy is that an entity that trusts the Root CA automatically trusts all the subordinate CAs, which eliminates the need for users and computers to trust each CA in the hierarchy.

TIP

A best practice to consider if you do establish a CA hierarchy is to take the root CA offline and then store the data file containing the root CA certificate in a very secure place. Because the root CA isn't needed on any regular basis after the subordinate CAs are created, taking the root CA offline doesn't affect the users. Nevertheless, by taking the root CA offline, you do ensure that it's not available to be compromised by hackers.

Web Enrollment

When you install the AD CS server role you have the option of installing an optional role service called Web Enrollment. Typically, Windows computers and users connected into the domain can use auto-enrollment features of AD CS to enroll for certificates. However, in the case of non-Windows computers and other computers not connected to the domain, the Web Enrollment service provides for a way to request new certificates through a web interface. ***Note:*** This role service doesn't have to coincide with the actual CA server; in fact, installing the Web Enrollment features on a separate server creates a more secure environment. (See Figure 10-4.)


Microsoft Active Directory Certificate Services - Windows Internet Explorer
http://localhost/certsrv/
Live Search
Microsoft Active Directory Certificate Services
Page
Tools
Microsoft Active Directory Certificate Services -- RootCA1
Home
Local intranet | Protected Mode: Off
100%

Welcome

Use this Web site to request a certificate for your Web browser, e-mail client, or other program. By using a certificate, you can verify your identity to people you communicate with over the Web, sign and encrypt messages, and, depending upon the type of certificate you request, perform other security tasks.

You can also use this Web site to download a certificate authority (CA) certificate, certificate chain, or certificate revocation list (CRL), or to view the status of a pending request.

For more information about Active Directory Certificate Services, see Active Directory Certificate Services Documentation.

Select a task:
- Request a certificate
- View the status of a pending certificate request
- Download a CA certificate, certificate chain, or CRL

Figure 10-4: The initial Web Enrollment screen.

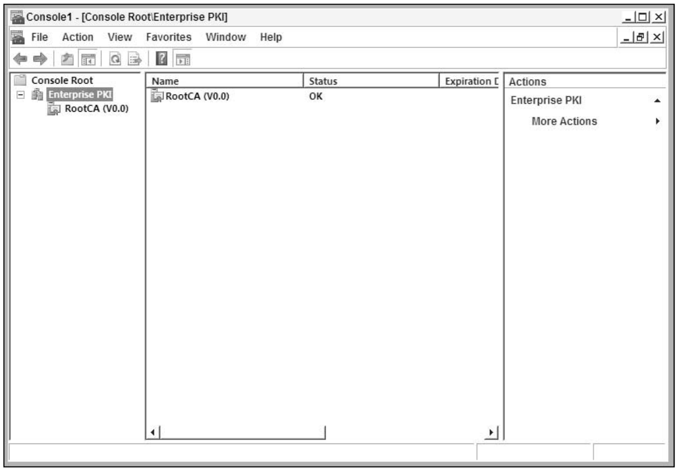
Online Responder Service

One of the scalability problems that most PKI infrastructures have is providing clients access to the CA CRL. With time, more certificates are being revoked; therefore, the CRL becomes larger and larger. This can create a client performance problem as the entire CRL would have to be examined to determine if the certificate is still valid. Another problem with CRLs is the fact that CRLs are updated only on a periodic basis, so a revoked certificate might not immediately appear in the CRL file.

RFC 2560 proposes the Online Certificate Status Protocol (OSCP) that addresses these issues. AD CS supports OSCP through the Online Responder Service (ORS) service. This is an another optional service that you have the opportunity to install at the time you install the AD CA server role. Like the Web Enrollment role service, it's recommended that you deploy the Online Responder Service on a server separate from the CA. To provide even greater scalability, you can deploy the Online Responder Service on multiple servers, creating an Online Responder Service array that can handle a large number of OSCP requests.

Enterprise PKI console

I'm happy to say that one of my favorite PKI tools from the Windows Server 2003 resource kit is now a standard console in Windows Server 2008. The tool was the PKI Health tool, but now it's in a new console named Enterprise PKI. From this console, you can check the health of your CA infrastructure and view any error conditions that might exist so that you can take corrective action. Definitely check out this tool if you're going to be deploying AD CS within your company. (See Figure 10-5.)



Name	Status	Expiration D
RootCA (V0.0)	OK	

Figure 10-5: The Enterprise PKI console.

TIP

More information about the Web enrollment service, Online Responder Service, and the Enterprise PKI console can be found in Microsoft's Windows Server 2008 TechCenter at:

```
http://technet.microsoft.com/windowsserver/2008
```

Active Directory Rights Management Services

One of the current trends in security doesn't involve the authentication of users or the authorization to data but rather the management of the data when you get access to it. The last component of Active Directory that I cover addresses this area.

Managing information usage

Managing what users do with information they're authorized to have is a popular security topic these days. Typically, IT security systems center primarily on securing the information that's in transit in the network and making sure that only authorized users can access the information. But what do you do to prevent authorized users from misusing the information when they get access to it? This misuse can range from the completely accidental to the downright malicious, including:

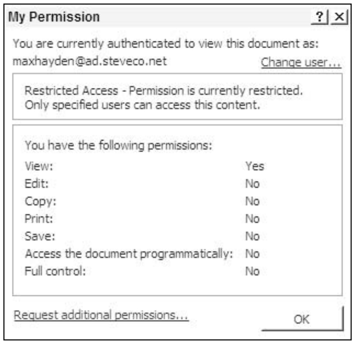
- Printing a confidential document to a shared printer, allowing someone that doesn't have access to the document to read it
- Sharing a document with other users, unaware that the document is confidential
- Editing a document to change its contents without the document owner knowing
- Intentionally sharing confidential data with the press or a competitor to harm the company

In addition to these reasons, new government laws, such as the Health Insurance Portability and Accountability Act (HIPAA) and Sarbanes-Oxley Act in the United States, expressly control who can view and share certain pieces of data. Among the public, the subject of digital rights management (DRM) is also a hot topic in information usage, particularly in the area of preventing copyright infringement in the music and movie industries.

While collaboration within and between companies continues, security technology needs to address these risks and laws because these issues won't be going away. The goal of Active Directory Rights Management Services (AD RMS) is to address these issues within a Windows technology-based environment. AD RMS allows a document owner to control exactly what can be done with a document after an authorized user gains access to it. This includes control of viewing, copying, printing, saving, and e-mailing. (See Figure 10-6.)

WARNING!

Because this area of security technology is relatively new, circumventing the security in ways these tools can't prevent is still possible. For example, you can't prevent someone from taking a digital photo of a document displayed on a screen and then e-mailing the picture or retyping the contents into a new document.



My Permission

You are currently authenticated to view this document as:
maxhayden@ad.steveco.net

Change user...

Restricted Access - Permission is currently restricted.
Only specified users can access this content.

You have the following permissions:

View:	Yes
Edit:	No
Copy:	No
Print:	No
Save:	No
Access the document programmatically:	No
Full control:	No

Request additional permissions...

Figure 10-6: Viewing the information usage rights on a document protected with AD RMS.

Inside Active Directory Rights Management Services

AD RMS is a PKI-based technology that uses certificates and encryption to enforce the rights that the document creator or owner wants to enforce on other users who access the document. RMS was introduced as an add-on to Windows Server 2003; in Windows Server 2008, RMS has evolved into an integral component of Microsoft's overall security strategy as well as Windows Server 2008. With AD RMS, authors can decide exactly who can view, edit, print, save, and copy data.

WARNING!

Even though I'm discussing certificates here, AD CS is not a requirement for deploying AD RMS. The AD RMS server can get its server certificate from AD RMS but there's no requirement to do so. Note also that in earlier versions of RMS, the certificate had to be issued by Microsoft. This is no longer the case with AD RMS in Windows Server 2008.

Before I go into the mechanics behind how AD RMS works, I need to explain a few concepts. Earlier in this chapter, I provide definitions for symmetric and asymmetric encryption. Both of these types of key-based encryptions are used within AD RMS. But, to review, a *symmetric encryption* algorithm (see Figure 10-7) is an encryption routine designed to take a *key* (a random number typically between 64 and 512 bits in length) and encrypt the document by using the key. This is symmetric encryption because only the same key can be used to decrypt the document. The benefit of symmetric encryption is that the algorithms aren't that complicated; therefore, they can be executed very quickly. However, the security of the encryption is a major issue because both the sender and the receiver of the document must have secure access to the same key.

In *asymmetric encryption,* two different keys are involved in the process: one for encrypting the document and a different but related key for decrypting the document. The two keys are generated at the same time so that if a document is encrypted with one of the keys, only the second related key can decrypt the document. (See Figure 10-8.) Typically, these keys are referred to as a *private key* and a *public key*. A private key is one that is generated for a particular user and is never shared with any other user or computer. A public key is typically one of the pieces of data that's stored in a PKI certificate. Although this type of encryption creates a very secure way of sharing data, an added benefit is you knowing that a piece of data decrypted by using a particular user's public key must have come from that user because no other user would have the private key the document was encrypted with.

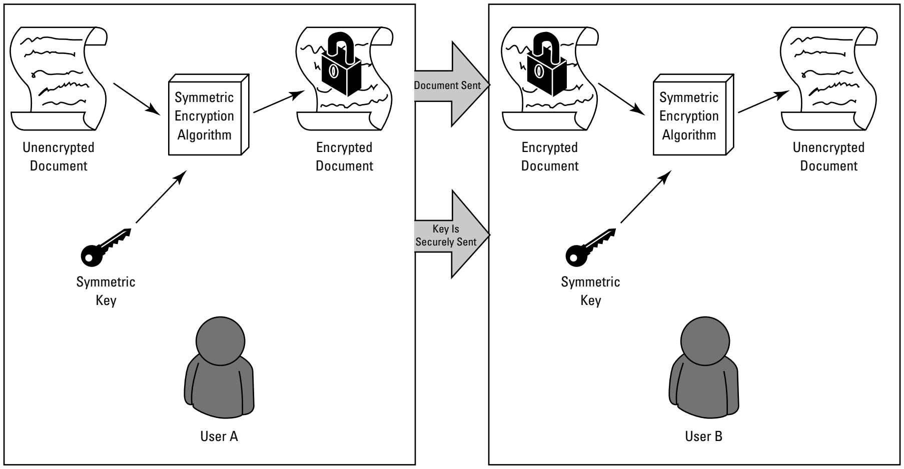


Figure 10-7: Symmetric encryption.

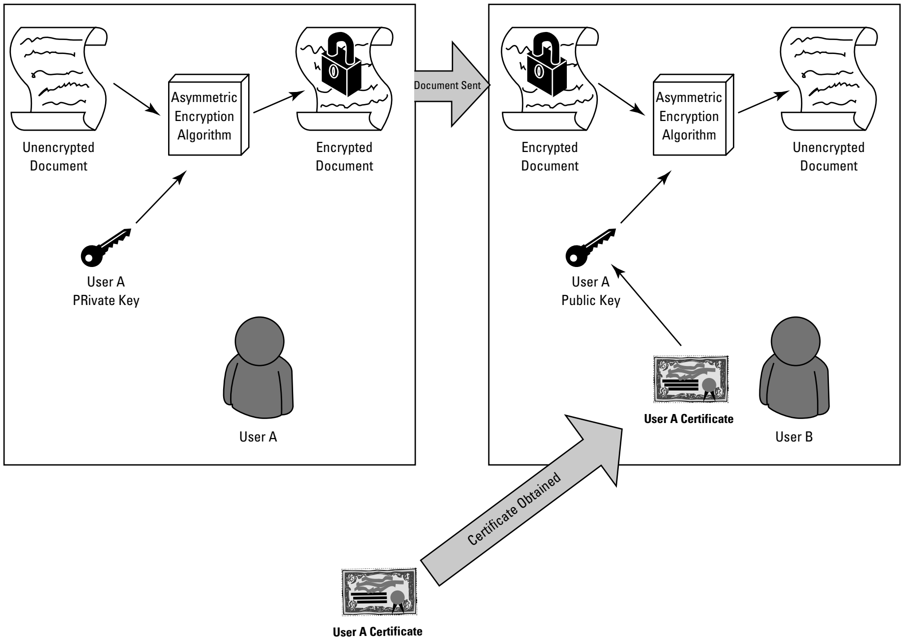


Figure 10-8: Asymmetric encryption.

Now that you understand these two types of encryption, you can see how they're used within AD RMS. The AD RMS server is a type of CA in that the server creates and issues certificates, but these certificates are for AD RMS use only. In addition to issuing certificates, the AD RMS also helps with the licensing of RMS-protected content. So, here's a look at the process of protecting a document in AD RMS. (See Figure 10-9.)

1. A document is created by using an application that's enabled to work with AD RMS. Examples of RMS-enabled applications include Office 2003, Office 2007, and Microsoft SharePoint Server.
2. The document author protects the document by using AD RMS. If this is the first time the author is using AD RMS, the author is issued a Rights Account Certificate (RAC) and a Client Licensor Certificate (CLC). The RAC is used to identify a user within the RMS environment and includes the user's public key. The associated private key is put into a protected store on the user's computer. The CLC contains a user public licensor key and the AD RMS server's public key. Additionally, a policy license is created that states what users may access the document and what level of access they have.
3. A random content key is created for the document to be protected. The document is encrypted with a symmetric encryption algorithm by using the content key. This means that to read the contents of the document, the user must have a copy of this content key.

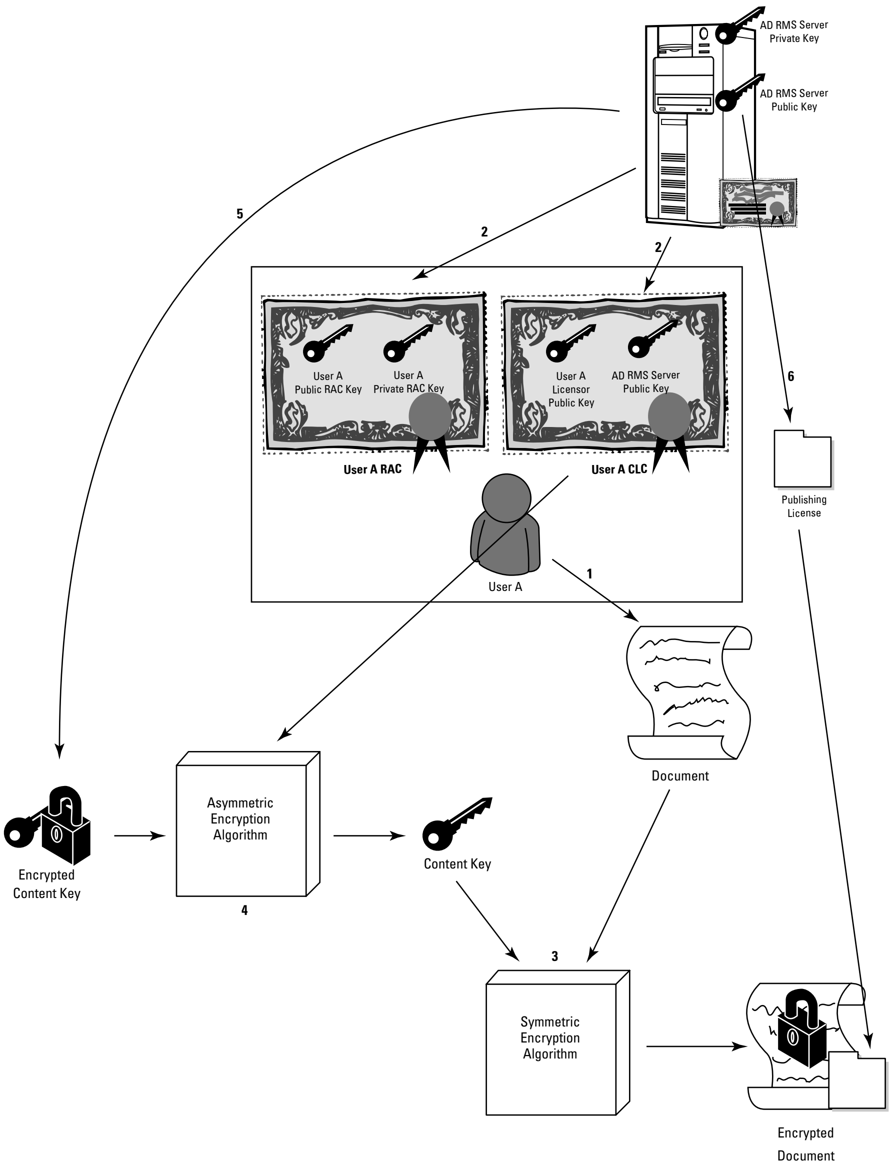


Figure 10-9: Protecting a document in AD RMS.

4. The content key is encrypted with asymmetric encryption by using the public key of the AD RMS server.

5. The encrypted content key and the policy license are sent to the AD RMS server. The server stores both items in a database (either locally or on a SQL server). The AD RMS server then creates a publishing license (PL) — a copy of the policy license and encrypted content key that's been signed by the AD RMS server. Anyone who reads the PL will know that it's valid because the AD RMS signed it.
6. The PL is sent back to the document author and is appended to the encrypted document.

   At this point, you have a document that's protected. This document can be shared with other users like any other file (via e-mail, file share, SharePoint, and so on).

The following covers what happens on the other end when an authorized user attempts to open the document (see Figure 10-10).

1. The user attempts to open the document. Any of the authentication/authorization necessary to open the file is done here. After file access has been granted, the RMS-aware application tries to open the file. The application recognizes that it's protected with AD RMS.
2. Like the first process, if the user opening the protected file has never used AD RMS before, an RAC and a CLC are issued to the user in the same way that the author received these certificates.
3. The application sends a request for a Use License (UL) to the AD RMS server (the URL of the AD RMS server is a part of the protected document). This request includes the user's RAC certificate and the appended PL in the document.
4. The AD RMS server validates the RAC and then examines the PL to determine whether the user may access the document. The AD RMS server takes the encrypted content key and decrypts it by using the AD RMS server's private key.
5. AD RMS then encrypts the content key by using the user's public key from the user's RAC. This encrypted key along with the rights that this specific user has to the document are put together as the UL and sent to the user.
6. The key in the UL is then decrypted with the user's private key.

   At this point, the document can be decrypted with the content key and opened. The application applies the required restrictions on the document as dictated by the UL.

That's basically how AD RMS works with an enabled application to enforce the usage rights defined by the document author. Keep in mind that this process is how AD RMS works with a user that has online access to the AD RMS. For offline access, the process works slightly different and makes more use of the keys in the CLC.

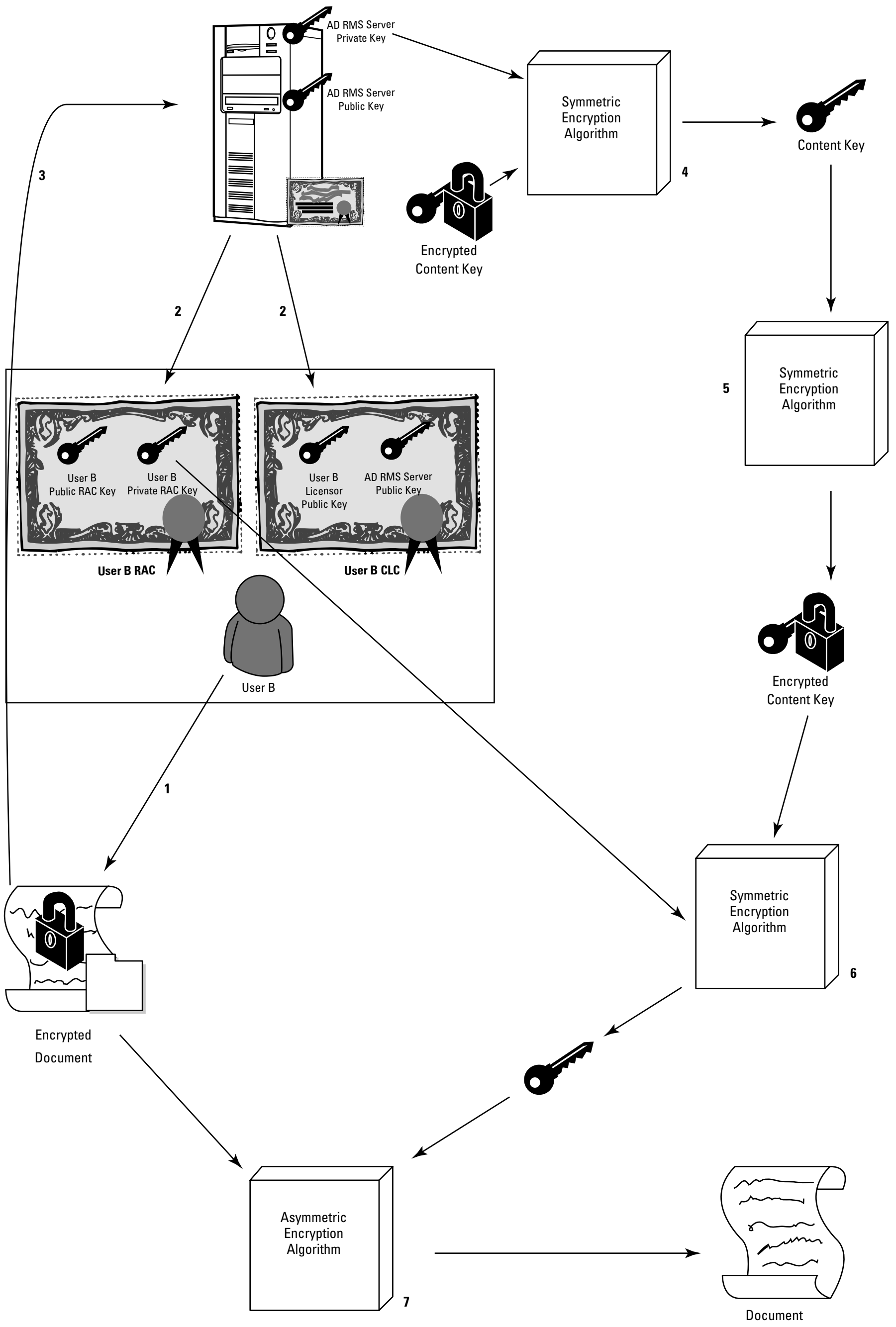


Figure 10-10: Opening a protected document in AD RMS.

Installing AD RMS

Like other AD components, AD RMS is an installable server role via Server Manager. A database store is required, and although you can use a local store, I strongly recommended you do that for test environments only. For production environments, you need to use a separate database server, such as Microsoft SQL Server 2005. ***Note:*** You should not deploy AD RMS on a domain controller because the AD RMS service requires a service account to run; if you run AD RMS on a domain controller, you must add the service account to the Domain Admins group, which basically gives the AD RMS service domainwide administration authority that isn't required.

On the client side, both Office 2003 and Office 2007 support AD RMS, but keep in mind that in Office 2007, only users running the Enterprise, Professional Plus, or Ultimate versions can actually author AD RMS-protected content. On the client operating system side, Windows Vista already comes with the AD RMS client installed. If you have clients running Windows XP, an RMS client download is available.

TIP

Although I don't have the room to cover it here, be aware that AD RMS not only works within a single internal company network but also supports AD FS so that you can protect documents made available through a federation. This requires only that AD RMS be set up in the account side of the federation.

TIP

More information about Active Directory Rights Management Services can be found in Microsoft's Windows Server 2008 TechCenter at:

```
http://technet.microsoft.com/windowsserver/2008
```

Part IV
Managing Active Directory

The 5th Wave By Rich Tennant

"It appears a server in Atlanta is about to go down, there's printer backup in Baltimore and an accountant in Chicago is about to make level 3 of the game 'Tomb Pirate.'"

In this part . . .

As much as we want to tell you that after you design and deploy Active Directory your job is done, that simply isn't the case. As the AD administrator, you have to deal with a lot of ongoing administration and maintenance tasks. That's where the chapters in this part come in. In this section we cover schema modifications, creating objects (such as users), managing the physical topology of Active Directory, administrating the security policies, and backing up and restoring the Active Directory database. The information in these chapters assists you with both day-to-day and advanced administrative tasks in Active Directory.

Chapter 11

Managing Users, Groups, and Other Objects

In This Chapter

- Creating and editing users
- Managing groups
- Working with organizational units
- Delegating administrative control

User and group administration is one of the more important aspects of system administration. After all, if the end-users can't access the appropriate network resources, then the network isn't functioning correctly, is it? In this chapter, I discuss creating and managing Active Directory objects including users, groups, and organizational units. I show you how to delegate administrative control over these objects, too.

Managing Users and Groups

Managing user and group objects is among the most typical day-to-day administrative tasks you perform in supporting Active Directory. In the following sections, I look at the various tools (both GUI-based and command line–based) that are available to create and manage these objects.

Creating user objects

Adding a user is a fairly easy process, and the GUI tool you use to create users is the Active Directory Users and Computers (ADUC) console. This console can be invoked in several ways. In Windows Server 2008, you can run the tool from the Server Manager program because ADUC is one of the available tools for managing the Active Directory Domain Services role (see Figure 11-1). You can also find the ADUC console under the Administrative Tools program group on the Start menu, or you can run the tool directly by entering **DSA.MSC** in the Run command.

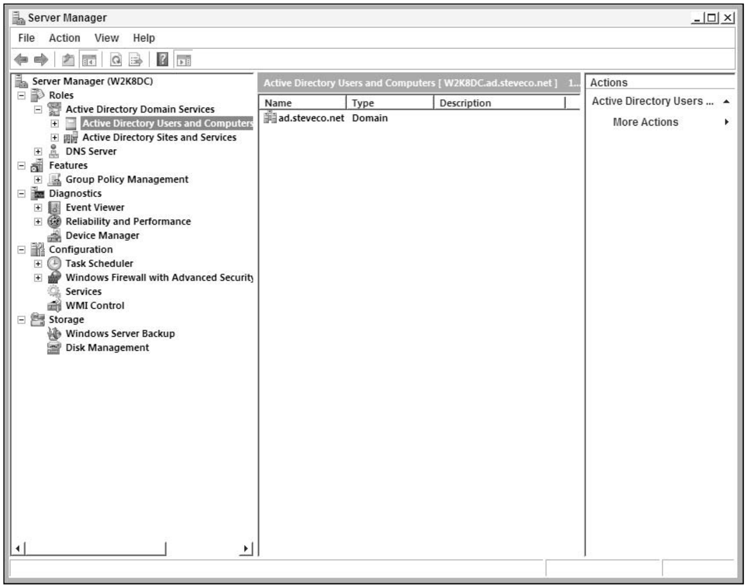


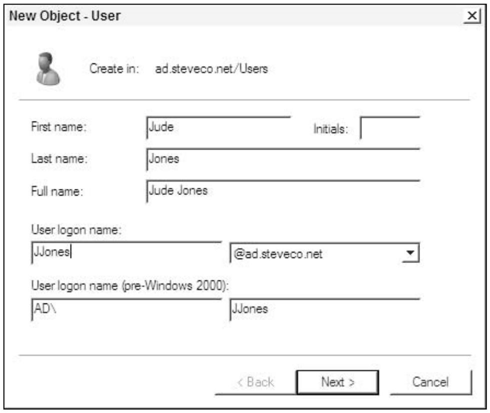
Figure 11-1: The Active Directory Users and Computers console in the Server Manager.

To add a user with ADUC, follow these steps:

1. **In the AD Users and Computers console, right-click the container into which you want to place the user.**
2. **From the toolbar, choose New➪User.**

   The New Object - User dialog box appears, as shown in Figure 11-2.
3. **Type the user information in the appropriate text boxes.**
4. **Click Next.**

   On the screen that appears, you specify the user’s password and password settings (see Figure 11-3).
5. **Type the user’s password, type it again to confirm the password, and choose the appropriate password settings.**



New Object - User

Create in: ad.steveco.net/Users

First name: Jude

Initials:

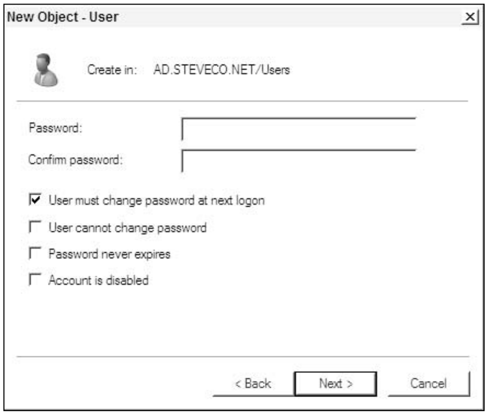
Last name: Jones

Full name: Jude Jones

User logon name: JJones @ad.steveco.net

User logon name (pre-Windows 2000): AD\ JJones

Figure 11-2: Adding a new user.



New Object - User

Create in: AD.STEVECO.NET/Users

Password:

Confirm password:

☑ User must change password at next logon

☐ User cannot change password

☐ Password never expires

☐ Account is disabled

Figure 11-3: Setting a user's password.

6. **Click Next.**

   On the Summary screen, review the user information you entered to ensure that everything is correct. (If you see a mistake, simply use the Back button to return to the correct screen and fix the mistake. Then use the Next button to return to the Summary screen.)

7. **Click Finish.**

REMEMBER

The ADSIEDIT console (ADSIEDIT.MSC) and LDP.EXE are two other GUI tools that technically can be used to create new objects, including user objects, in a directory. These tools aren't normally used to create new users in an AD DS forest, but for an AD Lightweight Directory Service directory, these are great GUI-based tools to manage objects. ***Note:*** You can't use ADUC to create or manage users in AD LDS.

Bulk administration tools

Even with commands like DSADD, you're still limited to executing separate commands for each object you want to create or modify. What if you need to create tens, hundreds, or even thousands of users? Fortunately, Microsoft has provided a couple of tools to help with this: LDIFDE and CSVDE. Both of these tools provide similar functionality but work by using two different types of data. LDIFDE utilizes the LDAP Data Interchange Format (LDIF) as specified in RFC 2849. The LDIF file is just a text file (that can be edited with Notepad) that has separate entries for each object you wish to create, modify, or delete. CSVDE works by using comma separated values (CSV) files. The great thing with using this tool is that you can create and modify CSV files by using a variety of programs including Microsoft Excel.

Both of these tools allow you to import and export data from either AD DS or AD LDS. The export functionality is particularly useful when you need to modify existing objects in the directory because you can export the directory information on those objects, modify the LDIF or CSV file, and then reimport the file into the directory.

ADUC is a great tool for when you want to create only one or two users. However, if you need to create a larger number of users, command line tools offer a way to create user objects in bulk. Also, if you're using a Windows Server 2008 Core Server, command line tools are the only tools available to you. The DSADD.EXE command can be used to create new objects — including users — in Active Directory.

TIP

More information about the various command line tools can be found in Microsoft's Windows Server 2008 TechCenter at:

```
http://technet.microsoft.com/windowsserver/2008
```

Editing user objects

With Active Directory, you're likely to spend a good deal of time modifying user attributes. Because the Active Directory database can hold such a wide variety of user information, you can use it for e-mail information, human resources information, emergency contact information, and even custom information that is unique for your company. Fortunately, you can edit all user information in one convenient location — Active Directory!

From the AD Users and Computers console, you can view the attributes on any user account by right-clicking the user account and choosing Properties. The Properties dialog box appears and offers you access to numerous attributes through the tabs at the top of the dialog box.

The General tab

As shown in Figure 11-4, each user account can contain a great many attributes. A user's account can be almost as individualized as her personality!

The information on the General tab requires little explanation. The Display Name is the name that shows up in the tree. The Other buttons next to the Telephone Number and Web Page fields enable you to enter multiple entries in those fields.

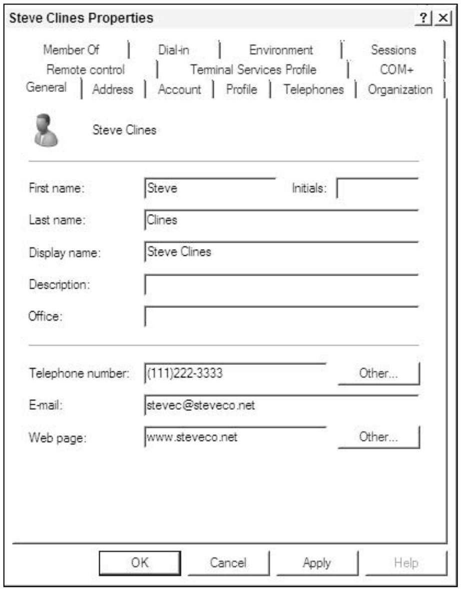
The Address tab

If you click the Address tab, your screen looks like the one shown in Figure 11-5. Again, the information is self-explanatory, but you really need to become very familiar with user attributes, so please take a brief look at the text boxes available on each tab.

The Account tab

The Account tab (see Figure 11-6) contains a bit more detail about this user object. Most of this information is self-explanatory, with the possible exception of the Logon Hours and Log On To buttons, which the following list describes:

- **Logon Hours:** Choose this button if, as a security measure, you want to restrict when a user can log on to the network.
- **Log On To:** Choose this button if you want to enable a user to log on to the network only from specific computers.



Steve Clines

First name: Steve Initials:

Last name: Clines

Display name: Steve Clines

Description:

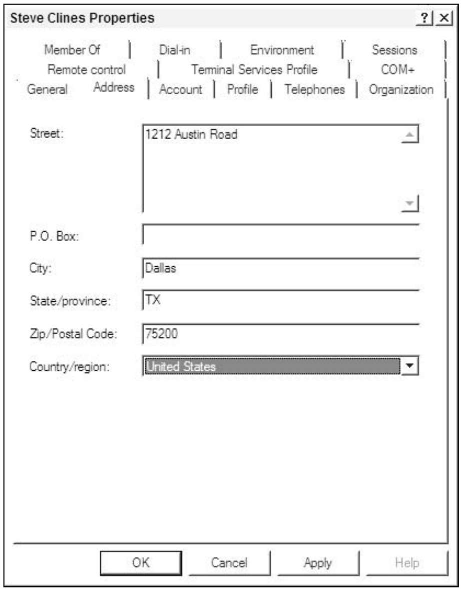
Office:

Telephone number: (111)222-3333

E-mail: stevec@steveco.net

Web page: www.steveco.net

Figure 11-4: The General tab of a user's Properties dialog box.



Street: 1212 Austin Road

P.O. Box:

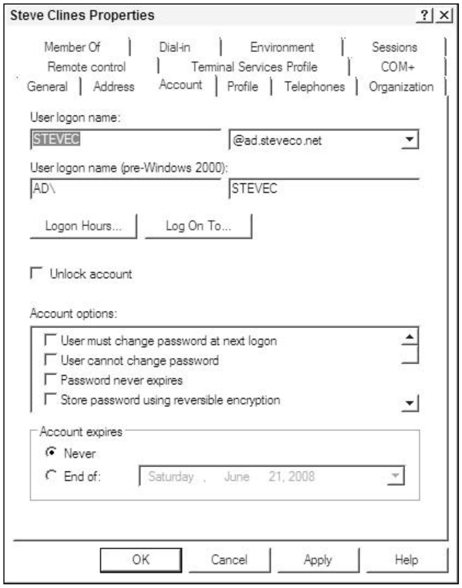
City: Dallas

State/province: TX

Zip/Postal Code: 75200

Country/region: United States

Figure 11-5: The Address tab.



User logon name: STEVEC @ad.steveco.net

User logon name (pre-Windows 2000): AD\ STEVEC

□ Unlock account

Account options:

- □ User must change password at next logon
- □ User cannot change password
- □ Password never expires
- □ Store password using reversible encryption

Account expires

- ● Never
- ○ End of: Saturday , June 21, 2008

Figure 11-6: The Account tab.

Although you can move among the tabs of the Properties dialog box by clicking each tab, remember to click the Apply button if you make any changes — before you move on to a different tab. Otherwise, when you're ready to close the Properties dialog box, you might forget that you made changes and forget to click OK.

The Profile tab

The Profile tab records the location of the user's profile, logon script, and home folder, as shown in Figure 11-7. The following list explains the options on this tab in more detail:

- **Profile path:** In this text box, type the local or network path to the user's profile.

  A *profile* is a group of settings that customizes the user environment, such as desktop settings. The user's preferred settings are stored in a profile so that each time the user logs on the settings take effect.

- **Logon script:** In this text box, type the local or network path to the logon script.

  When the user logs on to the network, the logon script runs and a variety of functions are performed, such as connecting to network shares or printers.

- **Local path:** If the database stores the user's home directory locally, instead of on a network server, type the path here.

- **Connect:** In this text box, specify a drive letter for the home folder.

  The home folder, or *home share,* is a disk location where the user can store her personal documents.

- **To:** In this text box, specify the network path to the home folder.

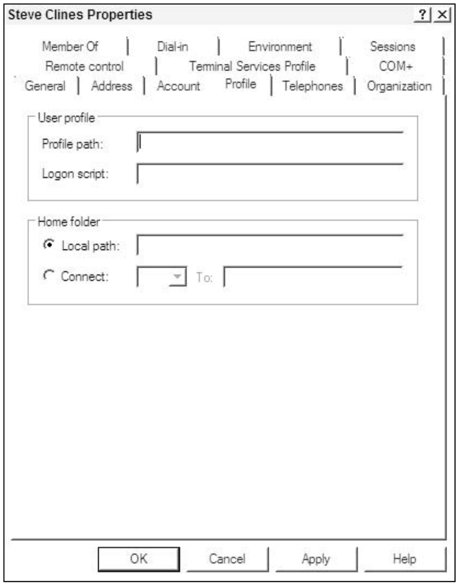


Figure 11-7: Enter the location of the profile, logon script, and home folder on the Profile tab.

The Telephones tab

The Telephones tab is completely self-explanatory (see Figure 11-8). Believe it or not, some users might need more fields for all of their phone numbers! Just remember that if you need to enter more than one telephone number for a field, click the Other button.

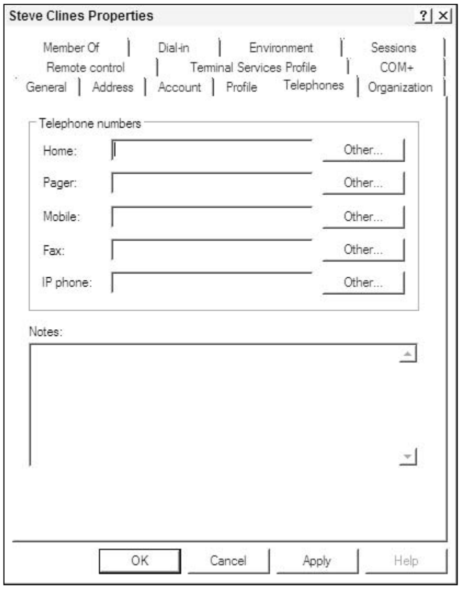


Figure 11-8: The Telephones tab.

The Organization tab

The Organization tab gives you an opportunity to store job-related information about each user, as shown in Figure 11-9. Most companies currently store much of this information in a variety of separate, unsynchronized databases. You might store job titles and managers in an HR database, for example, and store e-mail information and phone numbers in an Exchange database. Because Active Directory provides a single database for the entire organization, the Organization tab is a great way to take full advantage of Active Directory!

The Remote Control tab

If you're running Terminal Services, you also see the Remote Control tab in the Properties dialog box. Use this tab to configure the remote control settings for Terminal Services (see Figure 11-10). On this tab, you specify whether you can remotely control a user's Terminal Service session, whether you first need a user's permission to do so, and whether administrators can interact with the user session or simply view the user session.

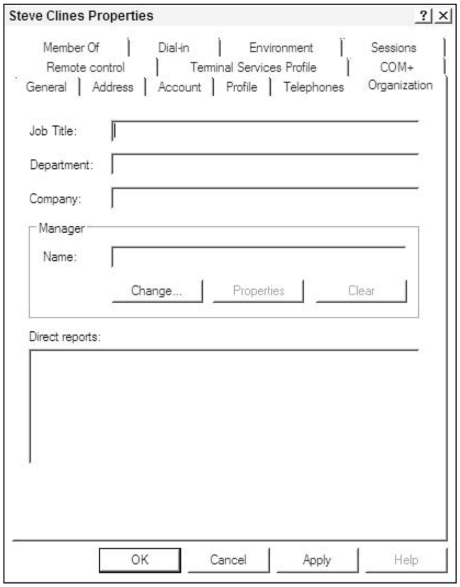
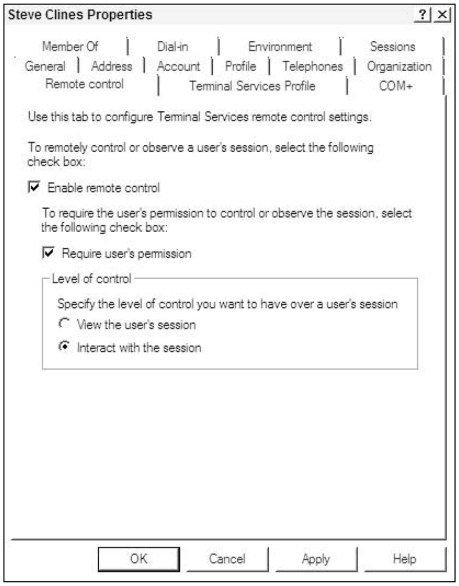


Figure 11-9: The Organization tab.



Use this tab to configure Terminal Services remote control settings.

To remotely control or observe a user's session, select the following check box:

☑ Enable remote control

To require the user's permission to control or observe the session, select the following check box:

☑ Require user's permission

Level of control

Specify the level of control you want to have over a user's session

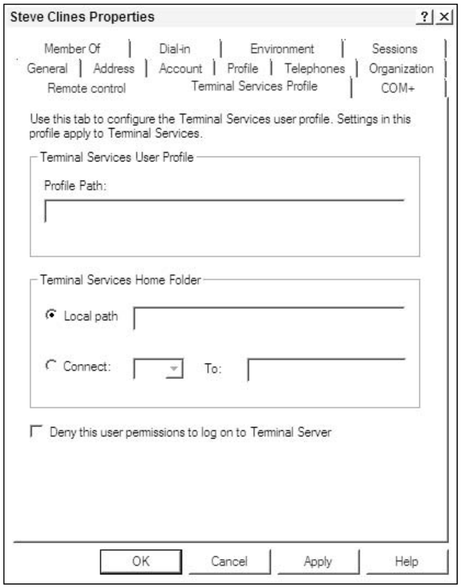
○ View the user's session

● Interact with the session

Figure 11-10: The Remote Control tab.

The Terminal Services Profile tab

The Terminal Services Profile tab, shown in Figure 11-11, is very similar to the Profile tab (refer to Figure 11-7). On this tab, however, the settings that you specify apply to Terminal Services. By selecting the check box from the Deny This User Permissions to log on to Terminal Services option, you prevent a user from using Terminal Services.



Use this tab to configure the Terminal Services user profile. Settings in this profile apply to Terminal Services.

Terminal Services User Profile

Profile Path:

Terminal Services Home Folder

● Local path

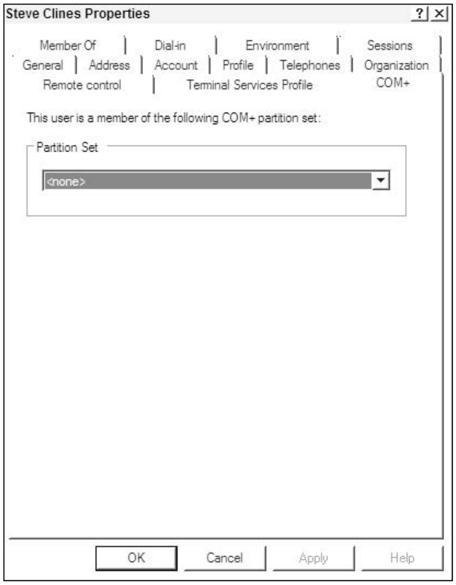
○ Connect: To:

□ Deny this user permissions to log on to Terminal Server

Figure 11-11: The Terminal Services Profile tab.

The COM+ tab

The COM+ tab, shown in Figure 11-12, provides a way to assign the user object to a particular COM+ partition set. So what is a COM+ partition set? COM+ (Common Object Model +), a Microsoft specification, is a programming interface that software creators can use to perform certain resource management tasks such as implementing multiprocessing in the software or implementing certain security processes. One of the things that COM+ provides is COM+ partitions. Imagine if you wanted to deploy multiple versions of the same software for different users but you wanted the software to be running on the same server. Normally this would be a difficult if not impossible task. COM+ provides partitions as a way of splitting up these different versions of the software so that they don't conflict. So that users are directed to the correct partition, you must assign the user to a particular partition. This tab in the User Object properties window provides an interface to assign the user to a particular partition.



This user is a member of the following COM+ partition set:

Partition Set

<none>

Figure 11-12: The COM+ tab.

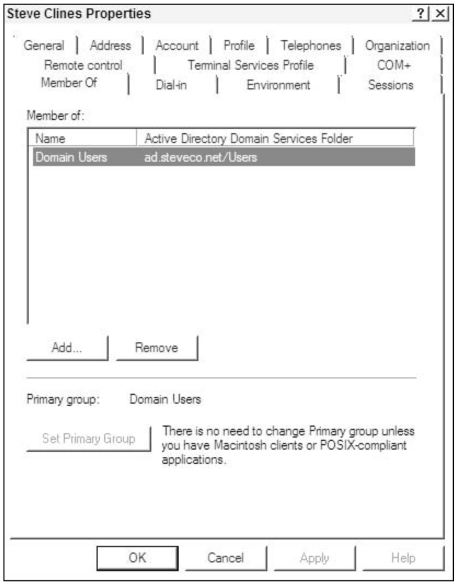
The Member Of tab

The Member Of tab lists which groups the user is a member of (see Figure 11-13). Use the Add and Remove buttons to quickly edit group memberships. Near the bottom of this tab is a Set Primary Group button. Every user has a primary group membership. If you specify no additional group memberships, the Primary Group, by default, is the Domain Users group.

The Dial-in tab

The Dial-in tab (see Figure 11-14) allows you to define some attributes of the user that relate to how he can remotely access this environment. The Routing and Remote Access service utilizes this information as well as Microsoft's Internet Authentication Service (IAS).

In Windows Server 2008, IAS has moved into the Network Policy and Access Services server role.



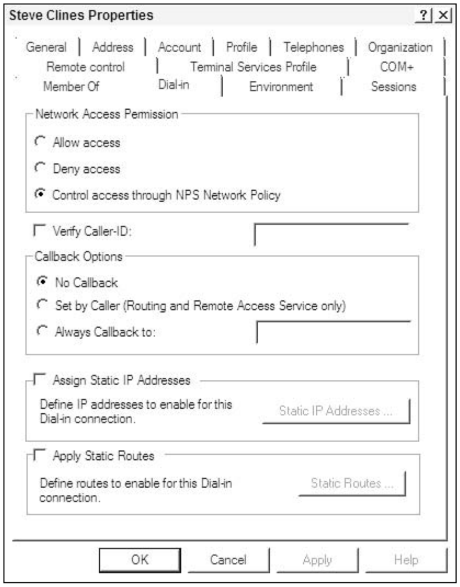
Member of:

Name	Active Directory Domain Services Folder
Domain Users	ad.steveco.net/Users

Primary group: Domain Users

There is no need to change Primary group unless you have Macintosh clients or POSIX-compliant applications.

Figure 11-13: The Member Of tab.



Network Access Permission

○ Allow access

○ Deny access

● Control access through NPS Network Policy

□ Verify Caller-ID:

Callback Options

● No Callback

○ Set by Caller (Routing and Remote Access Service only)

○ Always Callback to:

□ Assign Static IP Addresses

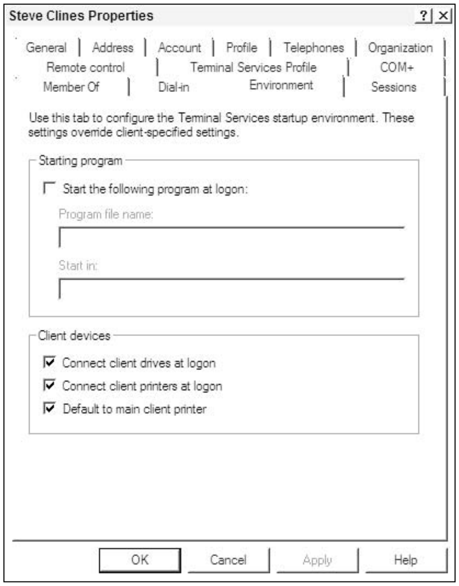
Define IP addresses to enable for this Dial-in connection.

□ Apply Static Routes

Define routes to enable for this Dial-in connection.

Figure 11-14: Configuring remote access parameters on the Dial-in tab.

The Environment tab (see Figure 11-15) is another tab that relates to Terminal Services. On this tab, you specify whether a program launches at startup and whether client-side drives and printers connect at logon.



Use this tab to configure the Terminal Services startup environment. These settings override client-specified settings.

Starting program

□ Start the following program at logon:

Program file name:

Start in:

Client devices

☑ Connect client drives at logon

☑ Connect client printers at logon

☑ Default to main client printer

Figure 11-15: The Environment tab.

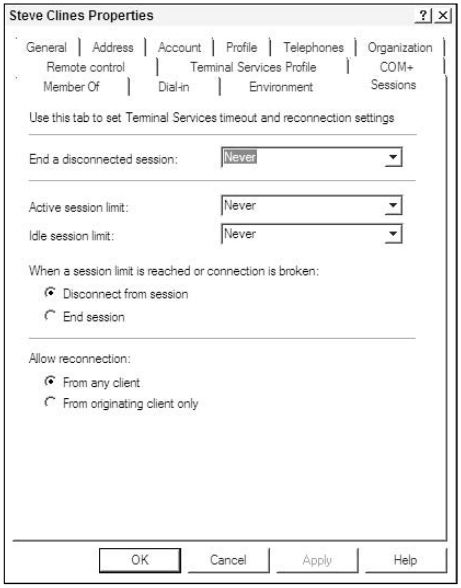
WARNING!

Be careful! All the settings you enable on this tab override user-specified settings on the client.

The Sessions tab

The Sessions tab (see Figure 11-16) also refers to Terminal Services. Use this tab to configure timeout limits and connection actions on the user's Terminal Services session.

In reviewing the user Properties tabs, I hope you notice that the more services you run in the domain, the more attributes that are available for each user object.



Use this tab to set Terminal Services timeout and reconnection settings

End a disconnected session: Never

Active session limit: Never

Idle session limit: Never

When a session limit is reached or connection is broken:

● Disconnect from session

○ End session

Allow reconnection:

● From any client

○ From originating client only

Figure 11-16: The Sessions tab controls Terminal Services timeouts.

Understanding groups

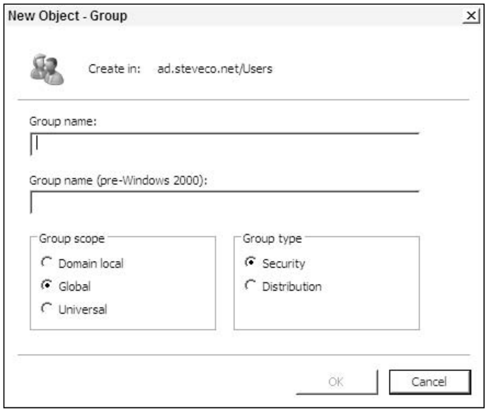
By creating groups and assigning users to these groups, administrators can manage large numbers of users simultaneously. Active Directory uses the following two types of groups:

- **Security groups:** These groups offer a means of gathering multiple users together and granting access to resources. System administrators can create a shared folder, for example, and then create a group that accesses the folder. Users who need access to the folder become members of the new group. You can also associate additional groups with varying access permissions with the shared folder. To one group, you might grant only Read permission to the folder. To another, you might offer Change permission. In large environments particularly, designating security groups is a much more efficient way to control user permissions than trying to manage individual users.
- **Distribution groups:** Distribution groups are used to group users together for non–security related purposes, such as sending e-mails. System administrators can, for example, create a distribution group for each division or department in an organization. By using these groups, you can target e-mail messages to specific subsets of users.

The other aspect of groups that you need to understand is their scope of usage. The scope controls the types of members that are placed into a particular group and within which AD forest domains the groups are available, as the following list explains:

- **Domain local groups:** Domain local groups are effective only within their local domain. You use them to grant permissions to resources within the domain, and administrators can view them only from within the specified domain.
- **Global groups:** Global groups grant permissions to a scope of trusted domains. You can view them anywhere within the tree. You can *nest* global groups — meaning that global groups can contain other global groups.
- **Universal groups:** Universal groups are effective and viewable across all domains in a forest. System administrators use universal groups to contain global groups. An administrator can, for example, create separate global distribution groups to contain user accounts of employees at two different branch offices in California. Then the administrator creates a universal distribution group for all employees in California. The two global groups become members of the universal group. Now the administrator can direct e-mails to employees at either of the branch offices or to all the employees in California.

Look at the New Object - Group dialog box shown in Figure 11-17. (See Chapter 6 for details on creating a new object.) In the bottom left of the dialog box, you specify the group scope (Domain local, Global, or Universal). To the right, you specify the group type (Security or Distribution).



New Object - Group

Create in: ad.steveco.net/Users

Group name:

Group name (pre-Windows 2000):

Group scope
○ Domain local
● Global
○ Universal

Group type
● Security
○ Distribution

Figure 11-17: Specifying the group scope and type in the New Object - Group dialog box.

Creating and editing groups

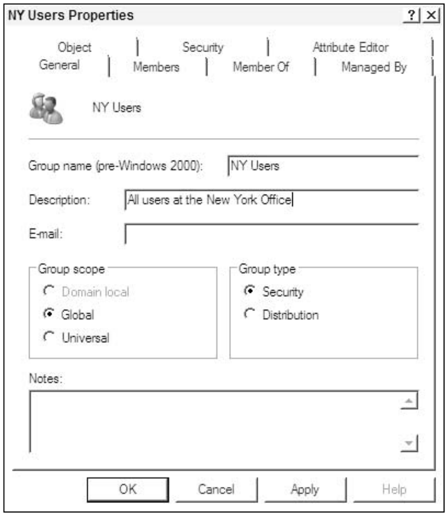
After you create a group, you can edit it by accessing the Properties page for that group. From the Active Directory Users and Computers console, right-click the user account and choose Properties from the contextual menu. The Properties dialog box appears and offers you access to numerous attributes through the tabs at the top of the dialog box.

Figure 11-18 shows the properties of a group I created that I call *NY Users*. The General tab of the Properties dialog box shows the information I selected when I created the group.

The Members tab, shown in Figure 11-19, lists users who belong to the *NY Users* group. You can use the Add and Remove buttons at the bottom of the tab to add or remove members of the group.

On the Member Of tab, shown in Figure 11-20, the Name column shows which groups the *NY Users* group is a member of. Again, you can edit directly from this tab by using the Add and Remove buttons.

The Managed By tab, shown in Figure 11-21, shows who has the assignment of managing this group. Someone must hold the final authority on who can gain access to a specific resource. The Managed By tab contains the contact information for the user who's the decision maker for this group.



NY Users

Group name (pre-Windows 2000): NY Users

Description: All users at the New York Office

E-mail:

Group scope

○ Domain local

● Global

○ Universal

Group type

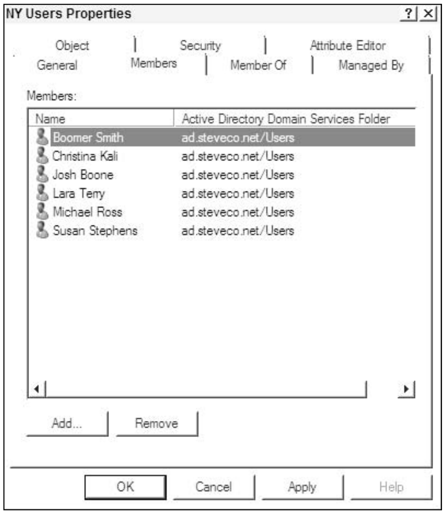
● Security

○ Distribution

Notes:

Figure 11-18: Viewing the General properties of the NY Users group.

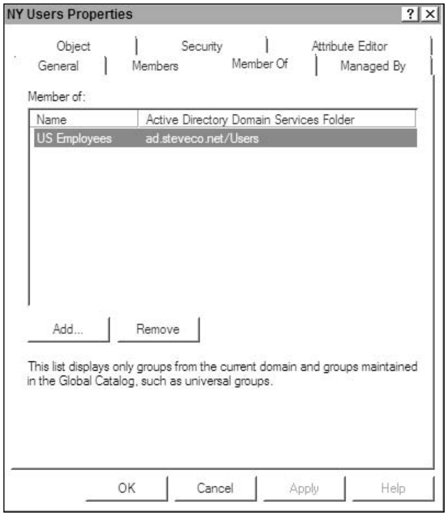
You can see from the Properties tabs for the NY Users group that managing and editing groups isn't a complicated process. You manage distribution groups and security groups in exactly the same way. Make sure that you specify the appropriate scope as you create the group because the scope determines how you can use the group and where you can view it.



Members:

Name	Active Directory Domain Services Folder
Boomer Smith	ad.steveco.net/Users
Christina Kali	ad.steveco.net/Users
Josh Boone	ad.steveco.net/Users
Lara Terry	ad.steveco.net/Users
Michael Ross	ad.steveco.net/Users
Susan Stephens	ad.steveco.net/Users

Figure 11-19: Members Tab of the New Object – Group screen.



Member of:

Name	Active Directory Domain Services Folder
US Employees	ad.steveco.net/Users

This list displays only groups from the current domain and groups maintained in the Global Catalog, such as universal groups.

Figure 11-20: Members Of Tab of the New Object – Group screen.

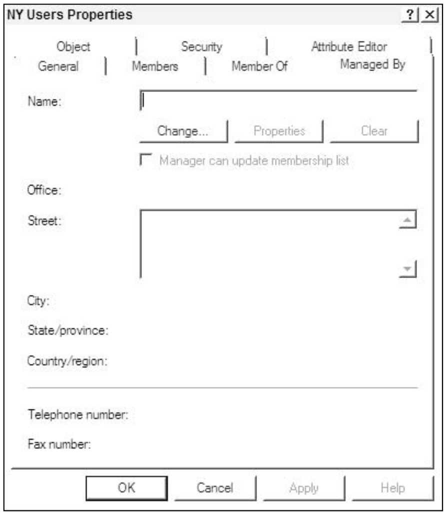


Figure 11-21: The Managed By tab.

Viewing default users and groups

When you install Active Directory, you create certain users and groups by default. Tables 11-1 and 11-2 list the default users and groups that are available.

Table 11-1 Default Users

Name	*Type*	*Description*
Administrator	User	Built-in account for administering the computer/domain.
Guest	User	Built-in account for guest access to the computer/domain.
krbtgt	User	Key Distribution Center Service Account.

Table 11-2 Default Groups

Name	*Type*	*Description*
Allowed RODC Password Replication Group	Security Group – Domain Local	Members of this group can have their passwords replicated to a read-only domain controller.
Cert Publishers	Security Group – Domain Local	Members of this group can publish PKI certificates to the directory.
Denied RODC Password Replication Group	Security Group – Domain Local	Members of this group cannot have their passwords replicated to a read-only domain controller.
RAS and IAS Servers	Security Group – Domain Local	Servers in this group can access remote access properties of users.
DnsAdmins	Security Group – Domain Local	DNS Administrators.
DnsUpdateProxy	Security Group – Global	Global DNS clients with permission to perform dynamic updates on behalf of some other clients (such as DHCP servers).
Domain Admins	Security Group – Global	Designated administrators of the domain.
Domain Computers	Security Group – Global	All workstations and servers that join to the domain.
Domain Controllers	Security Group – Global	All domain controllers in the domain.
Domain Guests	Security Group – Global	All domain guests.

(continued)

Table 11-2 *(continued)*

Name	*Type*	*Description*
Group Policy Creator Owners	Security Group – Global	Members in this group can modify group policy for the domain.
Read-only Domain Controllers	Security Group – Global	Members of this group are read-only domain controllers in the domain.
Enterprise Admins	Security Group – Universal	Designated administrators of the enterprise.
Enterprise Read-only Domain Controllers	Security Group – Universal	Members of this group are read-only domain controllers in the enterprise.
Schema Admins	Security Group – Universal	Designated administrators of the schema.
Account Operators	Security Group – Built-in Local	Members can administer domain user and group accounts.
Administrators	Security Group – Built-in Local	Administrators have full access to the computer and domain.
Backup Operators	Security Group – Built-in Local	Backup operators can only use a backup program to back up files and folders onto the computer.
Certificate Service DCOM Access	Security Group – Built-in Local	Members of this group are allowed to connect to Certification Authorities in the enterprise.
Cryptographic Operators	Security Group – Built-in Local	Members are allowed to perform cryptographic operations.

Name	*Type*	*Description*
Distributed DCOM Users	Security Group – Built-in Local	Members are allowed to launch, activate, and use DCOM objects on this machine.
Event Log Users	Security –Built-in Local	Members of this group can read from event logs on this machine.
Guests	Security Group – Built-in Local	Guests can operate the computer and save documents but can't install programs or make potentially damaging changes to the system files and settings.
IIS_IUSRS	Security Group – Built-in Local	Built-in group used by Internet Information Services.
Incoming Forest Trust Builders	Security Group – Built-in Local	Members of this group can create incoming, one-way trusts to this forest.
Network Configuration Operators	Security Group – Built-in Local	Members in this group can have some administrative privileges to manage configuration of networking features.
Performance Log Users	Security Group – Built-in Local	Members of this group may schedule logging of performance counters, enable trace providers, and collect event traces both locally and via remote access to this computer.

(continued)

Table 11-2 *(continued)*

Name	*Type*	*Description*
Print Operators	Security Group – Built-in Local	Members can administer domain printers.
Remote Desktop Users	Security Group – Built-in Local	Members in this group are granted the right to log on remotely.
Replicator	Security Group – Built-in Local	Supports file replication in a domain.
Server Operators	Security Group – Built-in Local	Members can administer domain servers.
Terminal Server License Servers	Security Group – Built-in Local	Members of this group can update user accounts in Active Directory with information about license issuance for the purpose of tracking and reporting TS Per User CAL usage.
Users	Security Group – Built-in Local	Users can operate the computer and save documents but can't install programs or make potentially damaging change to the system files and settings.
Windows Authorization Access Group	Security Group – Built-in Local	Members of this group have access to the computed `tokenGroupsGlobalAndUniversal` attribute on User Objects.

Managing Organizational Units

Remember, best practice dictates that you limit your trees to as few domains as possible. Organizational units offer a good alternative to domains, and in many cases, you can use them in place of child domains. Therefore, you might choose to have a tree that consists of only one domain. After you add all the domains to your tree that you want, you can start adding OUs. (Refer to Chapter 4 if you need a refresher on when to use OUs versus domains.)

Creating an OU is a fairly simple process. Follow these steps:

1. **Run the AD Users and Computers console.**
2. **In the left pane under AD Users and Computers, click the domain to which you want to add an OU.**
3. **Right-click the domain and choose New➪Organizational Unit from the contextual menu that appears (see Figure 11-22).**
4. **Type the name of the new OU in the Name text box of the New Object - Organizational Unit dialog box that appears (see Figure 11-23).**

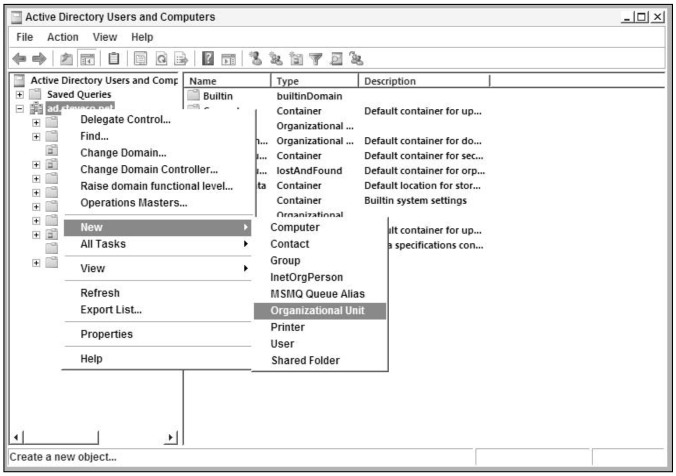
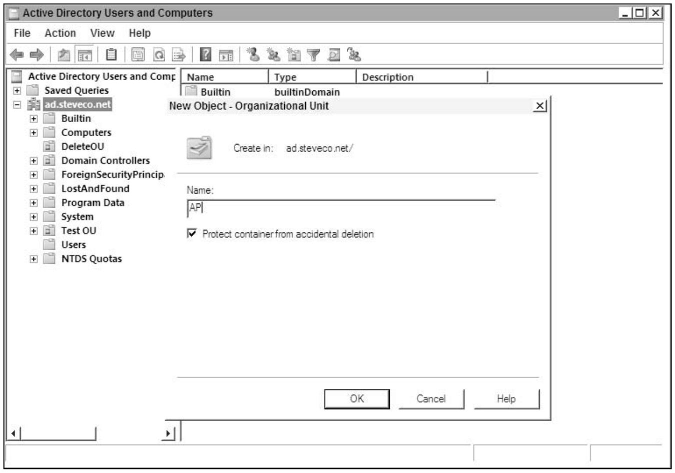


Figure 11-22: Adding an OU from the contextual menu.



Create in: ad.steveco.net/

Name:
AP

☑ Protect container from accidental deletion

Figure 11-23: Naming the new OU.

5. **Click OK to close the dialog box.**

That's all there is to it!

Delegating Administrative Control

In Chapter 5, you create both your forest and OU design to accommodate the administrative model. However, creating the forest and OU design is only half of the work of getting your administrative model implemented in Active Directory. The second part is delegating administrative control to those domains and OUs. Fortunately, Microsoft has created the Delegation of Control Wizard, which makes delegating permissions to implement that administrative model a snap.

To delegate administration of an Active Directory object, simply follow these steps:

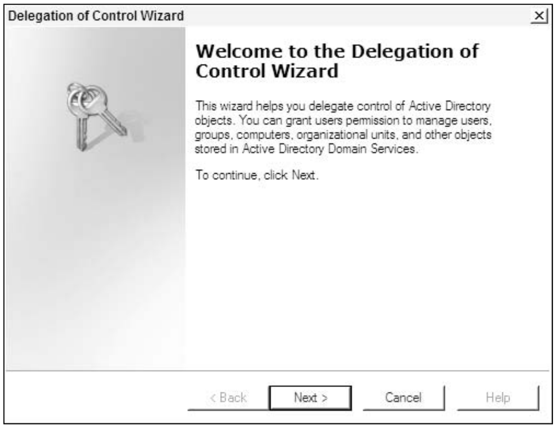
1. **From the Start menu, choose Programs➪Administrative Tools➪Active Directory Users and Computers.**

   The Active Directory Users and Computers console appears.

2. **In the Active Directory Users and Computers snap-in, right-click the object to which you want to delegate administration.**

3. **Choose Delegate Control from the contextual menu that appears.**

   No matter which object you're delegating control of, the procedure to start the wizard is always the same. Figure 11-24 shows the Welcome screen of the Delegation of Control Wizard.



Delegation of Control Wizard

Welcome to the Delegation of Control Wizard

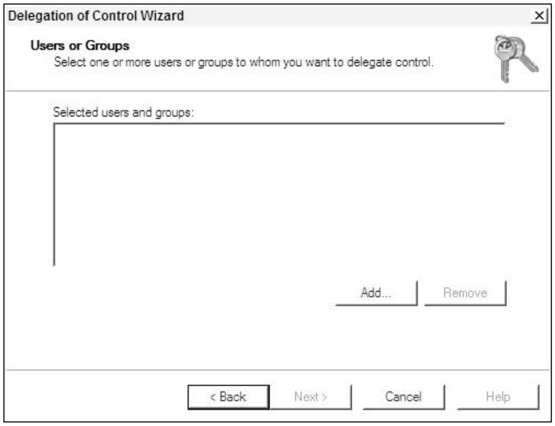
This wizard helps you delegate control of Active Directory objects. You can grant users permission to manage users, groups, computers, organizational units, and other objects stored in Active Directory Domain Services.

To continue, click Next.

Figure 11-24: The Delegation of Control Wizard Welcome screen.

4. **Click Next.**

   The Users or Groups screen appears. This screen is where you specify the groups or users to whom you're granting control of this OU, as shown in Figure 11-25.



Delegation of Control Wizard

Users or Groups

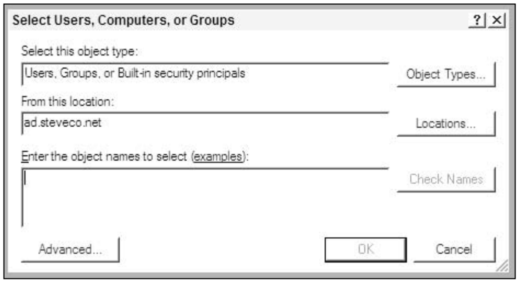
Select one or more users or groups to whom you want to delegate control.

Selected users and groups:

Figure 11-25: The Users or Groups screen.

5. **Click the Add button.**

   A dialog box that allows you to specify users or groups appears (see Figure 11-26).



Select Users, Computers, or Groups

Select this object type: Users, Groups, or Built-in security principals

From this location: ad.steveco.net

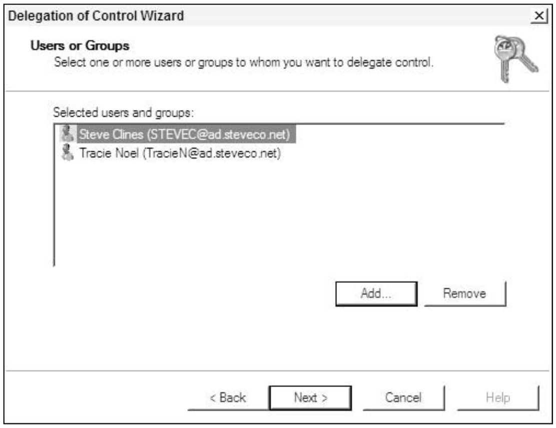
Enter the object names to select (examples):

Figure 11-26: Specifying the users and groups you're delegating control to.

6. **Click the users and groups to whom you want to delegate control of specific tasks.**

7. **Click OK.**

   You return to the Users or Groups screen, where your choices appear in the Selected Users and Groups list, as shown in Figure 11-27.



Delegation of Control Wizard

Users or Groups

Select one or more users or groups to whom you want to delegate control.

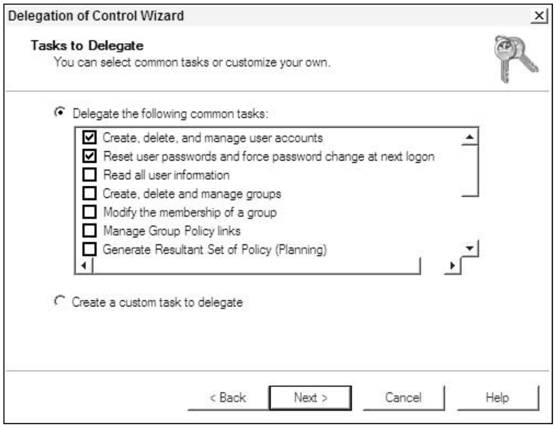
Selected users and groups:

- Steve Clines (STEVEC@ad.steveco.net)
- Tracie Noel (TracieN@ad.steveco.net)

Figure 11-27: The Users or Groups screen with your choices displayed.

8. **Click Next.**

   The Tasks to Delegate screen appears, as shown in Figure 11-28.



Delegation of Control Wizard

Tasks to Delegate

You can select common tasks or customize your own.

● Delegate the following common tasks:

☑ Create, delete, and manage user accounts
☑ Reset user passwords and force password change at next logon
☐ Read all user information
☐ Create, delete and manage groups
☐ Modify the membership of a group
☐ Manage Group Policy links
☐ Generate Resultant Set of Policy (Planning)

○ Create a custom task to delegate

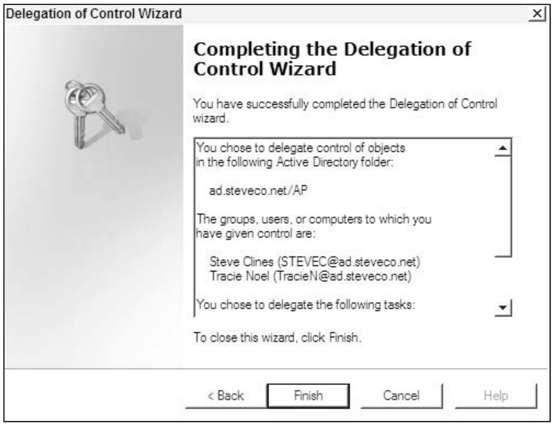
Figure 11-28: Delegating the administration of specific tasks.

On this screen, you choose which specific tasks to delegate. I'm not going to review each of the tasks that you can delegate because they're pretty straightforward. Figure 11-28 shows that I'm choosing to delegate the capability to create, delete, and manage user accounts as well as to reset passwords. Although I'm granting a user the capability to manage accounts in this OU, I'm not giving full administrative privileges to the OU nor granting privileges outside this particular container.

You can also select the second option and create custom tasks to delegate. You select this option, for example, when you want a specific user to run scripts within this OU.

9. **Click Next.**

   The final screen of the wizard appears, as shown in Figure 11-29. This screen summarizes your selections.



Completing the Delegation of Control Wizard

You have successfully completed the Delegation of Control wizard.

You chose to delegate control of objects in the following Active Directory folder:

ad.steveco.net/AP

The groups, users, or computers to which you have given control are:

Steve Clines (STEVEC@ad.steveco.net)
Tracie Noel (TracieN@ad.steveco.net)

You chose to delegate the following tasks:

To close this wizard, click Finish.

Figure 11-29: The Delegation of Control Wizard summarizes your selections.

10. If the information is correct, click Finish. If you need to correct any information, click the Back button and make the changes.

Delegating control isn't difficult, but it's something to plan carefully. If you use it correctly, delegating certain capabilities can free system administrators from a multitude of small (dare I say menial?) tasks. You can give end-users administrative control over their shared data without jeopardizing the integrity of the tree or the security of other users. Delegating administrative control is also a great way to decentralize administration by assigning authority over specific branches of the tree. Follow your organization's administrative model to determine how to delegate administration within the Active Directory tree.

Chapter 12

Managing Active Directory Replication

In This Chapter

- Exploring intrasite and intersite replication
- Putting your site topology to work
- Creating sites and subnets
- Creating site links and site link bridges

In a large environment or in an environment with slow wide area network (WAN) links, replication traffic can make or break a network. Replication is the periodic exchange of database information between the domain controllers within a domain, which ensures that all domain controllers contain updated, consistent data. Your job as a system or network administrator is to plan and control replication traffic. Excessive replication traffic can saturate a network, resulting in slow response times and application timeouts. Fortunately, Microsoft provides tools for the job!

Aside from controlling replication traffic, sites group computers for fast and efficient authentication. When a user logs on to a workstation, for example, a domain controller (DC) authenticates user ID and password.Instead of sending the authentication request across a WAN link for processing, the DC compares the IP address of the workstation with the subnets associated with each site. It then sends the authentication request to a domain controller in the local site.

In this chapter, I explain implementing a site topology, using sites and site links to control replication, and the difference between intrasite and intersite replication. (For more information on planning a site topology to control authentication and replication traffic, see Chapter 6.)

Understanding Replication

Before I show you the tools for managing replication, I should probably tell you a bit more about it. As I mentioned before, replication is an exchange of Active Directory (AD) information among domain controllers. Active Directory uses *multimaster replication,* which means that any domain controller can respond to service requests and record updates to the directory.

Active Directory uses site information to make replication and authentication more efficient. By grouping computers into sites, local domain controllers respond to logon requests. Sites also make replication traffic on the network more efficient. Replication can occur in two forms: intrasite and intersite.

Intrasite replication

Directory updates between DCs in the same site is *intrasite replication* (see Figure 12-1). Domain controllers within a site exchange information more often and more efficiently than do domain controllers in different sites. These frequent updates keep information fresh within a site. Domain controllers within a site are more likely to need fresh information about resources within the site. Updates are made less frequently to domain controllers outside the site because those domain controllers are less likely to need up-to-the-minute directory updates.

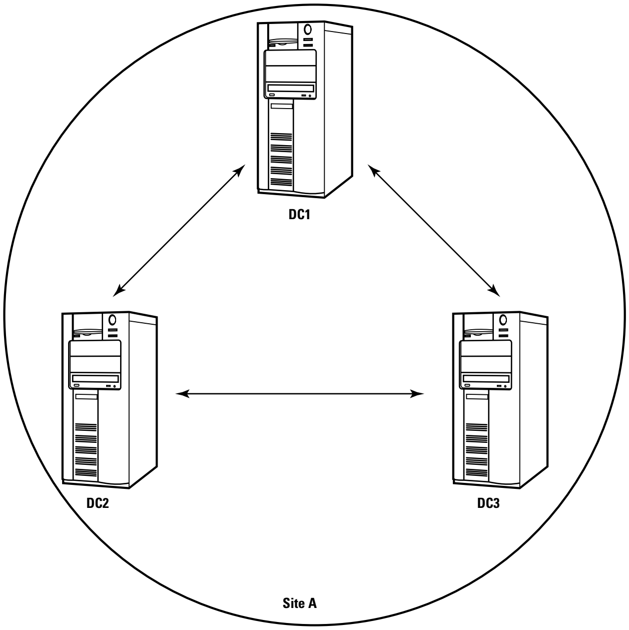


Figure 12-1: Intrasite replication occurs between domain controllers in the same site.

These frequent intrasite updates are why sites play such an important role in network traffic flow. Sites allow you to physically group computers that need to share resource information. Because these DCs exchange information frequently, a site should be limited to computers connected by fast links (such as local area network, or LAN, segments) instead of by slower WAN links. If you create your sites correctly, you optimize network traffic flow.

Active Directory automatically configures the replication topology between domain controllers in the same site, and AD also automatically configures intrasite connections and optimizes replication patterns across these sites.

This automatic replication topology, which provides multiple routes to each domain controller, is *fault tolerance.* Fault tolerance is important: If one DC is unavailable, the unavailable DC doesn't block replication to other DCs. The directory updates simply follow another path to reach all DCs within the site. If you add a new DC to the site, AD automatically adjusts the intrasite replication topology to include the new DC.

The DCs don't compress directory updates before sending them within a site like they do in intersite replication. The CPU cycles on the local DCs, therefore, remain free to service requests instead of compressing data.

Intersite replication

Intersite replication — replication between DCs in different sites — works differently than intrasite replication. In Figure 12-2, the intersite links are represented by the arrows between sites. (Comparatively, intrasite links are the arrows within the sites.) Here is where physical-structure diagrams of your network become invaluable. (See Chapter 6 to find out how to create these diagrams if you haven't already created them.) First, AD doesn't automatically create the links between sites. You create the site links based on the actual network links in your environment. I discuss how to create and configure site links in the section "Creating site links," a bit later in this chapter. For now, I just want you to understand how replication works across the links. Without site links, directory updates don't replicate to other sites.

After your site links are in place, you assign a *cost* to each site link. Doing so helps AD determine which site link to make a *primary route* between sites (to which you assign a low cost, such as 1) and also which to make a *secondary route* between sites (to which you assign a higher cost, such as 100). (Again, see the section "Creating site links," later in this chapter.) Use the available bandwidth measurements from your diagram to determine which site links are best suited to carry primary replication traffic.

To avoid saturating a network link with replication traffic during busy times of the day, you can associate schedules with the site link that determine when you can use the link for traffic. Remember that replication between sites can occur less frequently than intrasite replication.

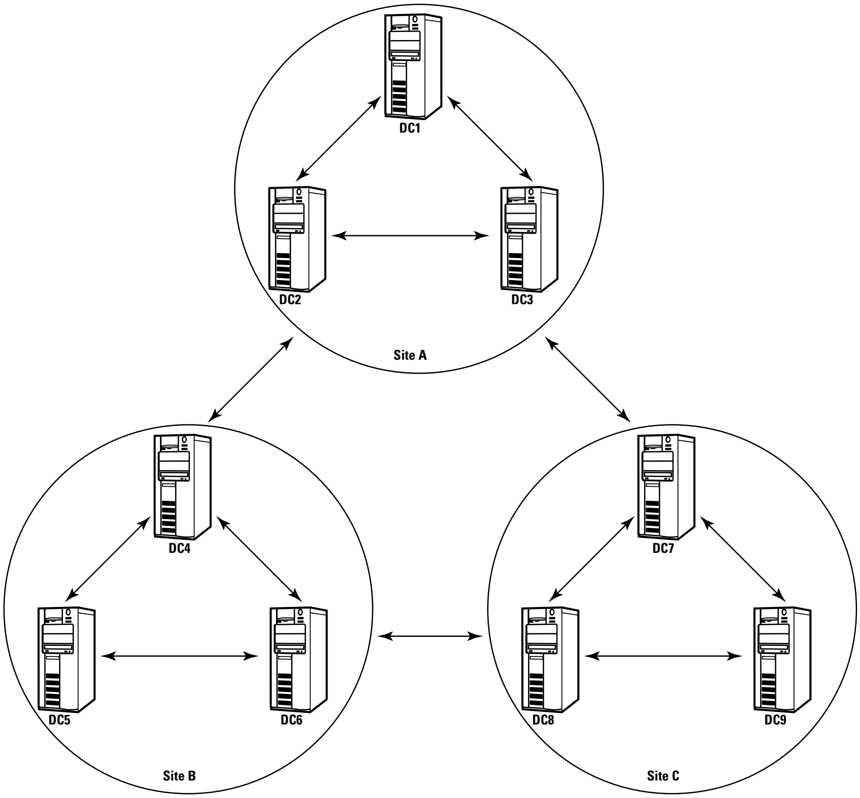


Figure 12-2: Intersite replication occurs between DCs in different sites.

Domain controllers typically compress intersite replication data before transmitting it. This process enables the data to transmit more rapidly. However, this compression occurs only when the amount of replication data to transmit benefits from the compression. For small pieces of replication data, the reduction in data from the compression/decompression process isn't worth the computing effort involved.

To optimize server resources, you can establish one server within a site as a *bridgehead server,* which handles all intersite directory replication. This server compresses all directory updates and replicates them to other sites. In turn, the bridgehead server receives directory updates from all other sites. After the bridgehead server receives the updates, it replicates those updates to the other DCs in the site.

Propagating updates

Propagating updates is just a fancy way of saying "replicating." I mention it only because you'll probably hear the phrase used in discussions about replication.

The updates are the changes you make to AD objects. Propagating the changes refers to copying the changes to all the other domain controllers in the tree. (I guess you sound more impressive saying, “The domain controllers are propagating updates throughout the domain” than “The domain controllers are replicating.” You don’t want to make your job sound too easy, do you?)

Implementing a Site Topology

As part of AD planning, you should create a physical structure design to accompany your design for a logical AD structure. (See Chapter 6 if you need help in creating the physical design. Chapter 5 helps you with the logical design.) The physical structure associates the underlying network with the logical structure so that AD functions efficiently on the network.

Sites — the major component of the physical structure — are groupings of subnets that connect to one another via fast links (typically LAN links). All the sites in the physical structure connect via site links and site link bridges. (*Site link bridges* connect specific site links. For more information, see the section “Creating a site link bridge” near the end of this chapter.) Sites, subnets, site links, and site link bridges make up the site topology.

In Figure 12-3, you can see that my network requires four sites. To connect these sites, I must create site links and possibly create site link bridges. Then by using available bandwidth data, I can assign a cost to each site link and schedule replication traffic across the site links.

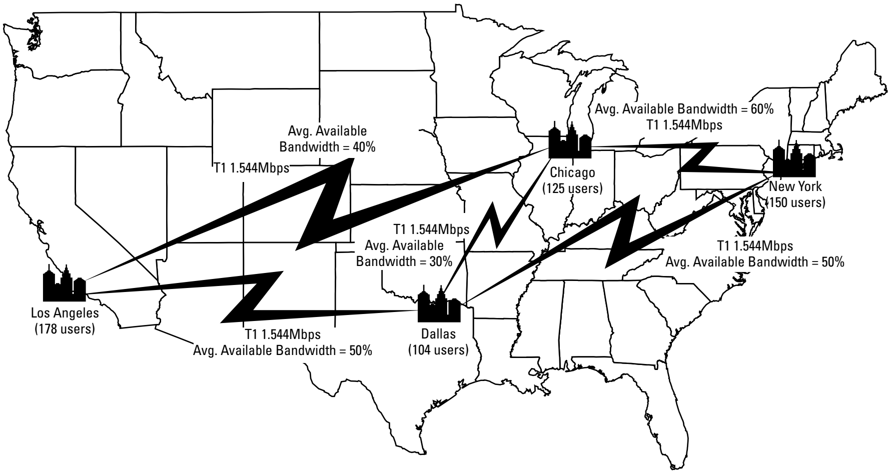

Avg. Available Bandwidth = 60%
T1 1.544Mbps
Avg. Available Bandwidth = 40%
T1 1.544Mbps
Chicago (125 users)
New York (150 users)
T1 1.544Mbps
Avg. Available Bandwidth = 30%
T1 1.544Mbps
Avg. Available Bandwidth = 50%
Los Angeles (178 users)
T1 1.544Mbps
Avg. Available Bandwidth = 50%
Dallas (104 users)

Figure 12-3: Use a site map to create the site topology.

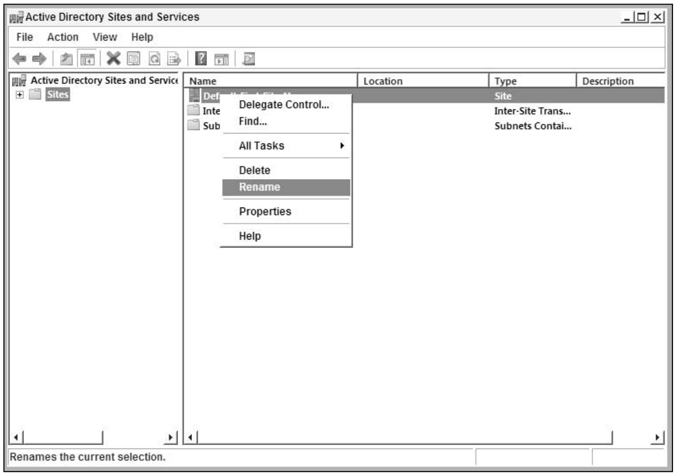
Creating sites

Creating sites is easy. *Configuring* the site links is what can getcha! Follow these steps to create a new site:

1. **Start the AD Sites and Services Manager by choosing Start➪Programs➪ Administrative Tools➪Active Directory Sites and Services Manager.**

   The AD Sites and Services screen opens.

   When you install AD, it automatically creates the first site for you and names it `Default-First-Site-Name`. Not a very original name, but at least it tells you how the site got there. Fortunately, you can rename this site quickly and easily in AD Sites and Services by right-clicking `Default-First-Site-Name` and choosing Rename from the contextual menu (see Figure 12-4). I'm working from the site map shown in Figure 12-3, so I'm renaming this site `Chicago`.

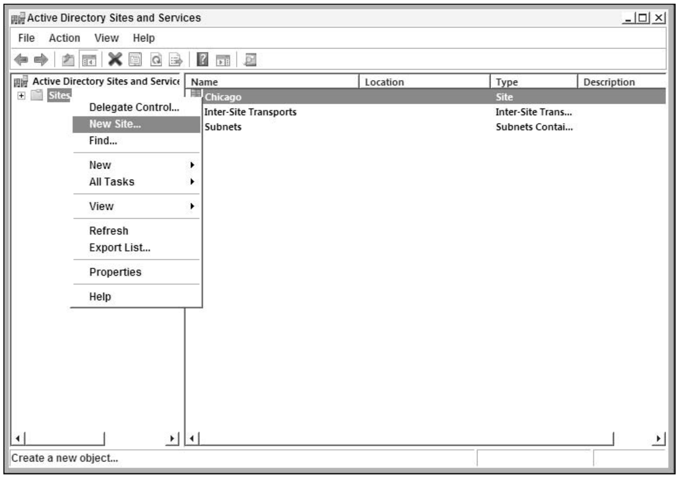


Renames the current selection.

Figure 12-4: Change the default name for the site.

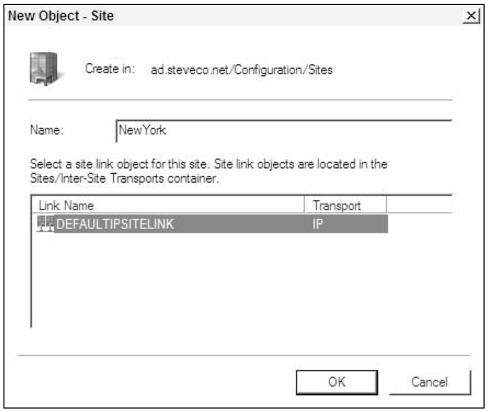
2. **Select the Sites container from the left pane of the AD Sites and Services screen.**
3. **Right-click and choose New Site from the contextual menu (see Figure 12-5).**
4. **In the New Object - Site dialog box that appears, type a name for the site in the Name text box (see Figure 12-6).**

   I'm still following my site map, so I'm naming this site `NewYork`.



Name	Location	Type	Description
Chicago		Site	
Inter-Site Transports		Inter-Site Trans...	
Subnets		Subnets Contai...	

Figure 12-5: Create a new site.



Create in: ad.steveco.net/Configuration/Sites

Name: NewYork

Select a site link object for this site. Site link objects are located in the Sites/Inter-Site Transports container.

Link Name	Transport
DEFAULTIPSITELINK	IP

Figure 12-6: Name the site and associate a site link object here.

5. **Choose DEFAULTIPSITELINK as the site link object.**

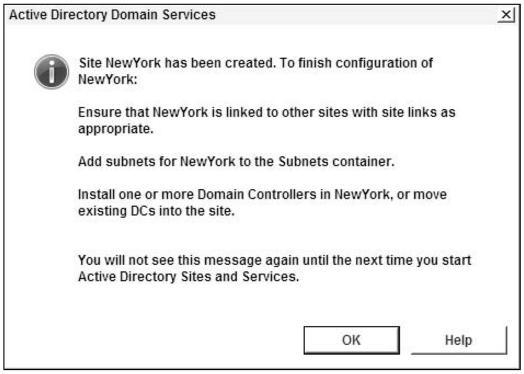
   No other site links exist yet.

6. **Click OK.**

   This part of the process is the part that I really like. A set of AD instructions appears that explains what you need to do to complete your site configuration (see Figure 12-7). Could things be any easier?

7. **After you read the instructions in the message box, click OK to close the box.**

The following sections explain how to do the tasks suggested in the dialog box.


Active Directory Domain Services

Site NewYork has been created. To finish configuration of NewYork:

Ensure that NewYork is linked to other sites with site links as appropriate.

Add subnets for NewYork to the Subnets container.

Install one or more Domain Controllers in NewYork, or move existing DCs into the site.

You will not see this message again until the next time you start Active Directory Sites and Services.

Figure 12-7: Active Directory tells you how to complete your site configuration.

Creating subnets

To identify the subnets that you associate with the site, follow these steps:

1. **On the AD Sites and Services screen, select the Subnets folder that appears beneath the Sites folder.**
2. **Right-click the Subnets folder and choose New Subnet from the contextual menu (see Figure 12-8).**

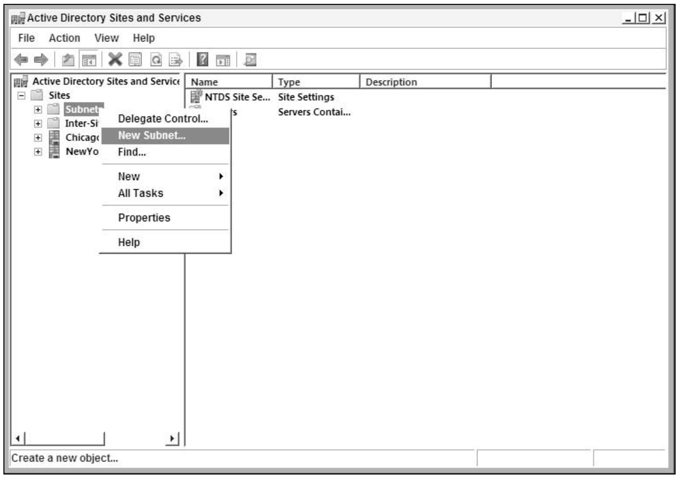
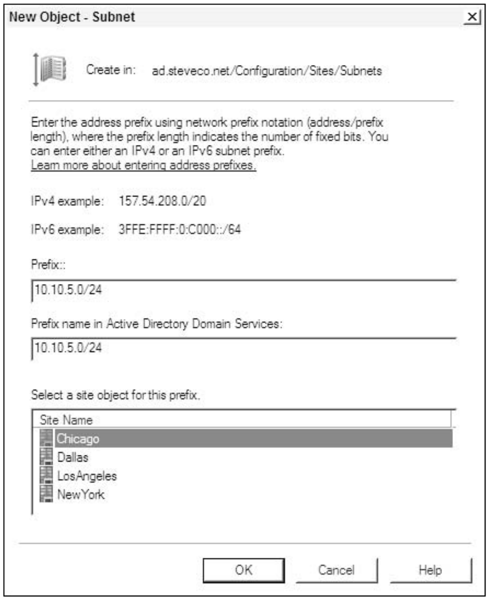


Figure 12-8: Create a new subnet.

3. **In the New Object - Subnet dialog box that appears, type the TCP/IP address prefix in the appropriate text boxes (see Figure 12-9).**

   An address prefix is a shortcut way of notating a TCP/IP subnet. A subnet in the TCP/IP world is defined by both a network address and a subnet mask. The network address is just an IP address. The subnet mask identifies what part of that network address is shared by all devices on the same subnet. In the address prefix number format, the subnet mask is written as the number of binary digits in the network address shared by all the devices on the subnet. So, for example, if you needed to supply the address prefix for the `192.168.10.0` network address where the first 24 binary digits in subnet number are shared by all devices on that subnet, the address prefix will be `192.168.10.0/24`.

Also keep in mind that I'm using only IPv4 addresses here. As I state in Chapter 1, Windows Server 2008 also provides support for IPv6 type addresses as well, so you can also create IPv6 type address prefixes for the subnet definition.



New Object - Subnet

Create in: ad.steveco.net/Configuration/Sites/Subnets

Enter the address prefix using network prefix notation (address/prefix length), where the prefix length indicates the number of fixed bits. You can enter either an IPv4 or an IPv6 subnet prefix.
Learn more about entering address prefixes.

IPv4 example: 157.54.208.0/20

IPv6 example: 3FFE:FFFF:0:C000::/64

Prefix::
10.10.5.0/24

Prefix name in Active Directory Domain Services:
10.10.5.0/24

Select a site object for this prefix.

Site Name
Chicago
Dallas
LosAngeles
NewYork

Figure 12-9: Associating the subnet with the correct site.

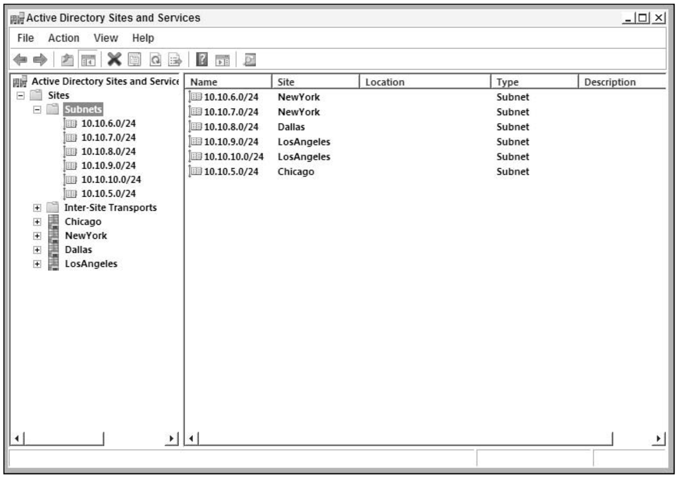
4. **Associate the TCP/IP address prefix with the correct site by choosing the appropriate name from the Site Name list box.**

   In my example, I associate the `10.10.5.0/24` address prefix with the Chicago site.

5. **Click OK to close the New Object - Subnet dialog box.**

I created all my subnets so that I can show you how the subnets appear in the AD Sites and Services Manager (see Figure 12-10).

Notice, too, that in the right-hand pane in Figure 12-10, each subnet lists the site with which it associates. In this example, you can see that each subnet corresponds to one of the four sites on my site map in Figure 12-3.



- Active Directory Sites and Service
  - Sites
    - Subnets
      - 10.10.6.0/24
      - 10.10.7.0/24
      - 10.10.8.0/24
      - 10.10.9.0/24
      - 10.10.10.0/24
      - 10.10.5.0/24
    - Inter-Site Transports
    - Chicago
    - NewYork
    - Dallas
    - LosAngeles

Name	Site	Location	Type	Description
10.10.6.0/24	NewYork		Subnet	
10.10.7.0/24	NewYork		Subnet	
10.10.8.0/24	Dallas		Subnet	
10.10.9.0/24	LosAngeles		Subnet	
10.10.10.0/24	LosAngeles		Subnet	
10.10.5.0/24	Chicago		Subnet	

Figure 12-10: The subnets list in the AD Sites and Services Manager.

Subnetting and subnet masks are beyond the scope of this book. For more information, refer to *TCP/IP For Dummies,* 5th Edition, by Candace Leiden, Marshall Wilensky, and Scott Bradner (Wiley).

Creating site links

Creating site links is only a bit more complicated than creating sites. After you create the site links, you schedule traffic and associate a cost with each site link. (Keep reading — I explain schedules and cost a bit later in this section!)

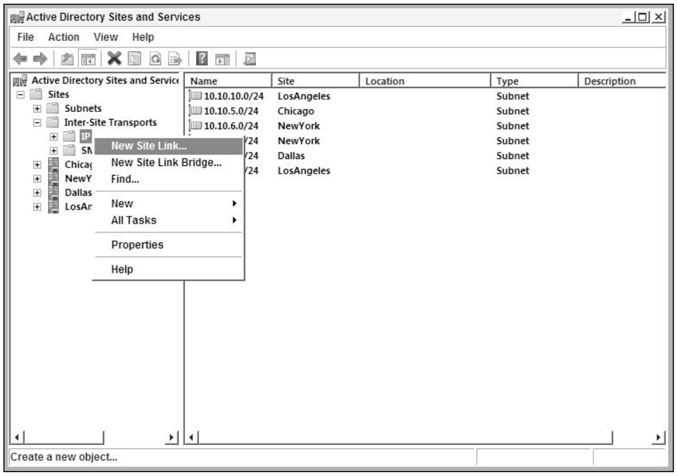
To create and configure a site link, you must specify the following items:

- The sites to be connected
- A transport protocol
- A schedule
- A cost

To create a site link, follow these steps:

1. **On the AD Sites and Services screen, click (or double-click, if necessary) to expand the Inter-Site Transports folder.**
2. **Right-click the IP folder and choose New Site Link from the contextual menu (see Figure 12-11).**

   You can also choose to create an SMTP site link. In this example, I use IP because I have T1 connectivity and ample available bandwidth.

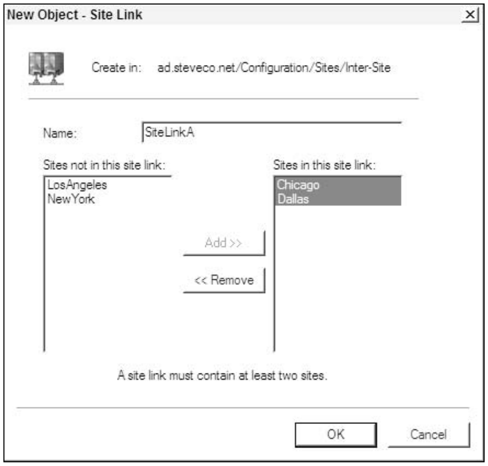


Name	Site	Location	Type	Description
10.10.10.0/24	LosAngeles		Subnet	
10.10.5.0/24	Chicago		Subnet	
10.10.6.0/24	NewYork		Subnet	
/24	NewYork		Subnet	
/24	Dallas		Subnet	
/24	LosAngeles		Subnet	

Create a new object...

Figure 12-11: Create a new site link.

3. **In the New Object - Site Link dialog box that appears (see Figure 12-12), type a name for the Site Link in the Name text box.**



Create in: ad.steveco.net/Configuration/Sites/Inter-Site

Name: SiteLinkA

Sites not in this site link:
LosAngeles
NewYork

Sites in this site link:
Chicago
Dallas

A site link must contain at least two sites.

Figure 12-12: Name the site link and associate sites with the site link.

4. **From the Sites Not in This Site Link box on the left, select each site that you want to connect with the new link and then click the Add button.**

   The site is moved to the Sites in This Site Link box on the right.

   Continue selecting sites and clicking the Add button after each one until all the sites you want included in this link are added to the box on the right.

5. **Click OK to close the New Object - Site Link dialog box.**

   You're finished! You just created your first site link.

TIP

As I mention in earlier parts of this book, site links are *transitive.* Thus, if you have a site link between Site A and Site B and another site link between Site B and Site C, the link between Site A and Site C also is a transitive link. Figure 12-13 illustrates this relationship.

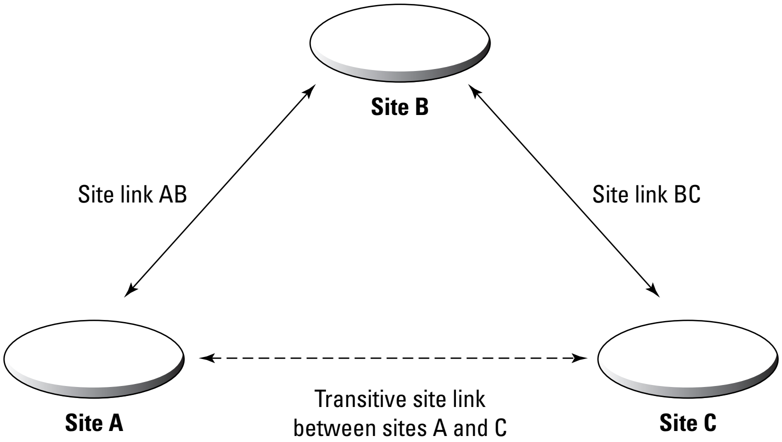


Figure 12-13: With site links between Sites A and B, and B and C, a transitive link exists between A and C.

Enabling transitive site links

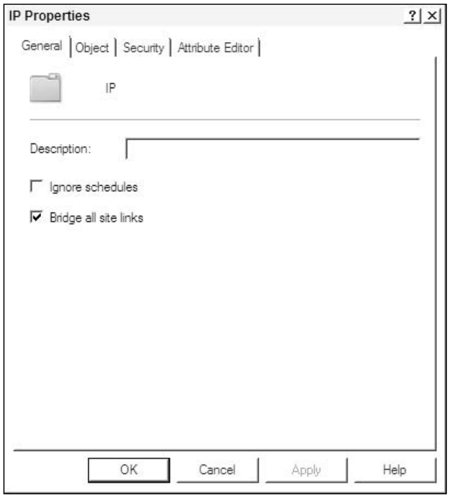
To enable transitive site links, enable the Bridge All Site Links option on the IP Inter-Site Transport, as I describe in the following steps:

1. **On the AD Sites and Services screen, click (or double-click, if necessary) to expand the Inter-Site Transports folder.**
2. **Right-click the IP folder and choose Properties from the contextual menu.**

   The IP Properties dialog box opens.
3. **On the General tab, select the Bridge All Site Links check box (see Figure 12-14).**
4. **Click OK to close the Properties dialog box.**

Creating a schedule

You can further optimize replication traffic on your network by scheduling the traffic to use a site link only at certain times of the day. For example, if more bandwidth is available during evening hours, you can schedule replication from 7–10 p.m. Or perhaps you want replication traffic to use one site link between 6 a.m. and 6 p.m., yet use another site link during the remaining hours.


IP Properties

General | Object | Security | Attribute Editor

IP

Description:

□ Ignore schedules

☑ Bridge all site links

OK | Cancel | Apply | Help

Figure 12-14: Enable transitive site links here.

When you restrict replication traffic to specific hours of the day, you're making a trade-off. You're sacrificing fresh directory content for efficient network traffic. If you can afford to have slightly stale directory data between sites until the scheduled replication occurs, you might choose to schedule intersite replication to occur infrequently.

To schedule replication traffic on a site link, follow these steps:

1. **On the AD Sites and Services screen, right-click the site link in the right-hand pane and choose Properties from the contextual menu.**

   The Properties dialog box opens.

2. **Click the Schedule tab.**

   This tab contains a weekly calendar. Use this calendar to block out times during which you want replication to be available or unavailable on this site link.

3. **After making your selections, click OK to close the Properties dialog box.**

Assigning a cost

As you will recall from Chapter 6, site link costs are used to indicate a preferred set of sites link to be used for AD replication. You might prefer to have replication traffic use one site link rather than another whenever possible. Perhaps you have a link that's heavily used, with little available bandwidth. You want to limit replication traffic on this link because of the bandwidth saturation. You have a second link with plenty of bandwidth available, so you prefer replication traffic to use this second link, but you'd like the first link to act as an alternate replication link if the preferred second link is unavailable.

To enable this scenario, you assign a cost to each of these links, placing a lower cost on the link that you would prefer to utilize, and assign a higher cost to the link don't want to use. You can use any value between 1–32,767 for the cost value. The values you choose are relative only to the other values you specify. For example, a site link with a cost of 1 is preferred to a site link at a cost of 20.

To assign a cost to a site link, follow these steps:

1. **On the AD Sites and Services screen, right-click the site link and choose Properties from the contextual menu.**

   The Properties dialog box opens.

2. **On the Cost tab, enter a cost value between 1 and 100.**

   A low value means that traffic frequently uses the site link. Conversely, the higher the value, the less frequently that traffic uses the site link.

3. **Click OK to close the Properties dialog box.**

Creating a site link bridge

Site link bridges are connectors between two site links. If you enable the Bridge All Site Links option (as I show you how to do in the section "Enabling transitive site links," earlier in this chapter), site link bridges are redundant. They provide the same function as transitive site links.

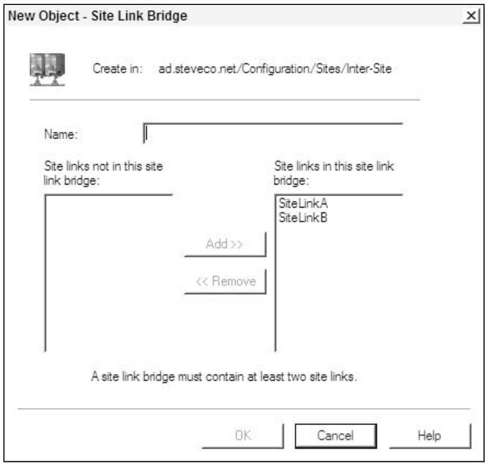
However, if you want total control over replication traffic patterns, you don't want to enable the Bridge All Site Links option. Instead, you need to create site link bridges between the site links. I don't recommend that you try this process unless you have slow links saturated with traffic and you want to closely control replication traffic. In most cases, this technique creates unnecessary work for the administrator and leaves a large margin for error. A better way to control replication traffic is to assign appropriate cost values to the site links.

A site link bridge essentially creates a replication path among available site links. The bridge groups the site links so that you can administer them as one object. For example, if you create a site link bridge between a Chicago-New York site link and a New York-Philadelphia site link, you essentially create a replication path between Chicago and Philadelphia. You can then assign costs and schedules to control use of this site link bridge.

To create a site link bridge, follow these steps:

1. **On the AD Sites and Services screen, click (or double-click, if necessary) to expand the Inter-Site Transports folder.**
2. **Right-click the IP folder and choose New Site Link Bridge from the contextual menu.**

   The New Object - Site Link Bridge dialog box opens, as shown in Figure 12-15.
3. **Type a name for the Site Link Bridge in the Name text box.**
4. **In the Site Links Not in This Site Link Bridge box on the left, select the site links connected by this site link bridge; then click the Add button to move these site links to the Site Links in This Site Link Bridge box on the right.**
5. **After making your selections, click OK to finish creating the site link bridge.**



Create in: ad.steveco.net/Configuration/Sites/Inter-Site

Name:

Site links not in this site link bridge:

Site links in this site link bridge:
SiteLinkA
SiteLinkB

A site link bridge must contain at least two site links.

Figure 12-15: Create a new site link bridge in this dialog box.

WARNING!

Again, I don't recommend that you create site link bridges. Because site links are transitive, replication paths are already in place. If you configure the site link bridge in a manner that conflicts with the configured cost values and schedules for its site links, you could cause replication between sites to fail.

Chapter 13

Schema-ing!

In This Chapter

- Understanding the Active Directory schema
- Identifying schema objects
- Accessing schema objects
- Extending the schema
- Exploring the schema master

As you may already know, the Active Directory schema contains definitions of all object classes (or object categories) and attributes that you can store in the directory. Make no mistake about it — understanding the schema is vital to understanding and managing Active Directory!

Schema 101

The Active Directory schema is part of the Active Directory database. If you remember from earlier in the book, the AD database is split up into several partitions with the schema being one of the partitions. The schema is all about Active Directory objects. A *schema* is a set of definitions that describe Active Directory objects and the objects' descriptive attributes. These definitions serve as rules, or templates, that dictate how you must describe an object. You have the following two categories of schema definitions:

- Classes
- Attributes

Object classes and object attributes are known as *schema objects,* or sometimes as *metadata*.

Introducing object classes

An *object class* is a set of mandatory attributes and optional attributes that combine to define a particular class of Active Directory objects (see Figure 13-1). A user is one object class and a printer is another. Obviously, you can't describe these vastly different objects by using the same set of attributes. So the user object class consists of different mandatory and optional attributes than does the printer object class.

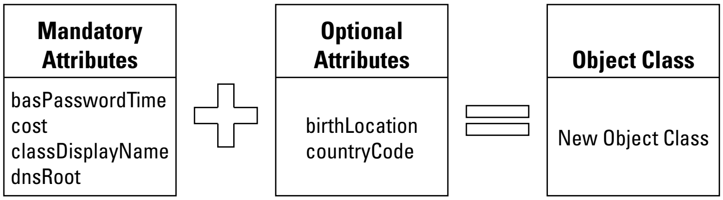


Figure 13-1: Mandatory and optional attributes combine to create object classes.

An *attribute* provides information about an object. Each attribute provides information about a different aspect of an object and uses a certain structure and syntax. Attributes combine in various groupings to form object classes.

The object classes and attributes that the schema defines take effect across the entire Active Directory tree or forest. This way, all similar objects in a forest conform to the same conventions. For example, all user objects have the same attributes, although the values of the attributes differ. Similarly, all shared folders are defined by the same attributes, but the values of those attributes differ. The schema itself resides within the Active Directory. As you initially install Active Directory, it automatically installs a base set of schema objects and attributes. (The objects and attributes are read into the Active Directory database from the `schema.ini` file in the `%systemroot%\system32` folder.) The schema is actually part of the `NTDS.DIT` file.

You can view the base schema by using the Active Directory Schema snap-in of the Microsoft Management Console. You must be a member of the Schema Administrators group to make changes to the schema. (For more information on the Schema Administrators group, see Chapter 1.)

You have to register the `schmmgmt.dll` before you can use the Active Directory Schema snap-in. The `regsvr32` utility, found in the `WINNT\System32` folder, adds registry entries that enable you to use the `schmmgmt.dll`.

To register the DLL file, choose Start⇨Run and type **cmd** in the text box to get to a command prompt. Then at the command prompt, type the following line:

```
regsvr32 schmmgmt.dll
```

Follow these steps to view the base schema:

1. **Open the Microsoft Management Console (MMC) by choosing Start⇨Run and typing** mmc **in the text box.**
2. **Select Console⇨Add/Remove Snap-in from the menu bar.**
3. **Select the Active Directory Schema snap-in. Click Add and then click OK.**

Only members of the Schema Administrators group can modify the schema. Limit membership in this group to a handful of select administrators. Any changes that you make to the schema by using the snap-in are irreversible, so make sure that you don't alter the schema unintentionally.

Figure 13-2 shows the Active Directory Schema in the left pane of the console window. Notice that the schema contains two types of components — classes and attributes — that appear in the right pane.

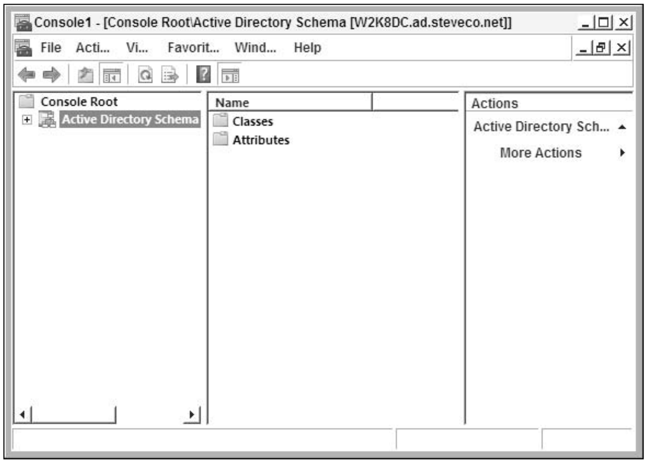
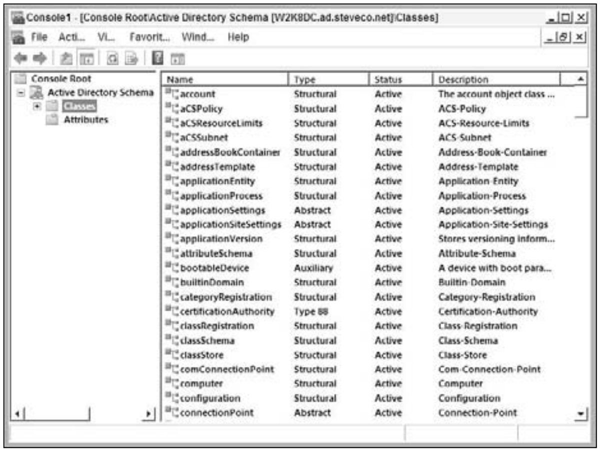


Figure 13-2: Viewing the Active Directory schema.

Explore a little bit further (just click any of the folders), and you see the default classes and attributes that make up the base schema of Active Directory. Clicking the Classes folder accesses a list of object classes in the right pane of the console window (see Figure 13-3).

Notice that each class that appears in Figure 13-3 includes a name, a type, and a description. Active Directory uses the Type category to create a structure within the directory. A class's type can be *Structural*, *Abstract*, or *Auxiliary* but you can only create new objects in the Structural type category.

The Abstract type serves as a template for creating new Structural classes. The Auxiliary type contains a list of attributes that can be associated with Structural class objects.


Console1 - [Console Root\Active Directory Schema [W2K8DC.ad.steveco.net]\Classes]

Name	Type	Status	Description
account	Structural	Active	The account object class ...
aCSPolicy	Structural	Active	ACS-Policy
aCSResourceLimits	Structural	Active	ACS-Resource-Limits
aCSSubnet	Structural	Active	ACS-Subnet
addressBookContainer	Structural	Active	Address-Book-Container
addressTemplate	Structural	Active	Address-Template
applicationEntity	Structural	Active	Application-Entity
applicationProcess	Structural	Active	Application-Process
applicationSettings	Abstract	Active	Application-Settings
applicationSiteSettings	Abstract	Active	Application-Site-Settings
applicationVersion	Structural	Active	Stores versioning inform...
attributeSchema	Structural	Active	Attribute-Schema
bootableDevice	Auxiliary	Active	A device with boot para...
builtinDomain	Structural	Active	Builtin-Domain
categoryRegistration	Structural	Active	Category-Registration
certificationAuthority	Type 88	Active	Certification-Authority
classRegistration	Structural	Active	Class-Registration
classSchema	Structural	Active	Class-Schema
classStore	Structural	Active	Class-Store
comConnectionPoint	Structural	Active	Com-Connection-Point
computer	Structural	Active	Computer
configuration	Structural	Active	Configuration
connectionPoint	Abstract	Active	Connection-Point

Figure 13-3: The object classes in the Active Directory schema.

A class with a Structural type follows an inheritance hierarchy that begins with an object class named `top`. As is true of any other class, both mandatory and optional attributes define `top`. Every structural object class descends from `top` and takes on *(inherits)* the attributes of `top`.

Figure 13-4 illustrates a structural-class hierarchy. `Top` is the parent class to `Object Class 1`, which inherits attributes from `top`. In turn, `Object Class 1` is the parent class to `Object Class 2`, which inherits its attributes and becomes parent to `Object Class 3`. The attributes that pass down from `top` extend through all the object classes in this branch. You can have several lines of inheritance descending from `top` — not all object classes descend in a single branch.

Examining object attributes

Now take a look at the second component of the Active Directory schema: object attributes. If you still have the console window open, click the Attributes folder to access a list of object attributes, as shown in Figure 13-5. (Refer to Step 2 in the earlier "Introducing object classes" section for instructions on opening the schema in Microsoft Management Console.)

Each of these attributes can become part of the definition of an object class. The attribute OID (object identifier), for example, is a mandatory attribute in every object class. The value of the attribute, however, is different for each object class.

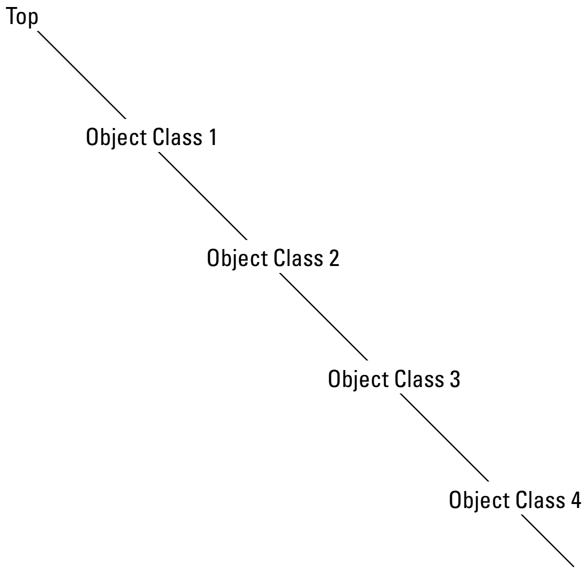
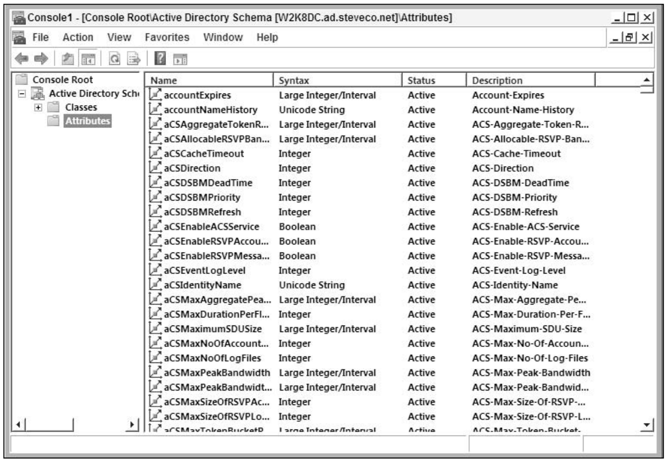


Figure 13-4: In a structural-class hierarchy, classes inherit the attributes of the `Top` object class.


Console1 - [Console Root\Active Directory Schema [W2K8DC.ad.steveco.net]\Attributes]
File
Action
View
Favorites
Window
Help
Console Root
Active Directory Sch
Classes
Attributes

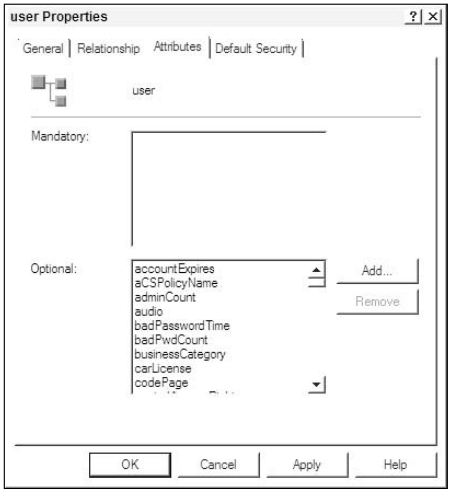
Name	Syntax	Status	Description
accountExpires	Large Integer/Interval	Active	Account-Expires
accountNameHistory	Unicode String	Active	Account-Name-History
aCSAggregateTokenR...	Large Integer/Interval	Active	ACS-Aggregate-Token-R...
aCSAllocableRSVPBan...	Large Integer/Interval	Active	ACS-Allocable-RSVP-Ban...
aCSCacheTimeout	Integer	Active	ACS-Cache-Timeout
aCSDirection	Integer	Active	ACS-Direction
aCSDSBMDeadTime	Integer	Active	ACS-DSBM-DeadTime
aCSDSBMPriority	Integer	Active	ACS-DSBM-Priority
aCSDSBMRefresh	Integer	Active	ACS-DSBM-Refresh
aCSEnableACSService	Boolean	Active	ACS-Enable-ACS-Service
aCSEnableRSVPAccou...	Boolean	Active	ACS-Enable-RSVP-Accou...
aCSEnableRSVPMessa...	Boolean	Active	ACS-Enable-RSVP-Messa...
aCSEventLogLevel	Integer	Active	ACS-Event-Log-Level
aCSIdentityName	Unicode String	Active	ACS-Identity-Name
aCSMaxAggregatePea...	Large Integer/Interval	Active	ACS-Max-Aggregate-Pe...
aCSMaxDurationPerFl...	Integer	Active	ACS-Max-Duration-Per-F...
aCSMaximumSDUSize	Large Integer/Interval	Active	ACS-Maximum-SDU-Size
aCSMaxNoOfAccount...	Integer	Active	ACS-Max-No-Of-Accoun...
aCSMaxNoOfLogFiles	Integer	Active	ACS-Max-No-Of-Log-Files
aCSMaxPeakBandwidth	Large Integer/Interval	Active	ACS-Max-Peak-Bandwidth
aCSMaxPeakBandwidt...	Large Integer/Interval	Active	ACS-Max-Peak-Bandwid...
aCSMaxSizeOfRSVPAc...	Integer	Active	ACS-Max-Size-Of-RSVP-...
aCSMaxSizeOfRSVPLo...	Integer	Active	ACS-Max-Size-Of-RSVP-L...
aCSMaxTokenBucketP	Large Integer/Interval	Active	ACS-Max-Token-Bucket-

Figure 13-5: The object attributes in the Active Directory schema.

To successfully create an object, the object must match all the criteria that the schema defines. The `classSchema` object defines the criteria for each class of object. These criteria consist of mandatory attributes and optional attributes. These mandatory and optional attributes define the rules for creating objects in the schema.

Similarly, the `attributeSchema` object defines attributes. The `attributeSchema` object defines how you create attributes in the schema. If you want to create a new attribute in the schema, the `attributeSchema` object tells you which attributes are required and which are optional.

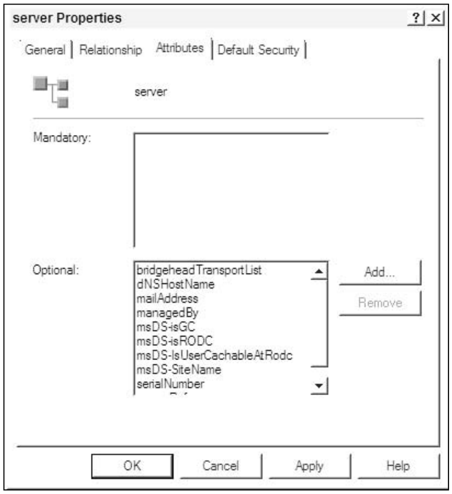
Every user object that you create in the tree must contain all the mandatory attributes that the classSchema object specifies. In addition, each user object can contain the optional attributes that you specify for the User object class. (You find out how to add classes and attributes in the following section.) Figure 13-6 shows the attributes of the User object class.



accountExpires
aCSPolicyName
adminCount
audio
badPasswordTime
badPwdCount
businessCategory
carLicense
codePage

Figure 13-6: The Attributes tab of the Properties dialog box for the User object class.

Figure 13-7 shows the attributes of the Server object class. Notice that the attributes that make up the Server object class differ from those that make up the User object class.



bridgeheadTransportList
dNSHostName
mailAddress
managedBy
msDS-isGC
msDS-isRODC
msDS-IsUserCachableAtRodc
msDS-SiteName
serialNumber

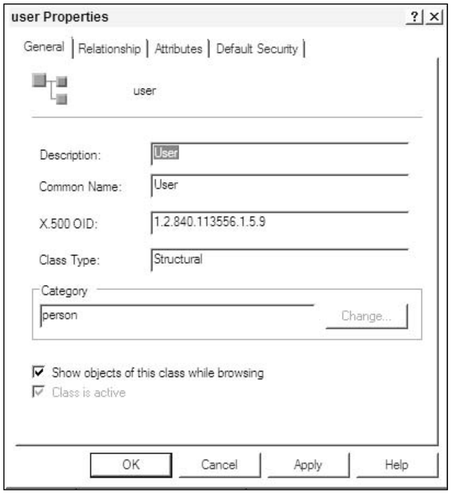
Figure 13-7: The Attributes tab of the Properties dialog box for the Server object class.

The following list describes all the properties of the User object class:

- **General tab:** Each item that you see on this tab is a mandatory attribute of the User object class (see Figure 13-8).
- **Relationship tab:** The parent class of the User object, along with Auxiliary Classes and Possible Superior appear on this tab (see Figure 13-9).

  The Relationship tab contains information about inheritance. The parent class is an object from which the User object class inherits attributes. In this example, the User object inherits attributes from a parent object named organizationalPerson. The User object class can also inherit attributes from any additional classes that appear under Auxiliary Classes or Possible Superior.

  To summarize the Relationship tab, the User object class inherits its attributes from its parent class and from any Auxiliary Classes. (Similarly, the parent class inheritvs attributes from its parent class, and so on.)
- **Attributes tab:** This tab lists the mandatory and optional attributes of the User object class (see Figure 13-10). Remember that some attributes are inherited from the parent class.



Description: User

Common Name: User

X.500 OID: 1.2.840.113556.1.5.9

Class Type: Structural

Category: person

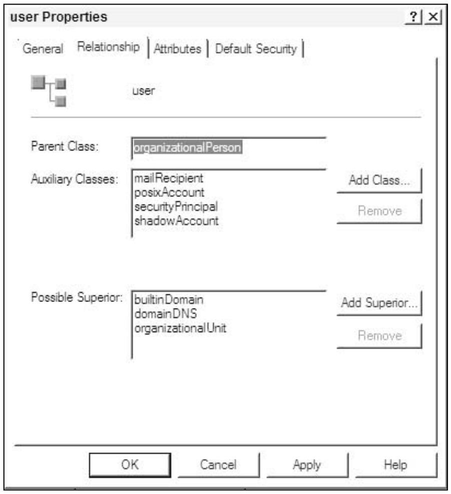
☑ Show objects of this class while browsing

☑ Class is active

Figure 13-8: The General properties table of the User object class.

  Inherited attributes don't appear on the Attributes tab. You must view the parent object class to see which attributes it passes on to the child object class.

- **Default Security tab:** This tab displays the permissions of each group or user in relationship to User objects (see Figure 13-11). You can see, for example, that the Account Operators group has full control of this object class.



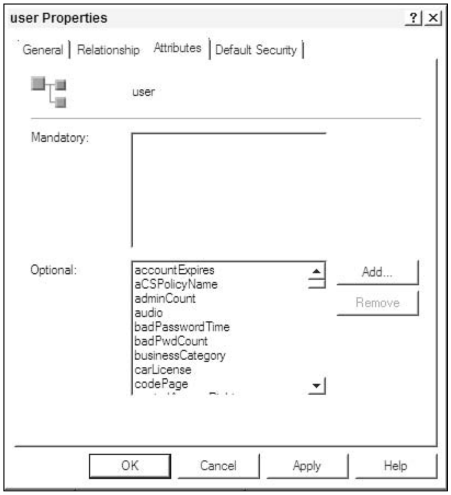
user

Parent Class: organizationalPerson

Auxiliary Classes: mailRecipient
posixAccount
securityPrincipal
shadowAccount

Possible Superior: builtinDomain
domainDNS
organizationalUnit

Figure 13-9: The Relationship tab of the `User` object class.



user

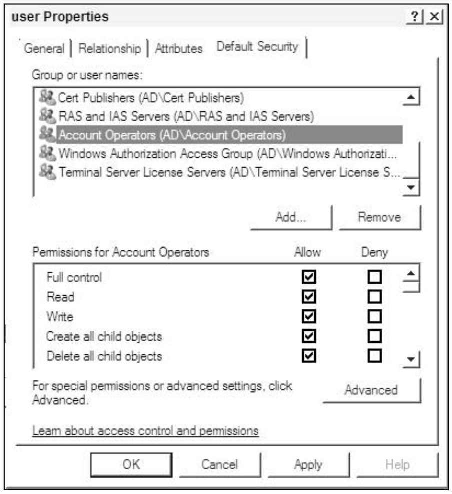
Mandatory:

Optional: accountExpires
aCSPolicyName
adminCount
audio
badPasswordTime
badPwdCount
businessCategory
carLicense
codePage

Figure 13-10: The Attributes properties tab of the `User` object class.

To summarize, the Active Directory schema contains a list of object attributes. From this list, attributes combine in groupings of mandatory and optional attributes to form object classes.

Note that you can also use the ADSIEdit console as an alternative tool to viewing the contents of the Active Directory schema.



Group or user names:

- Cert Publishers (AD\Cert Publishers)
- RAS and IAS Servers (AD\RAS and IAS Servers)
- Account Operators (AD\Account Operators)
- Windows Authorization Access Group (AD\Windows Authorizati...
- Terminal Server License Servers (AD\Terminal Server License S...

Permissions for Account Operators	Allow	Deny
Full control	☑	□
Read	☑	□
Write	☑	□
Create all child objects	☑	□
Delete all child objects	☑	□

For special permissions or advanced settings, click Advanced.

Learn about access control and permissions

Figure 13-11: The Default Security properties tab of the `User` object class.

Extending the Schema

As robust as the default schema that comes with Active Directory is, at times, you're going to want to make changes to the schema. Adding to or modifying the Active Directory schema is *extending* the schema. The schema is the core of the Active Directory database, and you need to treat it with extreme care. Yes, I said it earlier, but it bears repeating: Extending the schema is highly complex. You need to limit membership in the Schema Administrators group to a handful of skilled administrators.

Incorrect modifications to the schema can corrupt your Active Directory database. Whenever possible, use objects and attributes that are already defined by the base schema.

Modifications to the schema can include the following:

- Creating a class
- Extending a class
- Deactivating a class
- Creating an attribute
- Modifying an attribute
- Deactivating an attribute

Typically, extending the schema is done to support an application that uses Active Directory. Microsoft Exchange Server is a good example of this. Exchange uses AD to store information about how it's configured within the forest. This requires the creation of new object classes that are not a part of the normal AD schema. It is also necessary to extend the normal users, groups, and contacts objects to include additional attributes so that Exchange can enable these objects to send and/or receive e-mail. All these changes must be executed against the default AD schema before the first Exchange server can be installed.

Notice that deleting a class or attribute isn't an option. After you make changes to the schema, you can't remove them. You can disable (or deactivate) them, but you can't delete them. I'll talk about deactivating changes in a moment.

REMEMBER

You cannot make changes to the base system schema that is required for Active Directory to function properly. This prevents you from really messing things up!

You need to prepare sufficiently if you're going to extend the schema. Consider that adding an object or attribute requires a unique OID (object identifier) that a regulatory authority must issue. In addition, you must specify (among other items) syntax and indexing. You face a lot of restrictions on what you can or can't change within the schema, and these restrictions vary depending on whether the object is a base schema object or an extended object.

Adding classes and attributes

Adding a class or attribute is relatively simple. You have two methods for doing this: either by a scripted operation or by using the Active Directory Schema console or the ADSIEdit console. While using one of these consoles might seem like an easier approach, I strongly recommend that if you're going to do this in a production environment, you script the schema modification by using a tool, such as the LDIFDE command line tool. Because the changes cannot be deleted, you can reduce the chances of making a mistake during the change by scripting the operations out rather than using a GUI-based tool.

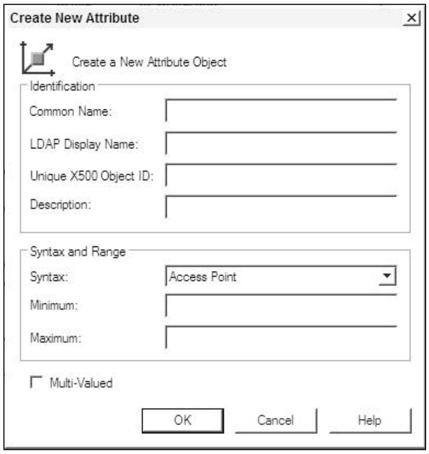
TIP

For more information about the LDIFDE tool, go to the Windows Server 2008 TechCenter at:

```
http://technet.microsoft.com/windowsserver/2008
```

If you choose to use the GUI MMC tool, open the Active Directory Schema Manager from the MMC as I describe in the earlier "Introducing object classes" section. To add a class, select the Classes folder, right-click, and choose Create Class from the contextual menu. After acknowledging the warning about modifying the schema, the Create New Schema Class dialog box appears.

To add an attribute, select the Attributes folder, right-click, and choose Create Attribute from the contextual menu. The Create New Attribute dialog box appears, as shown in Figure 13-12.


Create New Attribute

Create a New Attribute Object

Identification

Common Name:

LDAP Display Name:

Unique X500 Object ID:

Description:

Syntax and Range

Syntax: Access Point

Minimum:

Maximum:

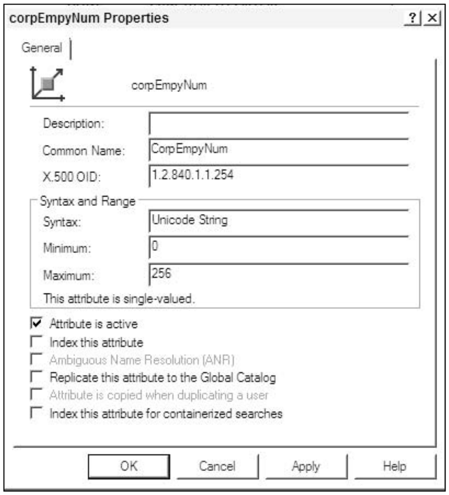
□ Multi-Valued

Figure 13-12: Add an attribute by using the Create New Attribute dialog box.

After you add a class or attribute, modifying it is much like modifying any other Active Directory object. Select the class or attribute that you want to modify, right-click, and choose Properties from the contextual menu.

Deactivating objects

Finally, you can deactivate classes and attributes by using the Active Directory Schema snap-in. If you right-click a particular class or attribute, the Properties dialog box for that object appears, as shown in Figure 13-13. Clear the Class is Active or the Attribute is Active check box depending on if you're disabling an object class or attribute. You can't, however, deactivate the base objects that install automatically with Active Directory. You can only deactivate objects that you add to your schema. Be careful in deactivating objects! Most objects are part of a parent-child inheritance structure, so you might inadvertently alter all the objects in an inheritance branch by deactivating a parent object. You can resurrect objects that you deactivate; simply select the same check box to reactivate the object.



corpEmpyNum

Description:

Common Name: CorpEmpyNum

X.500 OID: 1.2.840.1.1.254

Syntax and Range

Syntax: Unicode String

Minimum: 0

Maximum: 256

This attribute is single-valued.

☑ Attribute is active

□ Index this attribute

□ Ambiguous Name Resolution (ANR)

□ Replicate this attribute to the Global Catalog

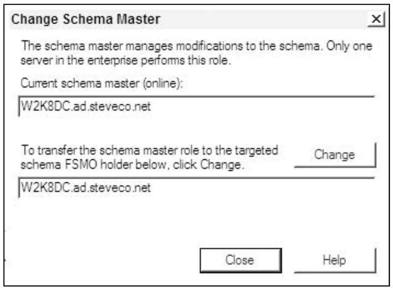
□ Attribute is copied when duplicating a user

□ Index this attribute for containerized searches

Figure 13-13: Deactivating a schema object by clearing the Attribute is Active option.

Transferring the Schema Master

Changes to the schema can take place on only one domain controller at a time. This DC is the *schema master operations master*. The first domain controller online becomes, by default, the schema master which is the only domain controller where changes to the schema can be written to. There might be times when you need to transfer the schema master role to another domain controller. To do so with the Active Directory Schema console, open the console and right-click the Active Directory Schema in the left pane of the MMC. On the resulting contextual menu, choose Operations Master, and the dialog box shown in Figure 13-14 appears. The server that's currently the schema master is displayed near the center of this dialog box. Click the Change button and type a different server name to change the schema master to a different server.



The schema master manages modifications to the schema. Only one server in the enterprise performs this role.

Current schema master (online):

W2K8DC.ad.steveco.net

To transfer the schema master role to the targeted schema FSMO holder below, click Change.

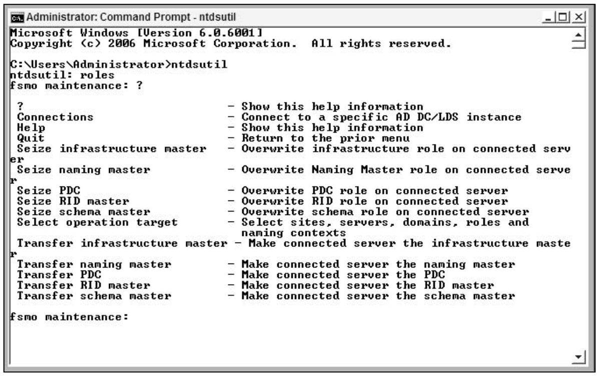
W2K8DC.ad.steveco.net

Figure 13-14: Dialog box for transferring the schema master operations master holder.

You can also transfer the schema master role by using the NTDSUTIL tool. You can use the NTDSUTIL tool to transfer the role, but this tool is really useful when the schema master domain controller becomes unavailable and you need to transfer the role forcefully to another server. (This process is called *seizing* the schema master role.)

To use NTDSUTIL, choose Start➪Run and type **cmd** in the text box to get to a command prompt. Then at the command prompt, type **ntdsutil roles ?**.

This starts NTDSUTIL and displays the menu shown in Figure 13-15. At the `fsmo maintenance` prompt, type **seize schema master**.



```
Microsoft Windows [Version 6.0.6001]
Copyright (c) 2006 Microsoft Corporation.  All rights reserved.

C:\Users\Administrator>ntdsutil
ntdsutil: roles
fsmo maintenance: ?

 ?                             - Show this help information
 Connections                   - Connect to a specific AD DC/LDS instance
 Help                          - Show this help information
 Quit                          - Return to the prior menu
 Seize infrastructure master   - Overwrite infrastructure role on connected serv
er
 Seize naming master           - Overwrite Naming Master role on connected serve
r
 Seize PDC                     - Overwrite PDC role on connected server
 Seize RID master              - Overwrite RID role on connected server
 Seize schema master           - Overwrite schema role on connected server
 Select operation target       - Select sites, servers, domains, roles and
                                 naming contexts
 Transfer infrastructure master - Make connected server the infrastructure maste
r
 Transfer naming master        - Make connected server the naming master
 Transfer PDC                  - Make connected server the PDC
 Transfer RID master           - Make connected server the RID master
 Transfer schema master        - Make connected server the schema master

fsmo maintenance:
```

Figure 13-15: Using NTDSUTIL to seize the schema master operations master holder.

Seizing the schema master role is a last-resort action. If the previous schema master comes back online after you seize the role, inconsistencies might exist between the two servers. If the new schema master contains newer schema updates, you might lose the new updates when the old schema master comes back online.

Reloading the Schema Cache

Each time a computer that's running Active Directory boots up, a copy of the Active Directory schema loads into the computer's memory. The memory-resident copy is the *schema cache,* and it serves as a performance enhancement because the system can access memory-resident data more quickly than it can access data from disk.

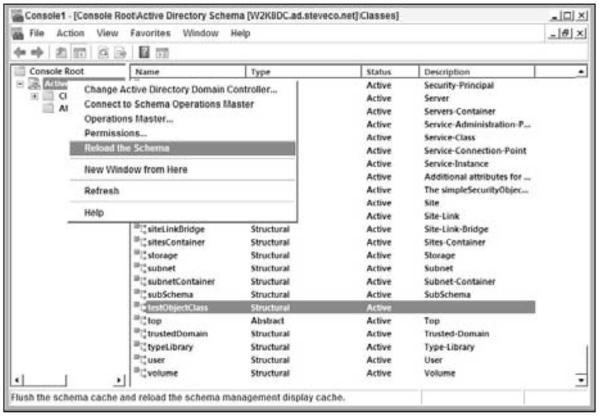
When you make changes to the schema, the changes take place on the disk-based version of the schema before the changes update in the schema cache.

The update begins five minutes after a schema modification. If you make a second change, the five-minute timer starts over.

Obviously, during the five-minute interval before your schema changes are updated in the schema cache, the two versions (disk and cache) are out of synch. You can shorten the time interval by selecting the Reload the Schema option from the Action menu in the Active Directory Schema Manager, as shown in Figure 13-16.

When making a series of changes to the schema, don't use the Reload the Schema option more than once. After the schema cache updates, Active Directory creates a new cache from the disk-based database. Thus, two cached copies of the schema can exist during the update process. By triggering Reload the Schema multiple times, you can end up with numerous copies of the schema cached in memory, which can significantly slow the performance of the system. Use the Reload the Schema option only once after completing all your modifications.

Because the schema is part of the Active Directory database, updates to the schema replicate to all domain controllers in the tree or forest. In a large enterprise, *replication latency* (the time between the start and finish of replication to all domain controllers) can temporarily result in out-of-synch schemas between domain controllers. Replication latency is unavoidable, but temporary. After all domain controllers have replicated, the updates are available across the domain.


Console1 - [Console Root\Active Directory Schema [W2K8DC.ad.steveco.net]\Classes]
File
Action
View
Favorites
Window
Help
Console Root
Active
Cl
At
Change Active Directory Domain Controller...
Connect to Schema Operations Master
Operations Master...
Permissions...
Reload the Schema
New Window from Here
Refresh
Help

Name	Type	Status	Description
		Active	Security-Principal
		Active	Server
		Active	Servers-Container
		Active	Service-Administration-P...
		Active	Service-Class
		Active	Service-Connection-Point
		Active	Service-Instance
		Active	Additional attributes for ...
		Active	The simpleSecurityObjec...
		Active	Site
		Active	Site-Link
siteLinkBridge	Structural	Active	Site-Link-Bridge
sitesContainer	Structural	Active	Sites-Container
storage	Structural	Active	Storage
subnet	Structural	Active	Subnet
subnetContainer	Structural	Active	Subnet-Container
subSchema	Structural	Active	SubSchema
testObjectClass	Structural	Active	
top	Abstract	Active	Top
trustedDomain	Structural	Active	Trusted-Domain
typeLibrary	Structural	Active	Type-Library
user	Structural	Active	User
volume	Structural	Active	Volume

Flush the schema cache and reload the schema management display cache.

Figure 13-16: Reloading the schema cache.

Chapter 14

Managing Security with Active Directory Domain Services

In This Chapter

- Examining authentication with Kerberos and NTLM
- Working with group policies
- Password and account lockout policies
- AD auditing

These days, it seems that you can't go for more than a week without hearing of a new security vulnerability on the Internet. Operating systems, Web browsers, mail systems, Web servers — all are vulnerable to persistent intruders who want access to your company's resources.

Although attacks from outside intruders are certainly the stuff of best-selling novels, they aren't that farfetched. Even more critical are the "attacks" from within, which are far more common. If you've been an administrator for very long, you know that even well-meaning administrators and users can make costly mistakes. Fortunately, Microsoft includes many security features in Active Directory Domain Services (AD DS) to secure your environment. In this chapter, I talk about authentication, Kerberos security, group policies, and some of the new security-related topics in Windows Server 2008.

NTLM and Kerberos

Active Directory offers the following two domain-authentication protocols:

- NTLM
- Kerberos

These two protocols are used to provide authentication and authorization services within Active Directory. *Authentication* is the process of proving who you are to an authority (in this case a domain controller). This is typically done at the initial login by providing your user logon ID and password. After you have proved your identity to AD, then you can attempt to access network resources. Before you can access a network resource, you must go through a process of *authorization* to prove that you are allowed to access the network resource. So let's look at each of these protocols and how they work.

NTLM authentication

NTLM is the authentication/authorization protocol that was used in Windows NT environments before AD and Kerberos were available. Even though Kerberos has been around since the release of AD in Windows 2000 Server, you may still have applications or computers that rely on NTLMrather than Kerberos. Therefore, this security protocol continues to be supported in Windows Server 2008.

Each user, computer, or security group object receives a unique *Security Identification Number (SID)* that distinguishes it from other objects in the network. This SID is how that object is identified to the security subsystem in AD. Similarly, each resource in the network — such as printers, files, and servers — maintains an Access Control List (ACL) of SIDs that can access the resource. The ACL consists of entries that detail what type of access (read-only, write, execute, and so on) the user has to the resource. These entries are known as Access Control Entries, or ACEs.

Each time a user logs on to the domain, the system retrieves the user's security information from a domain controller and encodes it in an *access token* that shows the SID and the resources the SID can access. The access token is the user's key to domain resources. Whenever a user attempts to access a domain resource, the system compares the access token with the resource's ACL. If the ACL lists adequate access privileges for the user, the system grants that user access to the resource. Figure 14-1 illustrates the NTLM authentication/authorization process.

Meet Kerberos, the guard dog

Kerberos Version 5, as RFC 1510 defines, is the primary security authentication/authorization protocol in AD DS. (If you need a refresher on RFCs, refer to Chapter 4.) Kerberos is a *distributed security protocol,* which means that it enables users to access resources anywhere on the network by using a single logon.

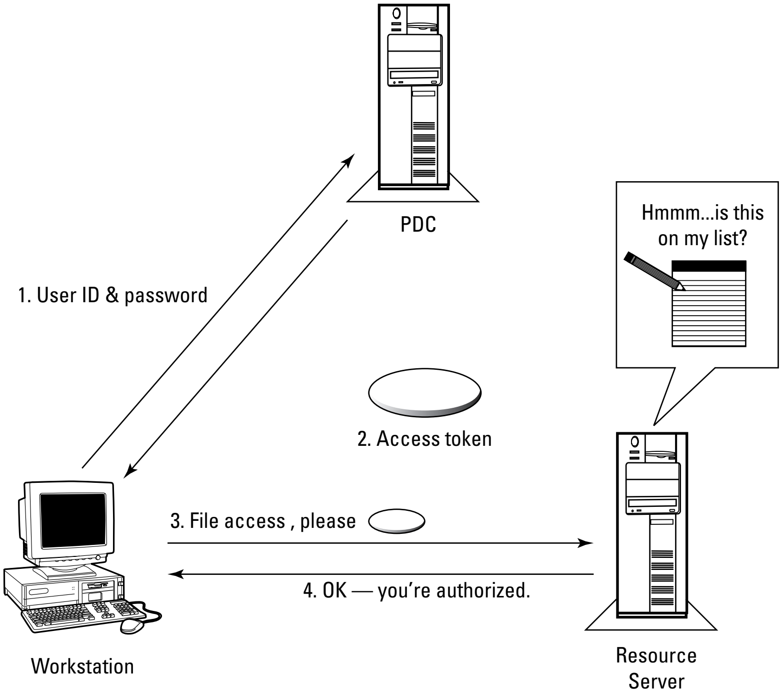


Figure 14-1: Authenticating user logons by using NTLM.

The name *Kerberos* is a little intimidating, isn't it? In Greek mythology, Kerberos (also known as Cerberus) is a three-headed dog who guards the gates of Hades (the underworld). The MIT (Massachusetts Institute of Technology) developers who created the Kerberos protocol in the 1980s thought it was an appropriate name for their new security protocol. I don't know about you, but I'm reluctant to associate my network with the gates of Hades! Despite the association, Kerberos is a mature industry-standard protocol that's well suited for distributed computing environments.

Client computers running Windows XP or Windows Vista use Kerberos to authenticate with an AD domain controller. Servers running Windows 2000, Windows Server 2003, and Windows Server 2008 also use Kerberos for authentication. Kerberos requires both client and server to authenticate, or log on, thus preventing an intruder from impersonating either client or server.

Following are the features that Kerberos brings to the Active Directory (AD) party:

- Faster logon authentication in a distributed computing environment
- Transitive trust relationships between domains
- Delegated (or pass-through) authentication for distributed applications
- Interoperability with non-Windows systems that use the Kerberos protocol

Kerberos uses a *shared secret,* also known as a *key,* so that both the client and a service, known as the *Key Distribution Center (KDC),* which runs on every domain controller, share the same key. In the case of a user authentication, this key is the hash of the user's password.

The KDC is actually made up of two components: the Authentication Service (AS), which provides the initial logon services, and the Ticket-Granting Service (TGS), which provides tickets for access network resources after the user has logged in. Kerberos takes the following actions to authenticate logon requests and then authorize the user to access a network resource:

1. **A user authenticates with the KDC by logging in.** (KDC runs as a service on domain controllers). The authentication message includes the user's login name, the domain name the user is logging into, and a timestamp. This information is encrypted with the hash of the user's password. This way the user's password is never sent across the network, thereby keeping it secure.
2. **The AS component of the KDC receives the authentication request and then validates that it can be decrypted with the hash of the user's password which it accesses from the AD database.** If the decryption is successful and the timestamp is within five minutes of the DC's current time, the authentication is considered successful. The KDC returns a *Ticket Granting Ticket* (or *TGT*) to the client. The TGT contains the user's SID and the SIDs of all groups of which the user is a member.
3. **The client caches the TGT until it needs to access a network resource; it then presents the TGT to the TGS component of the KDC and requests access to a resource server.**
4. **The KDC returns a *Session Ticket* (ST), which contains an encrypted key code known only to itself and to the resource server.**
5. **The client presents the ST to the resource server.**
6. **The resource server examines the ST for a key code that is known only to the server and the KDC.**
7. **If the key code matches the resource server's key code, the client receives access to the resource server. If the key code doesn't match the resource server's key code, the client doesn't receive access.**

The theory behind this transaction is that the resource server says, "Hmm. I know and trust the KDC. The KDC must know and trust this account because it gave the account this ST. If I trust the KDC and the KDC trusts the user account, I can, therefore, trust the user account, too." Figure 14-2 illustrates the Kerberos authentication process.

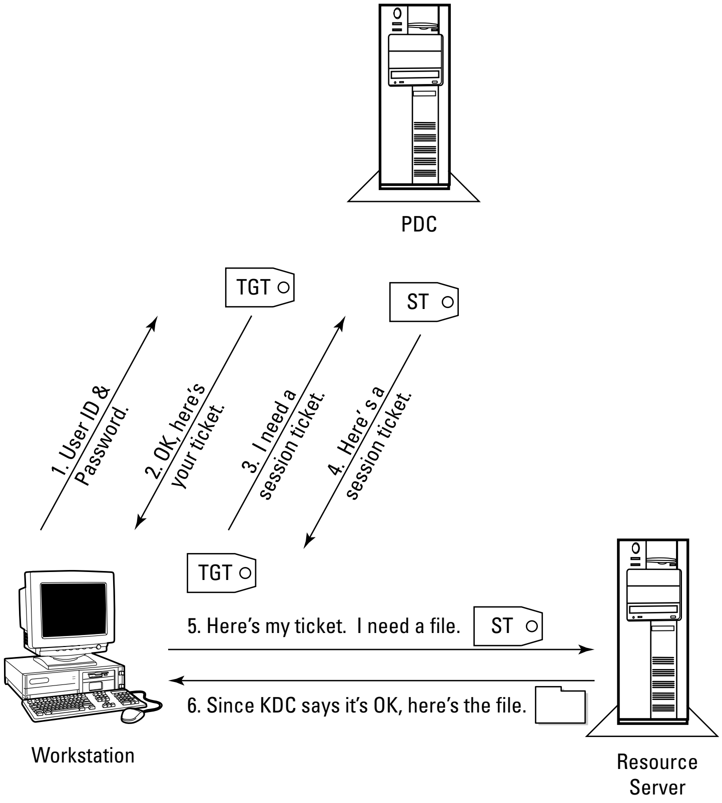

PDC
TGT
ST
1. User ID & Password.
2. OK, here's your ticket.
3. I need a session ticket.
4. Here's a session ticket.
TGT
5. Here's my ticket. I need a file.
ST
6. Since KDC says it's OK, here's the file.
Workstation
Resource Server

Figure 14-2: The Kerberos authentication process.

Despite the added complexity of Kerberos authentication, it's faster than NTLM authentication because the client can reuse its ST tickets during a logon session. And, because Kerberos is a standard Internet protocol, AD DCs (domain controllers) can authenticate clients running any operating system that uses Kerberos. Users on a network with other systems that support Kerberos (like UNIX) can log on and access any resource in either the AD environment or UNIX via a single logon.

Implementing Group Policies

One of the most powerful features that AD provides is to support Group Policies. The administrator uses group policies objects (GPOs) in AD to enforce a set of configuration settings to both the user's computer as well as the user's logon session on that computer. These policies contain configuration settings that can alter the desktop appearance, provide standard configuration settings for security, provide software installation services, and many other things. How often you use GPOs is proportional to the level of managed control you want to place over your users and computers. Before you decide if you should use GPOs, you have to carefully balance the needs of the users with your need to lock down the environment in order to provide benefit, such as reducing your total cost of ownership or creating a highly secured environment.

Using GPOs within Active Directory

Group policy objects exist as two separate components, as follows:

- The *GPO container* is an object within AD. This object's attributes include the name of the GPO, the GPO's ACL (which controls who can modify the GPO's contents), version control information, and the GPO's enabled status.
- The *GPO template* is stored as a set of files on the `SYSVOL` directory on every DC in the domain where the GPO exists. The template contains the administrative templates and scripts that are related to the GPO.

The `SYSVOL` directory exists on each domain controller, and its contents are replicated between all domain controllers in a domain via the File Replication Service (FRS).

With GPOs, you have the ability to alter settings that are related to computers and to users. The computer settings are applied during the computer's boot process. These settings affect computer operation: the startup and shutdown computer scripts, security settings, and software deployment. Because these computer settings are applied at boot time, they affect any user that logs into that computer. The user settings in a user's GPO are applied to a user's session on a computer only when the user logs into the computer.

Similar to the computer settings, the user settings can affect the appearance and behavior of the OS, provide user logon and logoff scripts, and provide software deployment services. The user GPOs settings remain in effect only during a user's logon session to that computer.

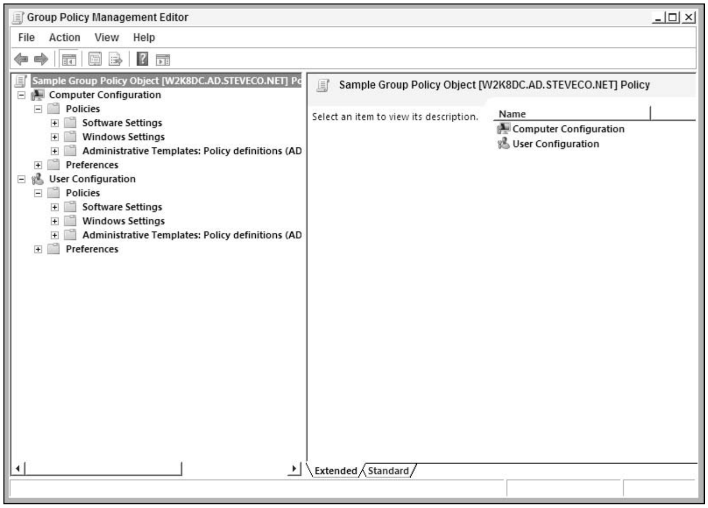
GPO settings are automatically refreshed every 90 minutes with a random offset of plus or minus 30 minutes. So, if you make a change to a GPO, you don't have to wait for a user to log off or for a computer to be rebooted before the settings are propagated out. Domain controllers are the exception to this rule. Because the security policy settings on a DC are critical to the environment, the default GPO refresh setting on DCs is five minutes. Ironically, you can change these numbers through a GPO setting if they don't meet your needs.

All settings in GPOs are split into two major areas: Computer Configuration and User Configuration. The Computer Configuration section controls how the computer is configured, and the User Configuration section controls the user's logon session. The Computer and User Configuration settings are then divided into the following areas:

- **Software Settings:** You can control software installation and configuration through Group Policy Objects. This includes the ability to assign or publish software to a user or computer. By assigning software to a user or computer, you automatically create entries for the software in the Start menu, which upon invocation, initiates an install of the application. Publishing an application makes it available through the Windows XP Control Panel app, Add/Remove Application; or, through the Windows Vista Control Panel app, Programs and Features.
- **Windows Settings:** Through Windows Settings, you can create and edit scripts that are executed during a computer's startup and shutdown sequence and in a user's logon or logoff process. You also find here settings that relate to a security and folder redirection. Within the security settings, you have the password policy that controls actions, such as password expiration, password length, and complexity settings.
- **Administrative Templates:** Within Administrative Templates, you have available a customized set of controls that affect application configuration, desktop appearance, group policy behaviors, and the level of customization that you can do to the computer/user environment.

The Group Policy Management Editor displaying each of these areas is shown in Figure 14-3.

Within Windows Settings, you (finally!) find the Security Settings. Table 14-1 details the more-common security policy areas that are configurable through a GPO.


Group Policy Management Editor
File
Action
View
Help
Sample Group Policy Object [W2K8DC.AD.STEVECO.NET] Po
Computer Configuration
Policies
Software Settings
Windows Settings
Administrative Templates: Policy definitions (AD
Preferences
User Configuration
Policies
Software Settings
Windows Settings
Administrative Templates: Policy definitions (AD
Preferences
Sample Group Policy Object [W2K8DC.AD.STEVECO.NET] Policy
Name
Computer Configuration
User Configuration
Extended
Standard

Select an item to view its description.

Figure 14-3: The configuration areas within a Group Policy Object.

Table 14-1 GPO Security Settings

Policies	*Settings*
Account Policies	Configure password policy, account lockout policy, and Kerberos policy.
Local Policies	Configure audit policies, user rights assignments, and security options.
Event Log	Configures application, system, and security logs.
Restricted Groups	Control group memberships for critical groups, such as Administrator and Schema Administrator.
System Services	Control security settings for system services.
Registry	Configure security on the registry.
File System	Control security on folders.
Public Key Policies	Configure trusted certificate authorities.
IP Security Policies	Control IP security settings throughout the network.

With the right combination of security settings, you can control and audit membership in administrative groups, restrict access to computer registries, restrict remote or local access to computers, and much more. If you use them correctly, group policies are a powerful tool for streamlining administration.

Some GPO security settings — including the Password Policy, Account Lockout Policy, and Kerberos Policy — are applicable only at the domain level. This means that the default behavior is for all users in the same domain to receive the same password policy. But, as you can read in just a moment, Windows Server 2008 provides a new feature that allows you to actually provide multiple password policies within the same domain.

GPOs exist as independent objects in AD. To enable a GPO to be applied to a user or computer, you must first link the GPO to an AD site, domain, or OU container in which the user or computer is located. By default, any linked GPO is applied to any computers or user objects in that container.

GPO inheritance and blocking

Active Directory permissions flow from the top of a container hierarchy to the bottom, resulting in a sum of permission settings. GPOs operate in a similar manner, when looking at the GPOs that apply to the user or computer from the top of the hierarchy to the bottom. For GPOs, that hierarchy is in the order of Local Policy (a policy on the local computer that is separate from AD), AD Site, AD Domain, and then OU. Note that you might have multiple GPOs applied in any one of these containers as well.

Figure 14-4, depicting an AD design comprising a single AD site, a single AD domain, and an OU structure consisting of the OUs, illustrates this concept. Each of these container objects has a GPO linked to it (trapezoids named GPO1–GPO5).

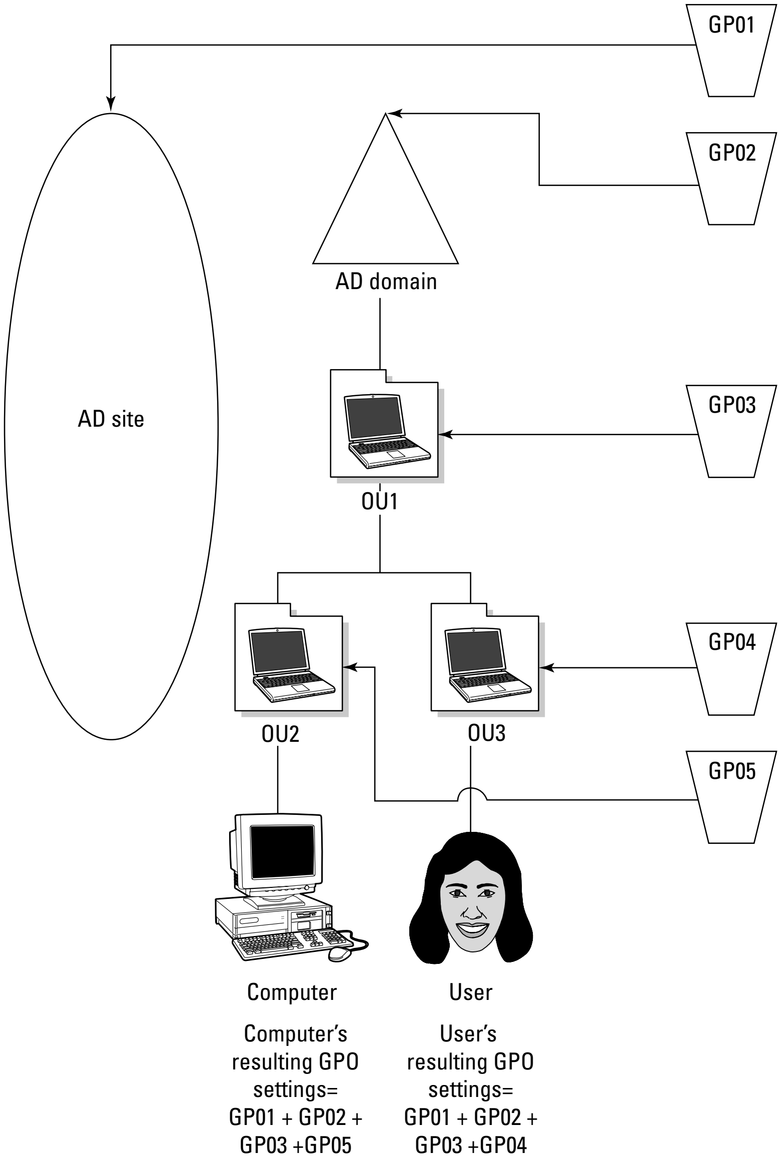

GP01
GP02
AD domain
AD site
GP03
OU1
GP04
OU2
OU3
GP05
Computer
User
Computer's resulting GPO settings= GP01 + GP02 + GP03 +GP05
User's resulting GPO settings= GP01 + GP02 + GP03 +GP04

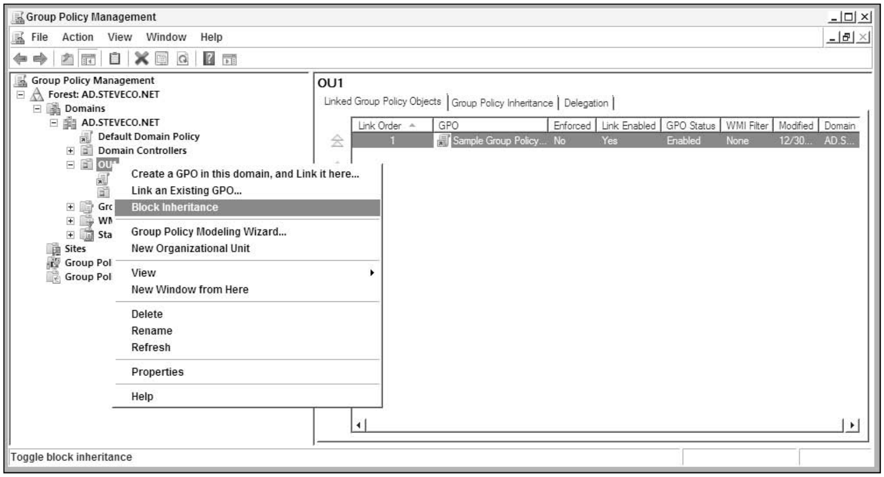
Figure 14-4: How GPOs are applied to objects in Active Directory.

The computer object exists in OU2, and the user object exists in OU3. When the computer boots, the following happens:

1. The local policy on the computer is loaded and applied to the computer.
2. The computer configuration portion of GPO1, from the computer's AD site, is applied.
3. The computer configuration portion of the computer's domain GPO is applied.
4. The computer configuration portion of GPO3 is applied to the computer because the computer is in OU1's hierarchy.
5. The computer configuration portion of GPO5 is applied as the last GPO.
6. When an end-user logs into the user object, the user configuration portions of GPOs 1, 2, and 3 applied. Then GPO4 is applied because the user is in OU3.

When GPOs are combined, the configuration settings in each GPO are combined to get the final configuration. What happens if the GPO configuration settings conflict? The lower-level container GPOs override the GPOs linked to the higher-level containers. However, you can alter this behavior. Through the Group Policy Management Console (which I discuss in just a second), you can configure a site, a domain, or an OU to block the inheritance of any higher-level GPO, as shown in Figure 14-5.

However, this blocking of inheritance is not absolute. Because the higher-level containers are normally supported by administrators whose responsibility is the entire infrastructure, Microsoft provides an Enforce option that you can apply to a GPO. With this option enabled, a GPO and its settings cannot be overridden by a lower-level GPO or by one in a container that has the Block Inheritance option turned on.



Link Order	GPO	Enforced	Link Enabled	GPO Status	WMI Filter	Modified	Domain
1	Sample Group Policy...	No	Yes	Enabled	None	12/30...	AD.S...

Figure 14-5: Edit the Block Inheritance option on an OU.

In Windows 2000 and 2003, the Enforce option is called No Override.

Figure 14-6 features two OU structures. In each case, GPO1 is linked to OU1, and GPO2 is linked to OU2. The Block Inheritance option has been turned on for OU2 in the first example. This blocks the higher level GPO (GPO1) from being applied to the computer in OU2. Thus, the computer's resulting settings come only from GPO2. In the second example, the Enforce setting is applied to GPO1. This means that any setting in this GPO cannot be overwritten by a lower-level GPO. Thus, the computer's resulting configuration is the combination of the settings in GPO1 plus the settings in GPO2 that don't conflict with GPO1.

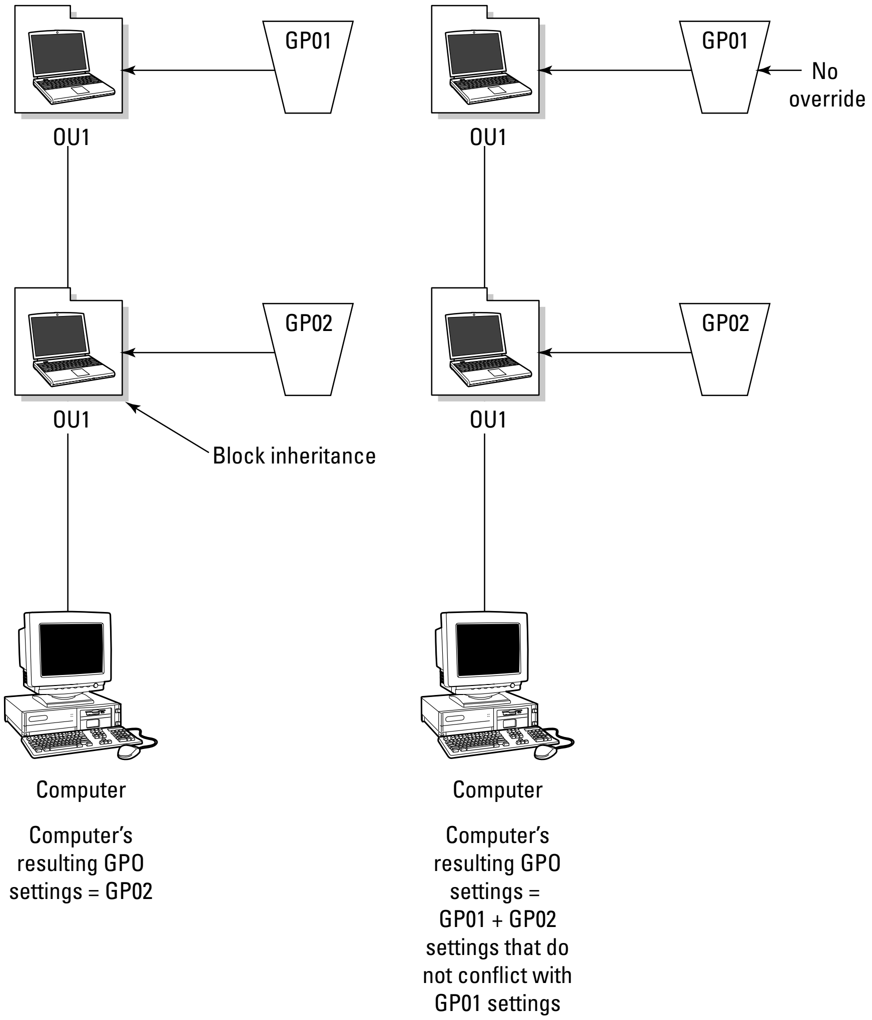

GP01
OU1
GP02
OU1
Block inheritance
Computer
Computer's resulting GPO settings = GP02
GP01
No override
OU1
GP02
OU1
Computer
Computer's resulting GPO settings = GP01 + GP02 settings that do not conflict with GP01 settings

Figure 14-6: GPO Block Inheritance and Enforce features.

Group policy management

Beginning in Windows Server 2003, Microsoft introduced a separate tool for managing group policies: the Group Policy Management Console (GPMC). That tool now comes built in to Windows Server 2008. Besides the ability to create, link, modify, and delete GPOs, this tool also allows you to create reports and model the effects of your GPOs on the user and computer. Take a look at some of these abilities.

Creating a group policy

Follow these steps to create a group policy for a site, a domain, or an OU:

1. **From the Start menu, choose Programs➪Administrative Tools➪Group Policy Management**

   The Group Policy Management Console (GPMC) appears, as shown in Figure 14-7.

2. **In GPMC, open the Forest container, open the Domain container, and then open the specific domain in which you want to create the GPO.**
3. **Right-click the Group Policy Objects container and choose New from the contextual menu.**
4. **Type in the name of the new GPO into the Name field.**
5. **(Optional) If you want to select a starter GPO on which to base the new GPO, select that GPO from the Starter GPO field.**
6. **Click OK.**

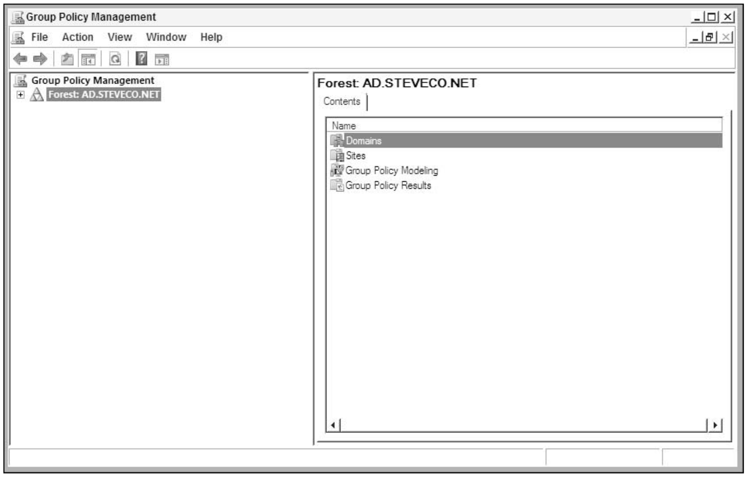


Figure 14-7: The Group Policy Management Console.

TIP

Starter GPOs are a new concept for Windows Server 2008. The idea is that instead of building your GPOs from scratch each time, you can create a starter GPO that can be used as a template for building additional GPOs. This can be useful especially if you need to create multiple GPOs that share most of the same settings. Note, though, that Starter GPOs have only the Administrative Templates section: Windows and Software Settings elements aren't available.

Editing a group policy object

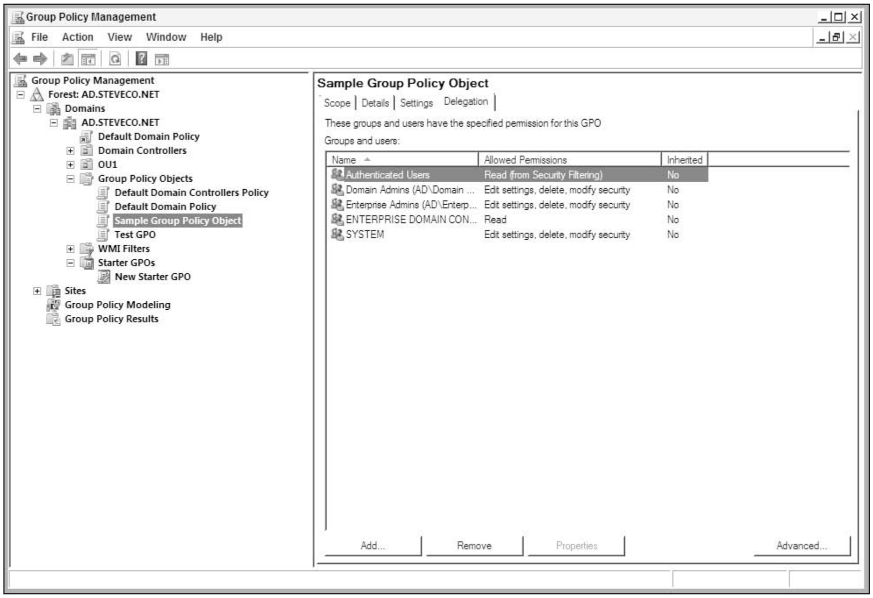
Earlier, I mention the two parts of a GPO: the GPO container and the GPO template. Inherently, you also have two parts of the GPO to edit. In editing the GPO container, you control the access to the GPO, whether the GPO is enabled, what parts of the GPO is enabled, and who can modify the GPO. To edit the GPO container, follow these steps:

1. **In the Group Policy Objects container in GPMC, select the GPO you want to edit.**
2. **From the Scope tab, you can control which users can have the GPO applied to them under the Security Filtering section.**

   The default is the members of the Authenticated Users group, which means that anyone logged into AD can have the GPO applied to them. Of course, the GPO will actually be applied to the user or computer if the GPO is actually applied to a container that you're in.
3. **From the Details tab, you can control whether the GPO is enabled. You can also choose to enable only the Computer Configuration or the User Configuration portion of the GPO.**
4. **From the Delegation tab (see Figure 14-8), you can view and change the security permissions that users and security groups have against this GPO.**

To edit the GPO template, simply right-click the GPO in the GPMC and choose Edit.

As much as I wish I could cover all the settings that are possible to control within the GPO templates, I simply can't. The templates that come in Windows Server 2008 have more than 2,400 settings! Fortunately, on the Explain panel for each setting, Microsoft does a pretty good job explaining what the settings mean. For example, in Figure 14-9, you see the explanation of the Minimum Password Length setting.

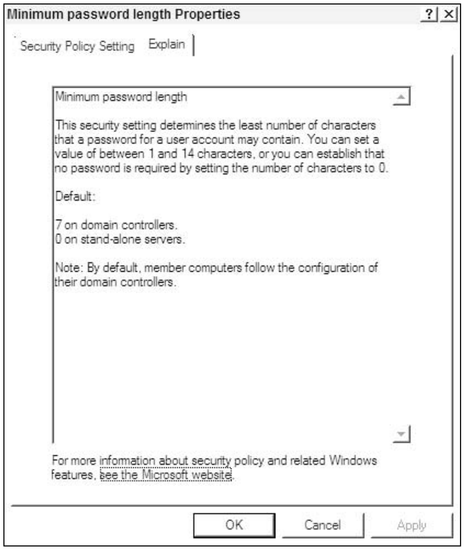


These groups and users have the specified permission for this GPO

Groups and users:

Name	Allowed Permissions	Inherited
Authenticated Users	Read (from Security Filtering)	No
Domain Admins (AD\Domain ...	Edit settings, delete, modify security	No
Enterprise Admins (AD\Enterp...	Edit settings, delete, modify security	No
ENTERPRISE DOMAIN CON...	Read	No
SYSTEM	Edit settings, delete, modify security	No

Figure 14-8: The Delegation tab controls the GPO permissions.



Minimum password length

This security setting determines the least number of characters that a password for a user account may contain. You can set a value of between 1 and 14 characters, or you can establish that no password is required by setting the number of characters to 0.

Default:

7 on domain controllers.
0 on stand-alone servers.

Note: By default, member computers follow the configuration of their domain controllers.

For more information about security policy and related Windows features, see the Microsoft website.

Figure 14-9: Read about settings on the Explain tab.

Linking GPOs

After you create and edit your GPO, the next step is to link that GPO to a site, a domain, or an OU object in AD. Here's how:

1. **In the GPMC, right-click the site, domain, or OU that you want to link the GPO to and then choose Link an Existing GPO from the contextual menu.**
2. **From the list of available GPOs that appears, select the GPO you want to link to the object.**
3. **Click OK.**

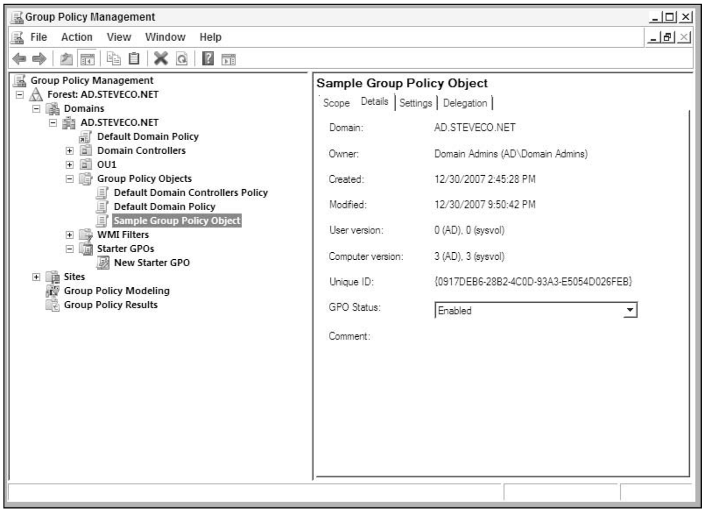
Disabling or deleting GPOs

There are several ways that you can disable a GPO. You can disable a GPO link by selecting the site, domain, or OU in GPMC; right-clicking the GPO, and then deselecting the Link Enabled option. This is useful when you've linked the GPO to multiple objects but want to disable the GPO on only one particular object. If you want to disable the GPO entirely, you can do that from the Details tab of the properties of the GPO (see Figure 14-10).

If you want to delete the GPO entirely, simply right-click the GPO and choose Delete.

TIP

Instead of *deleting* it, make *disabling* a group policy a standard practice. That way, if you find that you need to reinstate the GPO, you can simply enable the GPO again.



Sample Group Policy Object

Domain: AD.STEVECO.NET
Owner: Domain Admins (AD\Domain Admins)
Created: 12/30/2007 2:45:28 PM
Modified: 12/30/2007 9:50:42 PM
User version: 0 (AD), 0 (sysvol)
Computer version: 3 (AD), 3 (sysvol)
Unique ID: {0917DEB6-28B2-4C0D-93A3-E5054D026FEB}
GPO Status: Enabled
Comment:

Figure 14-10: The Details panel of the GPO properties.

Group policy reporting and modeling

One of the great things that the GPMC provides is reporting on group policies. Because you can deploy GPOs in multiple places (sites, domains, and OUs) and these GPOs become layered on top of each other, knowing the final results of the GPOs is important. This final result is referred to as the Resultant Set of Policy (RSoP). The GPMC provides two types of RSoP reports: RSoP reports of an existing user on a specified computer, and RSoP reports that show how the RSoP would change if certain aspects of the user or computer changed. This modeling allows you to see what the impacts are ahead of time. The generation of these reports is performed under the Group Policy Results container and the Group Policy Modeling container in GPMC.

Fine-Grained Password and Account Lockout Policies

In previous releases of Active Directory in 2000 and 2003, you could have only one Password Policy and Account Lockout Policy for a domain. You were forced to create separate domains for each group of users that needed different password and account lockout policies. This limitation resulted in cost increases for providing AD services, requiring additional DCs as well as administrative overhead. With Windows Server 2008, though, you can apply different password and account lockout policies to users in the same domain, thus reducing the need to deploy multiple domains. These policies can either be associated with individual users or security groups. They cannot be associated with an OU, though.

Unfortunately, setting up a domain to support multiple password policies is a bit involved and is also beyond what I can cover in this chapter. (You can find information on setting this up at the Microsoft Web site.) Just be aware that Windows Server 2008 does support this ability. However, just because you *can* do this doesn't mean you necessarily *should* do it. The downside of implementing multiple password/account lockout policies is increased complexity of your environment, which will increase your support costs.

Active Directory Auditing

In our increasingly security-conscious world, having the ability to track the changes made to an IT infrastructure is critical. When an unplanned change occurs, you need to discern whether the change was done mistakenly by an authorized administrator or whether the change came from another source (such as an internal or external hacker). If the change were a mistake, you can use this information to (hopefully) avoid this in the future. If the change came from another source, you can take steps to further secure your environment to prevent additional incidences.

Being able to track changes, and then determine who made the changes and when those changes were made, is collectively known as *auditing*. In Windows Server 2008, Active Directory Domain Services (as well as AD Lightweight Directory Services, or LDS) has greatly improved the ability to audit against changes in AD. Auditing has always been available in AD, but in 2008, the change events are documented in much more detail. The auditing is recorded in four different categories:

- **Directory service access:** This is simply the act of reading the directory.
- **Directory service changes:** These include any creation, modification, or deletion of objects and attributes in the directory.
- **Directory service replication:** This provides details of any attempt to replicate directory information.
- **Detailed directory service replication:** This category provides more details on the directory replication.

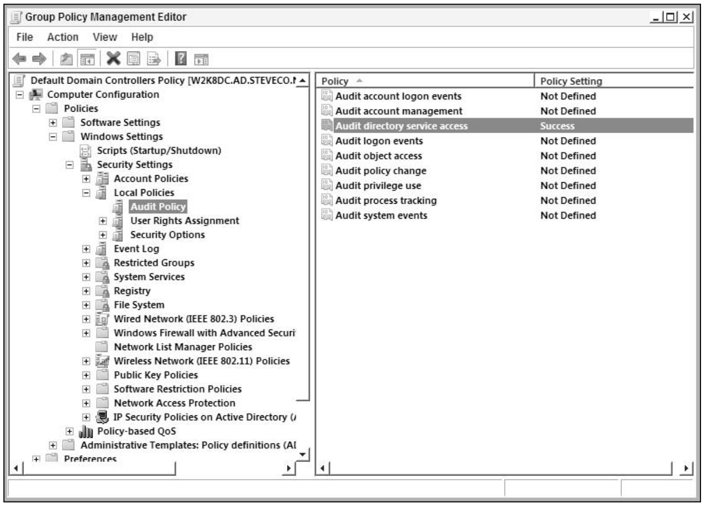
By default, DCs have the Directory Service Access Category enabled. To enable the other categories, you must set up auditing. Setting up auditing in AD is a two-step process.

- Enable auditing in the security policy of the DC(s) that you want to audit against.
- Edit the System ACL (SACL) that controls auditing on the objects that you want to audit.

To enable auditing on the DCs, you can do one of two things: Modify the Audit Directory Service Access setting in the security policy of the domain controller via GPMC (see Figure 14-11), or run the AUDITPOL.EXE command line tool. The advantage to using the AUDITPOL tool is that you can enable or disable each of the four categories I just mentioned. (If you do it in GPMC, all the categories are enabled, which might provide you with auditing information you are not interested in.)

To view the current audit settings with AUDITPOL, type in the following from a command prompt window:

```
Auditpol /get /category:"DS Access"
```


Group Policy Management Editor
File Action View Help
Default Domain Controllers Policy [W2K8DC.AD.STEVECO.I
Computer Configuration
Policies
Software Settings
Windows Settings
Scripts (Startup/Shutdown)
Security Settings
Account Policies
Local Policies
Audit Policy
User Rights Assignment
Security Options
Event Log
Restricted Groups
System Services
Registry
File System
Wired Network (IEEE 802.3) Policies
Windows Firewall with Advanced Securi
Network List Manager Policies
Wireless Network (IEEE 802.11) Policies
Public Key Policies
Software Restriction Policies
Network Access Protection
IP Security Policies on Active Directory (
Policy-based QoS
Administrative Templates: Policy definitions (AI
Preferences

Policy	Policy Setting
Audit account logon events	Not Defined
Audit account management	Not Defined
Audit directory service access	Success
Audit logon events	Not Defined
Audit object access	Not Defined
Audit policy change	Not Defined
Audit privilege use	Not Defined
Audit process tracking	Not Defined
Audit system events	Not Defined

Figure 14-11: Enabling Directory Service Auditing in GPMC.

You can then either enable or disable each of these categories with the following command:

```
Auditpol /set /subcategory:"<DS access category>"
```

With this tool, you can enable auditing of directory changes and disabling directory replication auditing, which may be a very likely scenario.

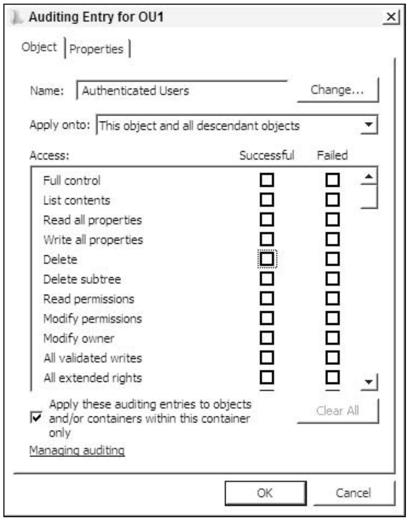
The next step in enabling auditing is to edit the SACL on the objects you want to audit. You can do this by editing the SACL on each of the objects, or you can edit the SACL of the container (either the domain or OU) of the objects so that the changes are propagated to the objects in that container. To edit the SACL of an OU, do the following:

1. **Open the AD Users and Computers console from either Server Manager or from the Administrative Tools group.**
2. **Right-click the OU that you want to audit on and choose Properties.**
3. **From the Security tab, select Advanced.**
4. **From the Auditing tab, select Add.**
5. **Decide which set of users you want to audit against (the user or group). Then click OK.**

   The safe bet is to use the Authenticated Users group to automatically include anyone who is making changes to AD.

6. **Select the types of operations you want to audit. Then click OK.**

   As you can see in Figure 14-12, you have a good number of operations to select from. If you simply want to audit against all operations, just select Full Control.



Auditing Entry for OU1

Object | Properties

Name: Authenticated Users [Change...]

Apply onto: This object and all descendant objects

Access:	Successful	Failed
Full control	□	□
List contents	□	□
Read all properties	□	□
Write all properties	□	□
Delete	□	□
Delete subtree	□	□
Read permissions	□	□
Modify permissions	□	□
Modify owner	□	□
All validated writes	□	□
All extended rights	□	□

☑ Apply these auditing entries to objects and/or containers within this container only

[Clear All]

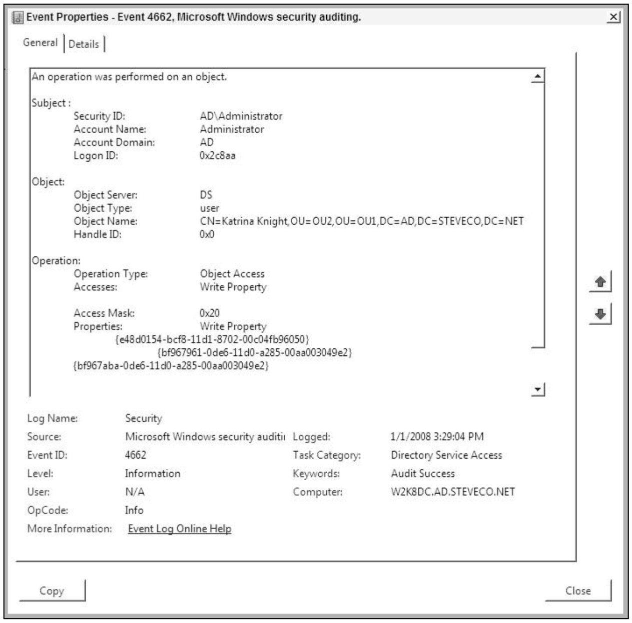
Managing auditing

[OK] [Cancel]

Figure 14-12: The Auditing System ACL.

If you're wondering why enabling auditing in AD is so complicated, you can thank Microsoft. Really. Microsoft purposely did not create a simple interface for enabling detailed auditing of AD. One of the downsides of auditing is that it creates a performance impact on the DCs because each access, modification, or replication has to be logged. So, by making the enable process somewhat involved, you avoid the temptation to enable auditing without knowing or without considering the impact that auditing creates. In other words, if you have to research how to enable auditing, you're likely going to discover what impacts the auditing has.

When you set up auditing, you can open the Security log in the Event Viewer to view the individual events that have been captured. Figure 14-13 depicts an audit entry that shows a change written to a user object that was performed by the Administrator account.


Event Properties - Event 4662, Microsoft Windows security auditing.
General
Details
Copy
Close

```
An operation was performed on an object.

Subject :
        Security ID:            AD\Administrator
        Account Name:           Administrator
        Account Domain:         AD
        Logon ID:               0x2c8aa

Object:
        Object Server:          DS
        Object Type:            user
        Object Name:            CN=Katrina Knight,OU=OU2,OU=OU1,DC=AD,DC=STEVECO,DC=NET
        Handle ID:              0x0

Operation:
        Operation Type:         Object Access
        Accesses:               Write Property

        Access Mask:            0x20
        Properties:             Write Property
                {e48d0154-bcf8-11d1-8702-00c04fb96050}
                        {bf967961-0de6-11d0-a285-00aa003049e2}
        {bf967aba-0de6-11d0-a285-00aa003049e2}
```

Log Name:	Security		
Source:	Microsoft Windows security auditi	Logged:	1/1/2008 3:29:04 PM
Event ID:	4662	Task Category:	Directory Service Access
Level:	Information	Keywords:	Audit Success
User:	N/A	Computer:	W2K8DC.AD.STEVECO.NET
OpCode:	Info		
More Information:	Event Log Online Help		

Figure 14-13: An audit entry showing a change to a user object.

Chapter 15

Maintaining Active Directory

In This Chapter

- Understanding the AD DS files
- Defragging AD
- Backing up and restoring Active Directory
- Restartable AD
- Other AD tools

As with most deployed IT systems, Active Directory requires some routine and not-so-routine maintenance tasks to be executed regularly to keep the system healthy. Active Directory Domain Services has files that are critical to its functionality. Understanding what these files are and how to work with them is a requirement if you're going to support AD DS. Fortunately, maintaining the Active Directory database isn't difficult, and Microsoft provides some great tools to help you. Don't let lingo such as tombstones and fragmentation intimidate you, because the database is largely self-sufficient. With a bit of simple maintenance, you can be certain of having a healthy, happy database!

Database Files

The database engine in Active Directory is the *Extensible Storage Engine (ESE),* which is based on the original Jet database engine that Exchange 5.5 and WINS used. The following key files make up the Active Directory database:

- Database file
- Transaction log files

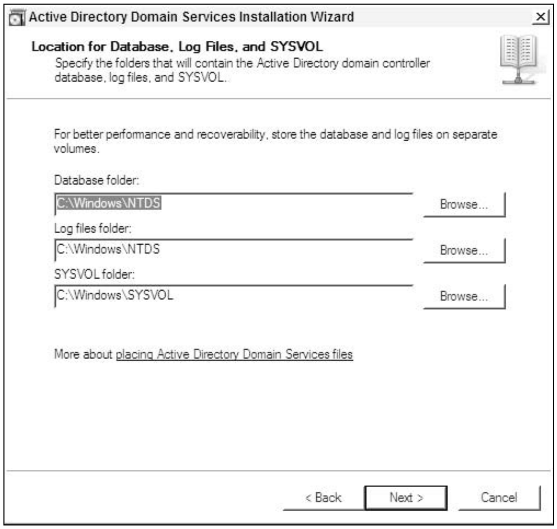
Each domain controller has a set of these files. In addition to these files, each domain controller has a SYSVOL folder in which the policy files of the Group Policy Objects are stored. All these folders and files are critical to the proper functionality of a domain controller.

The actual database file for AD DS is named `NTDS.DIT` (typically located in the `C:\WINDOWS\NTDS` directory) Within this file are the various partitions (schema, configuration, domain naming) as well as all the objects that have been created within the domain. Also, any additionally created application partitions are stored in this file. The transaction log files are the set of files associated with the `NTDS.DIT` file that contain a sequential list of each transaction executed against the DIT file. These log files are used to

- Provide a way to replay changes to the database if the DC loses power unexpectedly or the database file is found to be in an inconsistent state because of a server crash.
- Facilitate AD database backups while online by allowing for changes to the database file to be cached during the backup.

Specifying the location of the database files

When you run the DCPROMO.EXE tool (from the Start menu Run option) to create a new domain controller, one of the items that you must provide is the location of the folder that will store the database, transaction logs, and SYSVOL. Figure 15-1 shows the screen in the DCPROMO tool in which you can specify where these files should be placed on your domain controller.



Active Directory Domain Services Installation Wizard

Location for Database, Log Files, and SYSVOL

Specify the folders that will contain the Active Directory domain controller database, log files, and SYSVOL.

For better performance and recoverability, store the database and log files on separate volumes.

Database folder: C:\Windows\NTDS

Log files folder: C:\Windows\NTDS

SYSVOL folder: C:\Windows\SYSVOL

More about placing Active Directory Domain Services files

Figure 15-1: Use the Location for Database, Log Files, and SYSVOL screen to specify a location for your database files.

Optimizing domain controller disk performance

For best performance, Microsoft recommends that you place the log files on a physical disk separate from where you keep the database (`NTDS.DIT`). Doing so provides optimum disk performance because the read-and-write functions of the log files and database aren't vying for the same read/write head of the physical hard drive. For the same reason, Microsoft further recommends that you place the database and log files on separate physical disks from the operating system files and from the page file (and if possible even use separate hardware disk controllers for the database and log files).

Assuming that you have adequate physical disks, the following list describes the ideal disk management configuration:

- Two mirrored (RAID1) physical disks to hold the operating system files
- Two mirrored (RAID1) physical disks to hold the transaction log files
- And either a RAID1 or RAID5 disk array to hold the database

You can even separate the SYSVOL directory to a separate drive if you have the luxury of having another set of drives on your domain controller. Just remember the most important thing is to keep the transaction log files on their set of hard drives.

If you ever need to find where the database and log files are on an existing domain controller, simply do a search by filename (`NTDS.DIT` for the database file and `EDB.LOG` for the transaction log files). You can also get information about these files (including their location) by using NTDSUTIL, but only when the domain controller services are offline. (I show how to take the domain controller services offline and view the file information in the "Defragmenting the Database" section, coming up.)

How the database and log files work together

The database and log files work together to provide the store service that Active Directory depends on. Every time a write operation is executed against the AD store, the following process occurs:

1. The transaction is written to the transaction log file.
2. The transaction is then executed against the cached database in the memory of the domain controller.
3. The write transaction is actually committed against (written to) the database file.

The transactions are written to the log files first so that if there's a problem and the domain controller, for example, loses power before the transaction can be committed to the database, there's still a copy of the transaction in the log file that can be replayed when the domain controller is brought back up. The log file is always 10MB in size when initially allocated. But this is initially a 10MB empty file. As transactions are committed to the database, each transaction is written to the log file. When the file is full, another file is then created that is immediately ready to continue receiving transactions. But AD DS uses a scheme to manage the log files called *circular logging*.Instead of continuing to create more and more new transaction log files, all the log files that do not contain pending transactions to be committed to the database are deleted.

TIP

If you're familiar with Microsoft Exchange, AD DS uses basically the same database engine. In Exchange, it's recommended that you don't use circular logging, whereas in AD DS, it's recommended that you leave circular logging on. Turning off circular logging is recommended in Exchange so that data-bases can be recovered by replaying the contents of the transaction log files. This isn't a serious issue with AD DS because the rate of transactions is nowhere near as great as in Exchange. Also, you can recover a domain controller, when necessary, simply by directory replication of the domain controller with the other preexisting domain controllers in that domain. (Of course, this assumes that you didn't set up a domain with only one domain controller in it!)

Defragmenting the Database

Defragmenting (or *defragging*) the database file rearranges the pages of the database file into a more compact format. This process is very similar to how you defragment the hard drive of your PC.

You can defragment the database either online or offline. Online defragmenting rearranges the data but doesn't release any freed space to the file system. In other words, within the `NTDS.DIT` file, the used and unused space is consolidated but the actual size of the file doesn't change. Online defragmenting is a part of a regular automated process that normally runs every 12 hours on each domain controller. In addition to the defragmenting, a garbage collection is initiated. When an object is deleted from the database, the object isn't actually deleted but instead is marked as *tombstoned*. A tombstoned object is an object marked for removal from the database and hidden from any queries against the directory. Each tombstoned object remains in the database for the number of days defined as the tombstone lifetime. The garbage collection process that runs every 12 hours examines all the tombstoned objects to see whether any have exceeded their tombstone lifetime value. If so, that object is deleted from the database.

You might wonder why a deleted object is marked as tombstoned rather than deleted. The reason is because of the multimaster replication nature of the domain controllers. Tombstoning an object allows for the deletion action to be replicated to the other domain controllers. Tombstones also prevent another problem: keeping deleted objects from reappearing due to an AD restore. Imagine that a domain controller crashes and you decide to rebuild the DC from the last backup. But before you can complete the restore, an object in the directory is deleted (that is, marked as tombstoned). The problem is that the backup you are about to restore still contains the object that was just deleted. If tombstones didn't exist, you would have the potential of that deleted object reappearing in the directory after the restoration of the crashed DC is completed. But by marking the deleted object with a tombstone, the tombstoned object will be replicated to the restored DC thereby marking it as deleted on that DC as well.

Originally, in Windows 2000 Server and Windows Server 2003, the default tombstone lifetime was 60 days. In Windows Server 2003 Service Pack 1 and Windows Server 2008, this period is 180 days. The tombstone lifetime dictates both the maximum time that a domain controller can be offline and the maximum age at which an AD backup is still valid. Because restoring a backup older than the tombstone lifetime has the potential to make deleted objects reappear in the directory, extending the lifetime increases the number of backups that can be used to restore a domain controller.

Of course, a large tombstone lifetime can be a detriment when you're deleting a large number of objects and you need to decrease the size of the database. You might not want to wait 180 days before you can regain the disk space. Fortunately, you can change the default tombstone lifetime value by using the ADSIEDIT tool.

If you're going to change the tombstone lifetime value, be very sure that all your domain controllers are online and functioning well. If any of the domain controllers has been either offline or not replicating for a period greater than your new tombstone lifetime, you will create corruption in the directory. So be careful!

To change the default tombstone lifetime, follow these steps:

1. **At a domain controller's console, choose Start and then Run.**

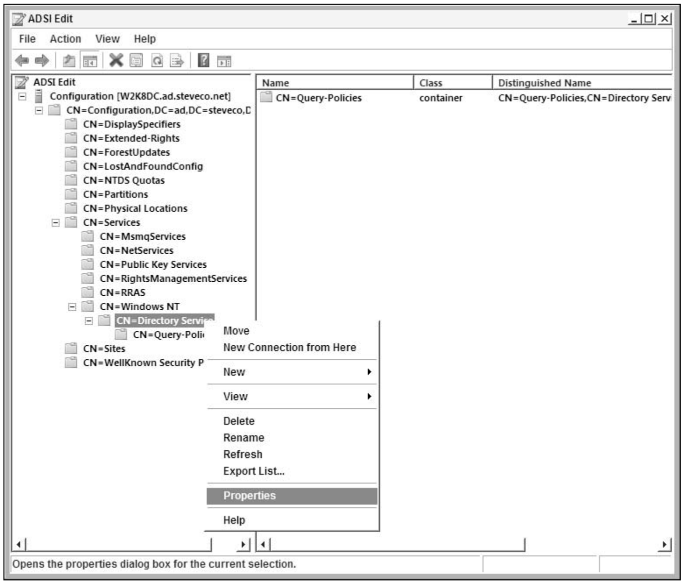
   You must be logged in to an account that's a member of the Enterprise Admins group.

2. **Type** ADSIEDIT.MSC **and then press Enter.**
3. **Right-click the ADSIEDIT root object in the left pane and select Connect To.**
4. **With the Select a Well Known Naming Context selected, change the context in the drop-down box to Configuration and then click OK.**

5. **Navigate to the CN=Directory Service,CN=Windows NT,CN=Services, CN=Configuration,DC=*`<your domain name>`* container and choose Properties (see Figure 15-2).**

6. **Right-click the Directory Services container and select Properties.**

7. **Scroll down the list of attributes until you get to the tombstoneLifetime attribute. Select it and then click the Edit button. Enter the new number of days that you want to use for the tombstone lifetime.**



Name	Class	Distinguished Name
CN=Query-Policies	container	CN=Query-Policies,CN=Directory Serv

Opens the properties dialog box for the current selection.

Figure 15-2: Locating the `Directory Services` container in the Configuration partition.

If you want to change the garbage collection frequency, that attribute is also in this same container (it's called `garbageCollPeriod`).

Online defragmentation

Online defragmentation doesn't reduce the size of the database file, though. Defrag only consolidates the used and unused space within the file. To gain disk space from the database file, you have to execute an offline defragmentation. Offline defragmenting rearranges the data and then releases the freed space to the file system. To perform an offline defragmentation, you must first take the domain controller services offline. You do this by entering the server into Directory Services Restore mode (DSRM).

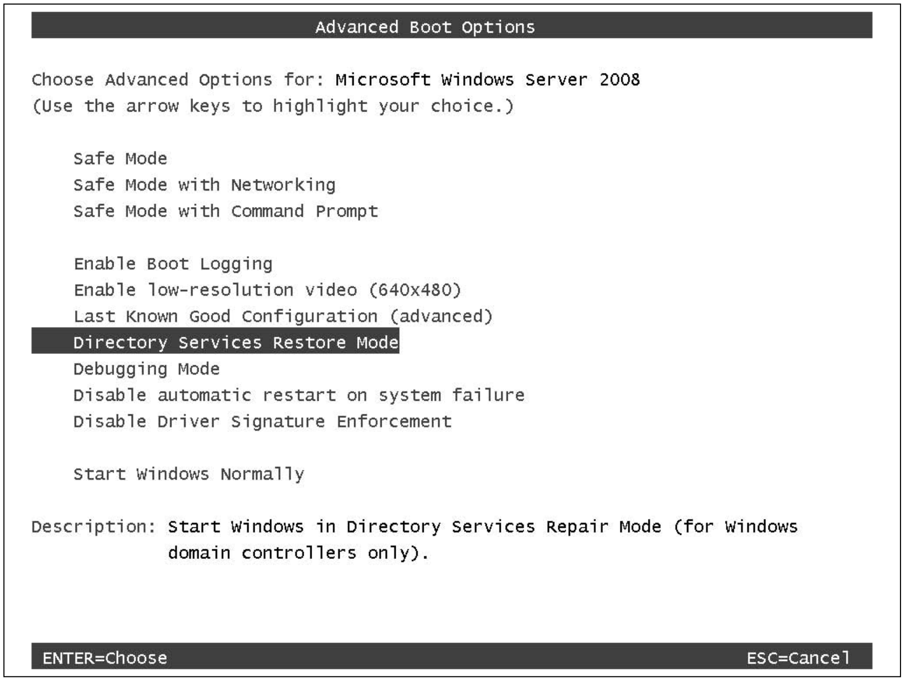
To enter DSR mode, follow these steps:

1. **Press F8 during the server's boot sequence to enter the Advanced Boot Options screen (see Figure 15-3).**
2. **Choose Directory Services Restore Mode from the list of options that appears.**

   The server continues to boot. Note that you must log in to the local administrator ID with the Directory Services Restore mode password that you specified at the time you ran DCPROMO.

TIP

Later in this chapter, I cover a new feature in Windows Server 2008 that allows you to bypass restarting the server and entering DSRM just to do an offline defragmentation.



```
                       Advanced Boot Options

Choose Advanced Options for: Microsoft Windows Server 2008
(Use the arrow keys to highlight your choice.)

    Safe Mode
    Safe Mode with Networking
    Safe Mode with Command Prompt

    Enable Boot Logging
    Enable low-resolution video (640x480)
    Last Known Good Configuration (advanced)
    Directory Services Restore Mode
    Debugging Mode
    Disable automatic restart on system failure
    Disable Driver Signature Enforcement

    Start Windows Normally

Description: Start Windows in Directory Services Repair Mode (for Windows
             domain controllers only).



ENTER=Choose                                                   ESC=Cancel
```

Figure 15-3: Entering into Directory Services Restore mode.

Offline defragmentation

Offline defragmenting creates a new, compacted copy of the database in a different directory. You can archive the old database file (`C:\WINNT\NTDS\NTDS.DIT`) to a separate directory and replace it with the compacted version, which also carries the name `NTDS.DIT`. The NTDSUTIL command line tool executes the offline defragmentation. You can use the following steps to execute an offline defrag by using NTDSUTIL in Directory Services Restore mode:

1. **After logging in to the local administrator ID in DSRM, open a command prompt window and enter** NTDSUTIL**.**

2. **Specify which directory instance you want to work with (in this case, the NTDS instance). At the NTDSUTIL prompt type**

   ```
   ACTIVATE INSTANCE NTDS
   ```

3. **Enter into the FILE mode of NTDSUTIL by typing**

   ```
   FILES
   ```

4. **You need to know is where the NTDS.DIT and log files are stored. At the file maintenance prompt, type the following command to get the directory paths to these files:**

   ```
   INFO
   ```

   The directory paths for both the database and the log files appear. Take note of the current directory for the NTDS.DIT file as you will need to know this for a later step.

5. **At the file maintenance prompt, type the following command to perform the offline defrag:**

   ```
   COMPACT TO <dir>
   ```

   <dir> is a directory on the server where you create the new defragged `NTDS.DIT` file. (Don't specify the current database directory!) Also take note of the two commands that you are prompted to execute after you exit the NTDSUTIL tool. These are to copy the newly compressed file to where the current NTDS.DIT file is and then to delete the existing log files. You will execute these commands in Step 8.

6. **Type** QUIT **and press <enter> to exit the FILE mode of the NTDSUTIL tool.**

7. **Type** QUIT **and press <enter> to exit the NTDSUTIL tool.**

8. **Copy over the old database file and delete the old log files. The commands for doing this were provided to you in Step 5.**

9. **Close the command prompt window and restart the domain controller.**

Keep the original database file until you're certain that the compacted file loads correctly.

Backing Up the Active Directory Database

You can take a couple of approaches to backing up the Active Directory database, as the following list describes:

- Use a third-party backup application that enables you to perform online database backups.
- Use the backup utility that comes with Windows Server software.

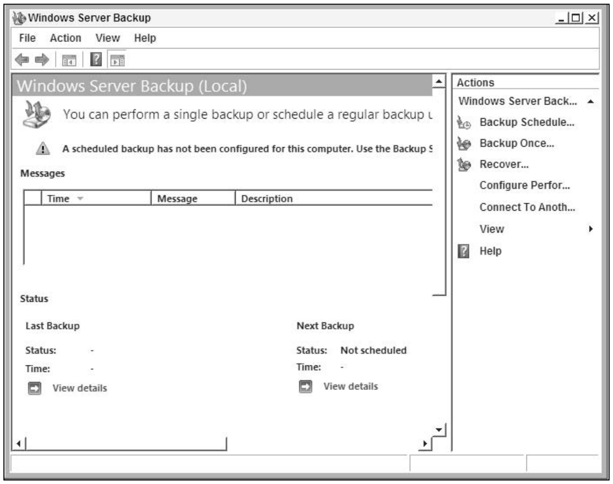
Numerous third-party backup applications are available for the Windows Server operating system and Active Directory. If you decide to use one of these applications, make sure that you choose one that specifically states that it can perform online database backups of Active Directory. These applications are sometimes sold as product add-ons, or *agents,* that you must purchase in addition to the backup application software.

Although I definitely recommend the use of a third-party backup application, you can always opt to use the backup utility that comes with Windows Server software. This utility doesn't necessarily have all the features that the third-party tools have, but it's a perfectly good tool for backing up a domain controller as well as the Active Directory database. The backup tool that comes with Active Directory has been updated in Windows Server 2008.

The first step is to install the Server Backup feature with the Server Manager tool. (Yes, Server Backup isn't a server role like many of the items you've installed; it's a feature.) To start the backup utility, choose Start➪Administrative Tools➪Windows Server Backup. This brings you to the Windows Server Backup console (see Figure 15-4).

The backup utility is easy to use. On the Actions pane you have several backup/restore options to choose from:

- Backup Schedule (to create scheduled backups)
- Backup Once (to initiate a single manual backup)
- Recover (to recover the entire server or a portion of the server from a backup)



You can perform a single backup or schedule a regular backup u

A scheduled backup has not been configured for this computer. Use the Backup S

Status

Last Backup

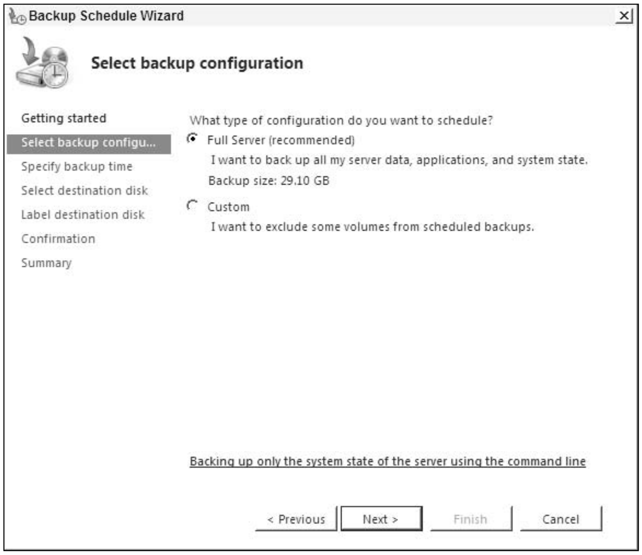
Status: -
Time: -
View details

Next Backup

Status: Not scheduled
Time: -
View details

Figure 15-4: The Windows Server Backup console.

The best approach is to simply perform a full backup of the domain controller. This is done by selecting the Backup Once action from the action pane or by selecting Backup Schedule if you want to create a reoccurring backup. When asked for the backup configuration you want, select Full Server, which backs up everything on the server including the Active Directory database. (See Figure 15-5.)



Select backup configuration

What type of configuration do you want to schedule?

● Full Server (recommended)
I want to back up all my server data, applications, and system state.
Backup size: 29.10 GB

○ Custom
I want to exclude some volumes from scheduled backups.

Figure 15-5: Selecting a Full Server backup from the Windows Server Backup tool.

The Windows Server Backup tool is also available from the WBADMIN command line utility, which allows you to set up more complex maintenance scripts that include backing up of the server. Also if you need to perform a backup with Windows Server Backup from a Server Core DC your only option is to use WBADMIN.

WARNING!

If you're familiar with backing up Active Directory in previous versions of the Windows Server OS, you need to understand that system state backups are no longer sufficient as a minimal way of backing up Active Directory. In Windows Server 2008, the system state even on a domain controller is designed for the restoration of the OS, not Active Directory. Therefore, if you want to back up Active Directory on a Windows Server 2008 DC, make sure that you back up all critical disk volumes on your server as well as the system state.

WARNING!

If you want to schedule reoccurring backups, only an ID with administrator authority can do this. Members of the Backup Operators group, although capable of running a manual backup, do not have the authority to schedule backups. This is true regardless of whether you use the GUI or command line–based Windows Server Backup tool.

Restoring Active Directory

Of course, like many server applications, backing up Active Directory and its associated database is the easy part. The more difficult part is the restoration of that data. The two types of Active Directory restores are non-authoritative and authoritative. Each is covered in the following sections.

Non-authoritative restore

The goal of executing a non-authoritative restore is simply to restore the Active Directory database on a domain controller from a backup. This is achieved by executing the restore of an existing full backup of the domain controller via either the GUI interface or the WBADMIN command line tool.

If the domain controller is functioning normally and you need only to restore the database, enter DSR mode and then execute a restore of the `NTDS.DIT` file from the backup. Also, make sure that you delete the log files as well. (I describe entering DSR mode and the offline defragmentation process in the earlier "Defragmenting the Database" section.) If the domain controller's OS is not functional, then you must reinstall the Windows Server OS and execute a full server restore from the last server backup.

In reality, non-authoritative restores of domain controllers are typically executed only when the domain contains just one domain controller. If you need to re-create a domain controller for an existing domain with other DCs that are functioning normally, it is recommended that you simply reinstall the OS on the DC and then run DCPROMO on the server to promote the DC back into the domain. This method is often a faster way to rebuild the DC. If you do this, make sure that you delete all objects in Active Directory that correspond to this DC before reinstalling the OS; otherwise, the installation will fail.

Authoritative restore

Because Active Directory administrators are human, there are times when objects (or even entire OU structures) are accidentally deleted. This is where authoritative restores come in handy. With authoritative restores, your goal is to restore one or more deleted objects to the directory. The first step is to perform a non-authoritative restore with a backup of the database that contains the objects that you want to restore. After the database is restored, but before you exit DSRM, you mark those previously deleted objects as being authoritative. By doing this you prevent these objects from being deleted when the tombstoned versions of those objects on the other domain controllers replicate to this DC. Instead, the domain controller on which you perform the restore replicates the objects marked as authoritative out to the other domain controllers, effectively deleting the tombstoned versions of the objects and thereby making the objects show in the directory service again.

Mark the objects as authoritative via the NTDSUTIL, as follows:

1. **After you restore the AD database while still in DSRM, open a command prompt and type**

   ```
   NTDSUTIL
   ```

2. **From the NTDSUTIL prompt type**

   ```
   AUTHORITATIVE RESTORE
   ```

3. **At the Authoritative Restore prompt, specify which object(s) you want to mark authoritative.**

   To do this you need to know the distinguished name of the objects you want to restore. For example, if you need to restore a user object called `JohnS` in the `Users` container of the `STEVECO.NET` domain, the command would be

   ```
   RESTORE OBJECT CN=JOHNS,CN=USERS,DC=STEVECO,DC=NET
   ```

   You can also restore an entire OU and all the objects in that OU with a single command. To illustrate, the command to restore the `ENG` OU in the `STEVECO.NET` domain would be

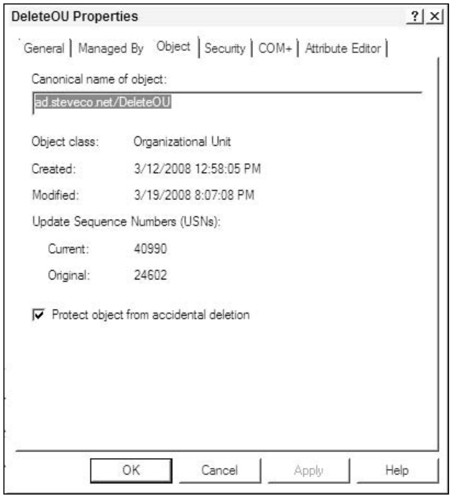
   ```
   RESTORE SUBTREE OU=ENG,DC=STEVECO,DC=NET
   ```

4. **Exit the NTDSUTIL tool and then restart the domain controller in Normal mode.**

   After the AD directory service starts, the marked objects replicate out to all the other domain controllers and are available in AD again.

Preventing accidental deletions

The ability to perform authoritative restores when objects need to be restored to the directory is great, but it doesn't mean that admins enjoy doing it or users enjoy waiting for the restore to complete. One new feature in Windows Server 2008, allows you to protect individual objects and containers from accidental deletion (see Figure 15-6). The check box for this option is located on the Object panel of each object in the AD Users and Computers console. What this option is really doing is adding an entry to the object's ACL that denies all users the Delete ability on the object. If this option is enabled on a container, such as an OU, then that OU can't be deleted. However, this doesn't directly protect the objects in the OU from being deleted because this permission is configured against the OU and is not inherited by objects in the OU. So, if you decide to use this feature, you need to enable it on each object you want to protect from accidental deletion.



Canonical name of object:
ad.steveco.net/DeleteOU

Object class: Organizational Unit

Created: 3/12/2008 12:58:05 PM

Modified: 3/19/2008 8:07:08 PM

Update Sequence Numbers (USNs):

Current: 40990

Original: 24602

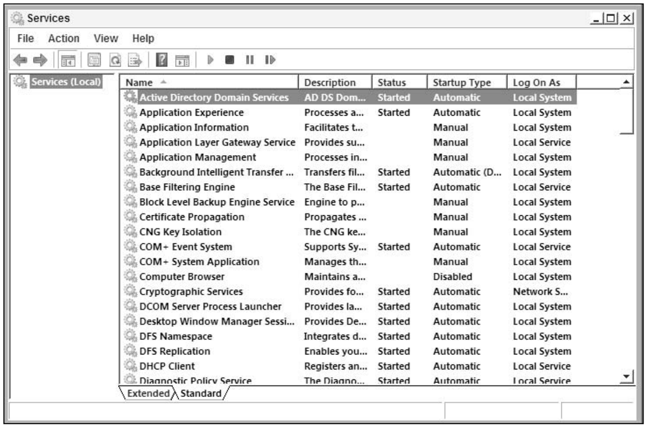
☑ Protect object from accidental deletion

Figure 15-6: The accidental deletion prevention option.

Restartable Active Directory

One of the useful things that Microsoft decided to do with AD DS in Windows Server 2008 was to provide more separation between the software providing the domain controller role on a server (that is, domain controller) and that server's local operating system. In earlier releases, there was no clean way to stop a domain controller from acting as a domain controller without turning off all local authentication services on the server (provided by the Netlogon service, or LSASS). In 2008, Microsoft has provided for the AD DS service to be separate from the Netlogon service on a domain controller. This means that you now have the ability to stop and start AD DS on a server without disabling all authentication services on the domain controller.

To stop the AD DS service on the domain controller, open the Services console from the Administrative Tools group. At or near the top of the list of services, you see the Active Directory Domain Services service. By stopping this service, you're effectively disabling the server from acting as a domain controller. When you stop this service, this server becomes like any other member server in the domain (assuming that other online DCs in this domain are still functioning). Having AD DS separated as a service on the domain controller means administrative tasks, such as offline defragmentations, security updates, and other tasks that in the past would require restarting the DC and going into DSRM, can now be done simply by stopping the AD DS service and performing the maintenance, which saves time and effort. (See Figure 15-7.)



Name	Description	Status	Startup Type	Log On As
Active Directory Domain Services	AD DS Dom...	Started	Automatic	Local System
Application Experience	Processes a...	Started	Automatic	Local System
Application Information	Facilitates t...		Manual	Local System
Application Layer Gateway Service	Provides su...		Manual	Local Service
Application Management	Processes in...		Manual	Local System
Background Intelligent Transfer ...	Transfers fil...	Started	Automatic (D...	Local System
Base Filtering Engine	The Base Fil...	Started	Automatic	Local Service
Block Level Backup Engine Service	Engine to p...		Manual	Local System
Certificate Propagation	Propagates ...		Manual	Local System
CNG Key Isolation	The CNG ke...		Manual	Local System
COM+ Event System	Supports Sy...	Started	Automatic	Local Service
COM+ System Application	Manages th...		Manual	Local System
Computer Browser	Maintains a...		Disabled	Local System
Cryptographic Services	Provides fo...	Started	Automatic	Network S...
DCOM Server Process Launcher	Provides la...	Started	Automatic	Local System
Desktop Window Manager Sessi...	Provides De...	Started	Automatic	Local System
DFS Namespace	Integrates d...	Started	Automatic	Local System
DFS Replication	Enables you...	Started	Automatic	Local System
DHCP Client	Registers an...	Started	Automatic	Local Service
Diagnostic Policy Service	The Diagno...	Started	Automatic	Local Service

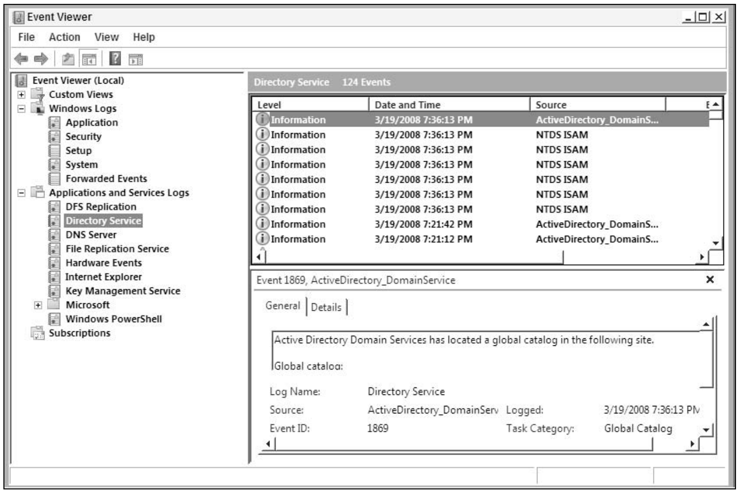
Figure 15-7: The restartable AD DS service.

Other Tools for Maintaining AD

In closing this chapter, I want to cover some other tools that you should be very familiar with if you're supporting Active Directory Domain Services. This certainly isn't the entire list of tools that Microsoft has made available, but it includes some of the more popular ones.

Event Viewer

Your primary tool for troubleshooting Active Directory should always be the Event Viewer tool. If an individual domain controller is experiencing problems with providing AD services, these conditions are probably being logged within Event Viewer. If you're familiar with Event Viewer from earlier versions of the Windows Server product, the version in 2008 looks a bit different. Previously, all AD messages were grouped into the Applications group, and if you wanted to view only the AD messages, you had to create a custom filter. In the Windows Server 2008 Event Viewer, Active Directory has a separate grouping built in without the need to create a filter. (See Figure 15-8.)



Directory Service 124 Events

Level	Date and Time	Source
Information	3/19/2008 7:36:13 PM	ActiveDirectory_DomainS...
Information	3/19/2008 7:36:13 PM	NTDS ISAM
Information	3/19/2008 7:36:13 PM	NTDS ISAM
Information	3/19/2008 7:36:13 PM	NTDS ISAM
Information	3/19/2008 7:36:13 PM	NTDS ISAM
Information	3/19/2008 7:36:13 PM	NTDS ISAM
Information	3/19/2008 7:36:13 PM	NTDS ISAM
Information	3/19/2008 7:21:42 PM	ActiveDirectory_DomainS...
Information	3/19/2008 7:21:12 PM	ActiveDirectory_DomainS...

Event 1869, ActiveDirectory_DomainService

Active Directory Domain Services has located a global catalog in the following site.

Global cataloa:

Log Name: Directory Service

Source: ActiveDirectory_DomainServ Logged: 3/19/2008 7:36:13 PM

Event ID: 1869 Task Category: Global Catalog

Figure 15-8: Event Viewer in Windows Server 2008.

Snapshots and the AD Database Mounting Tool

Here is a cool new feature. In Windows Server 2008, you now have the ability to take *snapshots,* or *shadow copies,* of the AD database and log files. This snapshot is a copy of the entire volume in which the database and log files are stored when the snapshot is taken. These snapshots are taken by using the SNAPSHOT option within the NTDSUTIL tool. These snapshots can be used as alternatives to the normal backup and restore processes. To take a snapshot, follow these steps:

1. **Open a command prompt and run NTDSUTIL.**
2. **At the NTDSUTIL prompt, type**

   ```
   SNAPSHOT
   ```

3. **At the SNAPSHOT prompt, type the following command to work with the AD DS database. (Yes, you can use this procedure to work with AD LDS instances, too.)**

   ```
   ACTIVATE INSTANCE NTDS
   ```

4. **At this point, you're ready to create a snapshot. This is done simply be typing**

   ```
   CREATE
   ```

   You're provided with a long number — a GUID — that's associated with the snapshot. Figure 15-9 shows an example of a GUID. You can use this GUID to refer to this snapshot in future commands. But fortunately you can use a shortcut. If you type in the command

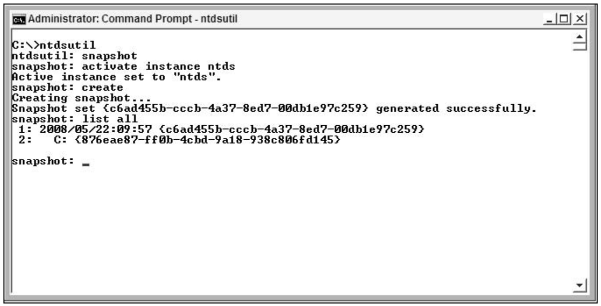
   ```
   LIST ALL
   ```

   You are provided with a list of the current available snapshots. In Figure 15-9, the GUID of the snapshot is listed as number 1. You can refer to the snapshot by using this number as well.

Although no scheduling ability is built in to NTDSUTIL, an administrator can create a script that creates a snapshot and then execute that script on a scheduled basis. To restore from a snapshot, you can mount a previously taken snapshot within the SNAPSHOT menu of NTDSUTIL and then access the NTDS.DIT file from the mounted snapshot.

TIP

Snapshots are also a great way to create a copy of the directory database to be used for an Install from Media (IFM) deployment of a domain controller, which I cover in Chapter 7.



```
C:\>ntdsutil
ntdsutil: snapshot
snapshot: activate instance ntds
Active instance set to "ntds".
snapshot: create
Creating snapshot...
Snapshot set {c6ad455b-cccb-4a37-8ed7-00db1e97c259} generated successfully.
snapshot: list all
 1: 2008/05/22:09:57 {c6ad455b-cccb-4a37-8ed7-00db1e97c259}
 2:   C: {876eae87-ff0b-4cbd-9a18-938c806fd145}

snapshot: _
```

Figure 15-9: Creating a snapshot by using NTDSUTIL.

While you create more and more snapshots, knowing which snapshot to use for a recovery can get confusing. For this reason, Microsoft provides DSAMAIN.EXE, the Active Directory database mounting tool. With this tool, you can take a copy of an active domain controller's `NTDS.DIT` file (from either a snapshot or from a restored copy of the file from a server backup) and make the contents available for reading via LDAP without interfering with the production directory running on the domain controller. You mount a database with the following command:

```
DSAMAIN /dbpath <database file path> /ldapport: <port#>
```

<*database file path*> is the full file path to where the `NTDS.DIT` file you want to mount is located, and <*port#*> is the TCP port number you want to mount the database to. Note that you can't use 389 here because that's already used by the production AD DS directory running on the domain controller. Best practice is to use a number higher than 10,000 to prevent any conflicts. After the database is mounted, you can use such tools as LDP or even AD Users and Computers to view the contents of that database file to determine whether it contains the objects that you need to restore.

REPADMIN

When you're troubleshooting an AD DS forest, very often you're dealing with issues surrounding the replication of the directory between domain controllers and, more specifically, with DCs that are not replicating with each other or are not in sync with each other . Although the technology that makes the replication work is solid, from time to time you have to diagnose and resolve issues in this area. REPADMIN.EXE is a command line tool that provides information about the replication health of a domain controller and provides some functions that help resolve existing issues. REPADMIN offers too many options for me to cover here, but if you're having replication issues, I strongly urge you take a closer look at this tool.

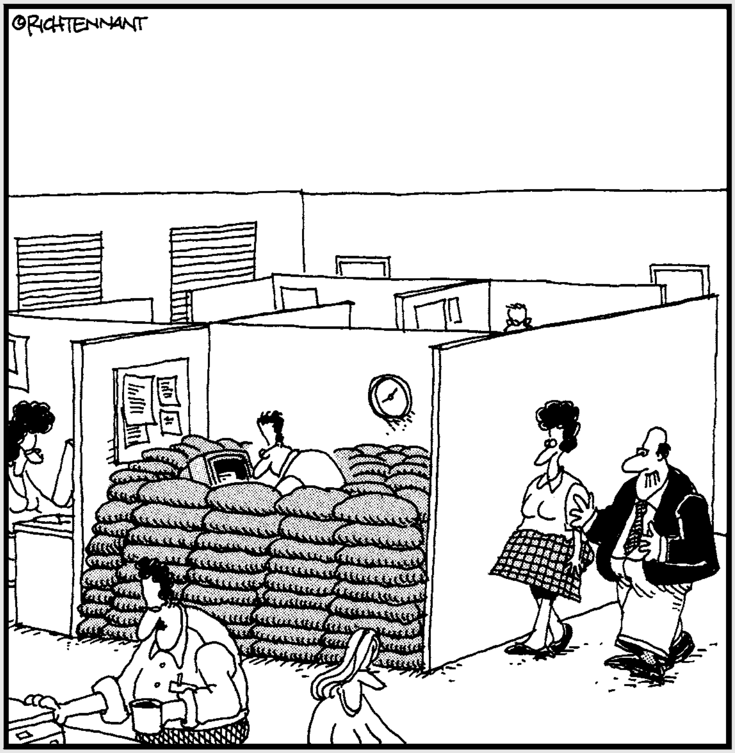
TIP

I cover this tool in more detail in Appendix A.

Part V

The Part of Tens

The 5th Wave By Rich Tennant



"I assume everyone on your team is on board with the proposed changes to the office layout."

In this part . . .

The Part of Tens is a tradition in *For Dummies* books. It gives us the opportunity to add a little fun to the technical topics. Here, you discover ten great Internet resources for additional information on Active Directory. You also find — concisely summarized — the ten most important points about Active Directory design, and finally, ten troubleshooting tips to help you work through problems related to Active Directory.

Chapter 16

The Ten Most Important Active Directory Design Points

In This Chapter

- Plan, plan, plan!
- Design AD for the administrators
- What's your forest scope?
- Often a single domain is enough
- AD is built on DNS
- Your logical AD structure isn't based on network topology
- Limit AD schema modifications
- Understand your identity management needs
- Place DCs and GCs near users
- Keep improving your design

Active Directory is a big subject (big enough to write a book about, apparently!). Although I've covered a number of design principles and guidelines, I think the top ten design tips bear repeating. I hope this list proves helpful to you — sort of like a cheat sheet! So — drum roll, please — the following are my ten most important Active Directory design points!

Plan, Plan, Plan!

The single most important thing to know about Active Directory is that you can't just turn it on and expect it to operate in an efficient manner. To function effectively, an Active Directory implementation requires lots of planning. Active Directory can lower total cost of ownership by reducing hardware costs and enabling centralized administration. However, if the implementation isn't planned correctly, it ends up costing you money instead.

In Chapter 2, I cover the need to gather both business and technical information about the environment that you're going to deploy within. Being armed with this information is the only way you're going to create an AD design that meets the needs of your company.

After gathering this information, you can design the various elements of an overall Active Directory design including:

- The namespace and DNS design
- The logical AD structure
- The physical AD topology
- The domain/OU delegation strategy
- The security policies
- Disaster Recovery plans

Design AD for the Administrators

When you're designing Active Directory, particularly the logical structures of forests, domains, and organizational units, keep in mind that these structures need to be designed around how AD will be supported to ease the administrative burden for the IT staff. In most cases, the end-users never see these logical structures; instead, they see the flat list that the global catalog provides through a directory query. But, if the logical AD structure doesn't match your organizationally or geographically structured IT groups, you're going to have problems with administrators having too much (or not enough) access to the objects they're responsible for administrating. Similarly, if you have a single centralized IT group and you've created overly complicated logical structures, the support of the environment will be more difficult.

So why am I concerned with the difficultly of administration? One reason: cost. The ongoing support of the IT infrastructure is the most costly element of providing that environment. The more difficult an environment is to support, the more support time it takes. Therefore, it becomes more costly to support. By designing the logical structures of Active Directory for the administrators, you save long-term support costs.

What's Your Forest Scope?

For most companies, a single AD DS forest is sufficient. But for some companies, multiple forests might be required. Typically, a company needs multiple forests when it has different groups that need to be separated for business and maybe even legal reasons. If you're considering multiple forests, take a careful look at any planned applications that will be using Active Directory and the impact that having multiple forests will have on that application. A great example is Microsoft Exchange. The default Global Address Book in Exchange is a one-to-one match with the AD forest that you deploy Exchange within. Therefore, if you have multiple forests, you must decide how that will affect your Exchange design. If you have good reasons for deploying multiple forests, there are ways to get around these difficulties (including directory synchronization between the forests), but these solutions add complexity to the support of the environment. So, bottom line, make sure you fully understand the impact of deploying multiple forests before doing so.

Often a Single Domain Is Enough!

A general design principle for any IT system is to keep things as simple as possible. This is very much the case when it comes to creating your domain structure in AD DS. Always start with a single domain and only add additional ones when there's sufficient justification.

REMEMBER

A single domain can easily hold over a million objects, making it sufficient for most companies.

Active Directory Is Built on DNS

Active Directory Domain Services and all the elements in the AD umbrella are dependent on DNS. Bottom line: Active Directory simply can't function without DNS! You certainly aren't required to use the Microsoft DNS, but here are some advantages to doing so:

- Microsoft DNS supports SRV records, which Active Directory requires.
- Microsoft DNS supports Dynamic DNS (DDNS), which Active Directory doesn't require — but that I *strongly recommend* for it.
- Although you can use some other DNS service that supports SRV records and DDNS, Microsoft DNS can store DNS data in Active Directory. By storing DNS data in the directory, you remove DNS replication from the network because Active Directory replication includes the DNS data. Other DNS services can't store DNS data in the directory.

Your Logical Active Directory Structure Isn't Based on Your Network Topology

Active Directory is based on the logical hierarchy of your organization. It organizes and manages resources in a user-friendly way. Network topology has no bearing on how you structure domains, trees, forests, and OUs. These items are all based on the logical business processes of the organization and not on the location of resources on the network. (For additional information on the logical AD structure, see Chapter 5.)

Limit Active Directory Schema Modifications

Modify the AD schema only when absolutely necessary. If you can find an existing attribute or class that can satisfy your schema needs, use it instead of creating a new schema object. Of course, there will be times when you must update the schema — especially to support an application that utilizes Active Directory as an information store. In those cases, make sure that you're carefully following the best practices for updating the schema, including making the changes directly on the schema master operations master DC and verifying that AD replication is working normally.

Understand Your Identity Management Needs

When you're developing your AD design, it's easy to focus on just the needs of the internal users in your company; however, don't forget about the needs of external users as well. The new AD server roles (Active Directory Federation Services, Active Directory Rights Management Services, Active Directory Certificate Services and Active Directory Lightweight Directory Services) can be used with your AD DS design to create an overall solution for the management of identities and access in multiple environments and to control what data access these identities have. In some cases, this might require the deployment of multiple instances of AD DS or AD LDS and can have an impact on your network design and firewall design and placement.

Place Domain Controllers and Global Catalogs Near Users

When deciding where to place domain controllers and global catalog servers, be sure that you place them near the users that need their services. Each Active Directory site should have at least one domain controller and one global catalog server so that users can reach these services within their own site. If a location doesn't have enough users to justify a local DC and GC server, make sure that the location's Active Directory site has a DC for the location to use and that the WAN between the location and the DC is reliable.

For those branch offices that require a DC but don't have adequate facilities to secure it, consider using a read-only domain controller to provide those services in a secure manner. Also, remember that you can enable local credential caching on that RODC for the users that are located in that branch office to help provide local authentication in case of network failure between the branch and the rest of the company.

Keep Improving Your Design

It's extremely unlikely that your first AD DS design attempt will be optimal. AD design is an *iterative* process, meaning that you most likely need to create the design on paper several times before you come up with the best solution. So, create your first design and then weigh it against the requirements to see how well you've met them. Also, make sure that you get feedback and opinions from other parties in your AD design team because they likely will come up with observations you haven't considered.

Chapter 17

Ten Cool Web Sites for Active Directory Info

In This Chapter

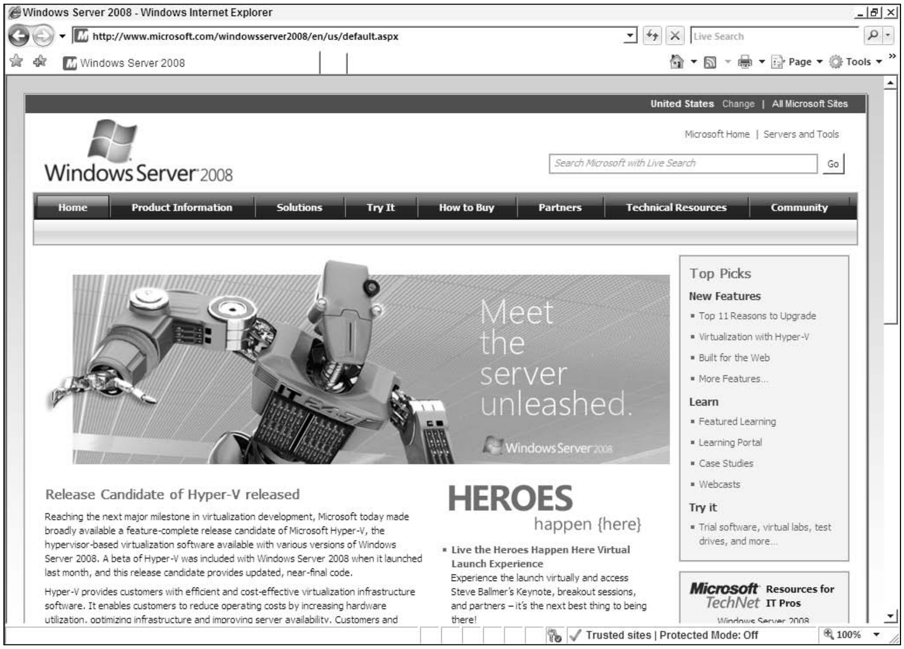
- Microsoft's Windows Server 2008 Web site
- Windows Server 2008 TechCenter
- *TechNet* magazine
- Directory Services Team blog
- Exchange Server Team blog
- *Windows IT Pro* magazine
- Windows Server Team blog
- Windows Server 2008 Most Recent Knowledge Base Articles feed
- Windows Server 2008 Most Popular Downloads
- My blog

Because Active Directory is a popular topic in the Microsoft IT arena, you can imagine that there's quite a bit of information about the subject on the Internet. And there is. In this chapter, I quickly cover some of my favorite sites on Active Directory, including a few blogs that cover AD. Some are Microsoft Web sites and some are not.

Microsoft's Windows Server 2008 Web Site

The most natural place to start your searching is Microsoft's Web site on Active Directory. Here, you can get information (including pricing) about the various versions of Windows Server 2008. Also, access to virtual labs and other downloads are available. (See Figure 17-1.)

```
www.microsoft.com/windowsserver2008
```



Windows Server 2008 - Windows Internet Explorer
http://www.microsoft.com/windowsserver2008/en/us/default.aspx
Live Search
Windows Server 2008
Page
Tools
United States Change | All Microsoft Sites
Microsoft Home | Servers and Tools
Search Microsoft with Live Search
Go
Windows Server 2008
Home
Product Information
Solutions
Try It
How to Buy
Partners
Technical Resources
Community
Meet the server unleashed.
Windows Server 2008
Top Picks
New Features
Top 11 Reasons to Upgrade
Virtualization with Hyper-V
Built for the Web
More Features...
Learn
Featured Learning
Learning Portal
Case Studies
Webcasts
Try it
Trial software, virtual labs, test drives, and more...
Microsoft TechNet Resources for IT Pros
Trusted sites | Protected Mode: Off
100%

Release Candidate of Hyper-V released

Reaching the next major milestone in virtualization development, Microsoft today made broadly available a feature-complete release candidate of Microsoft Hyper-V, the hypervisor-based virtualization software available with various versions of Windows Server 2008. A beta of Hyper-V was included with Windows Server 2008 when it launched last month, and this release candidate provides updated, near-final code.

Hyper-V provides customers with efficient and cost-effective virtualization infrastructure software. It enables customers to reduce operating costs by increasing hardware utilization, optimizing infrastructure and improving server availability. Customers and

HEROES happen {here}

- Live the Heroes Happen Here Virtual Launch Experience
  Experience the launch virtually and access Steve Ballmer's Keynote, breakout sessions, and partners – it's the next best thing to being there!

Figure 17-1: The Windows Server 2008 Web site.

Windows Server 2008 TechCenter

Although you can get to this Web site from the Windows Server 2008 site, this site is so important I list it separately. From this site, you can literally find information to keep you busy for days. One of the cool things included here that I found useful while writing this book is a Web version of the Help files installed on a Windows Server 2008 server.

```
http://technet.microsoft.com/windowsserver/2008
```

TechNet Magazine

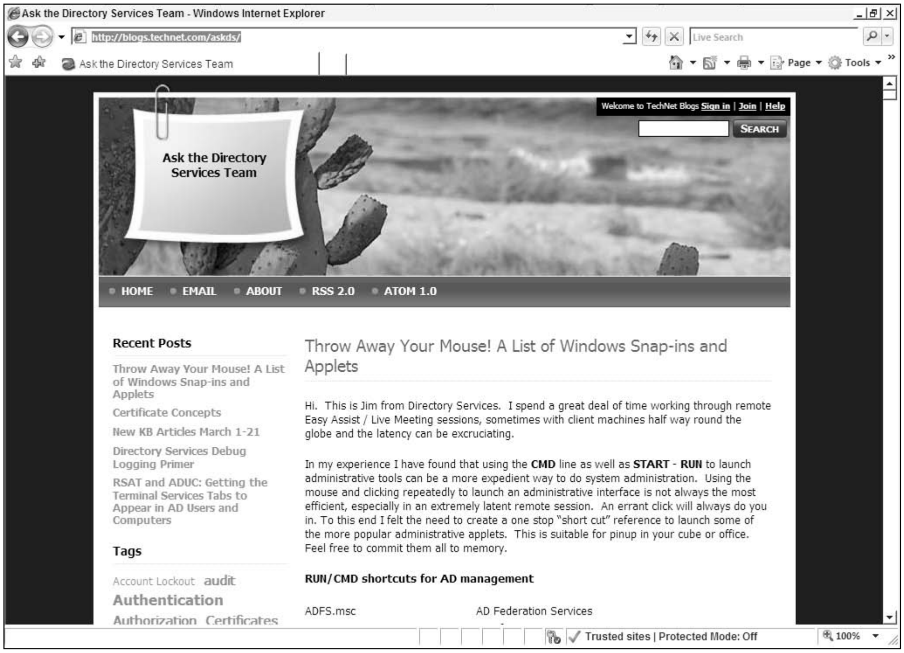
If you're an IT professional working with Microsoft products, I strongly urge you to consider subscribing to Microsoft's *TechNet* magazine. The subscription is free, so what do you have to lose! In addition to a print version, Microsoft also has the magazine available in an online format. Although not every issue is dedicated to Active Directory, more often than not you can find articles related to products that use Active Directory. Check it out.

```
http://technet.microsoft.com/magazine
```

Directory Services Team Blog

Blogs have become a popular way of disseminating information on the Web. The development team at Microsoft responsible for Active Directory has a blog. Where could you get better information on Active Directory than from the people that built it and that support it? (See Figure 17-2.)

```
http://blogs.technet.com/askds
```


Ask the Directory Services Team - Windows Internet Explorer
http://blogs.technet.com/askds/
Live Search
Ask the Directory Services Team
Page
Tools
Welcome to TechNet Blogs Sign in | Join | Help
Search
Ask the Directory Services Team
HOME
EMAIL
ABOUT
RSS 2.0
ATOM 1.0
Trusted sites | Protected Mode: Off
100%

Recent Posts

Throw Away Your Mouse! A List of Windows Snap-ins and Applets

Certificate Concepts

New KB Articles March 1-21

Directory Services Debug Logging Primer

RSAT and ADUC: Getting the Terminal Services Tabs to Appear in AD Users and Computers

Tags

Account Lockout audit Authentication Authorization Certificates

Throw Away Your Mouse! A List of Windows Snap-ins and Applets

Hi. This is Jim from Directory Services. I spend a great deal of time working through remote Easy Assist / Live Meeting sessions, sometimes with client machines half way round the globe and the latency can be excruciating.

In my experience I have found that using the **CMD** line as well as **START - RUN** to launch administrative tools can be a more expedient way to do system administration. Using the mouse and clicking repeatedly to launch an administrative interface is not always the most efficient, especially in an extremely latent remote session. An errant click will always do you in. To this end I felt the need to create a one stop "short cut" reference to launch some of the more popular administrative applets. This is suitable for pinup in your cube or office. Feel free to commit them all to memory.

RUN/CMD shortcuts for AD management

ADFS.msc	AD Federation Services

Figure 17-2: The Directory Services Team blog.

Exchange Server Team Blog

Another useful blog on Active Directory is the one maintained by the Exchange Server development team. Because Exchange is so tightly dependent on AD, very often you find information here concerning Active Directory, particularly when it involves AD and Exchange working together.

```
http://msexchangeteam.com
```

Windows IT Pro Magazine

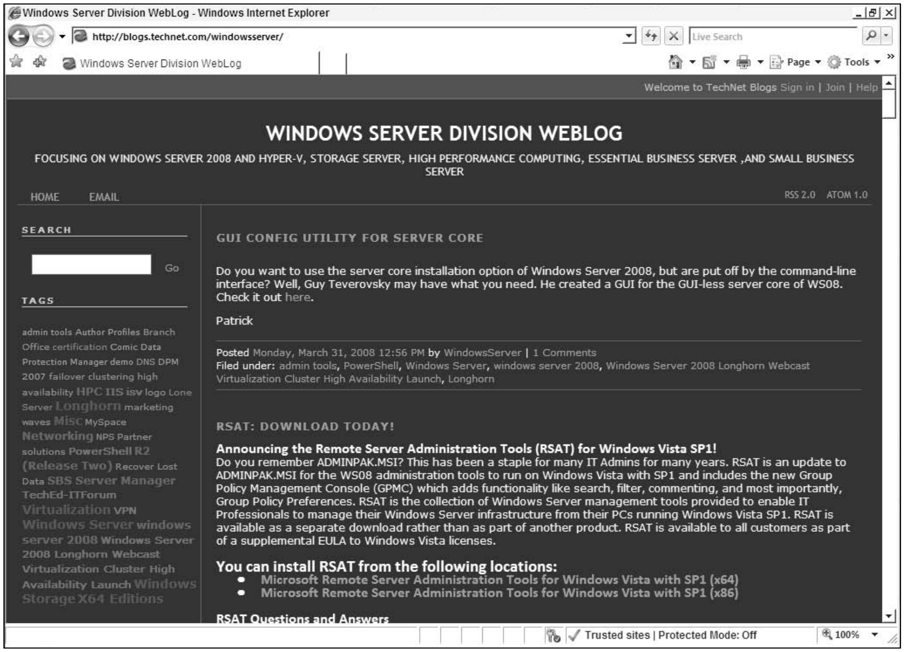
Although it has gone through several name changes over the years, *Windows IT Pro* magazine has long been a great source of information on Active Directory.

```
www.windowsitpro.com
```

Windows Server Team Blog

Of course, the Microsoft team responsible for the Windows Server product has a blog as well. (See Figure 17-3.)

```
http://blogs.technet.com/windowsserver
```


Windows Server Division WebLog - Windows Internet Explorer
http://blogs.technet.com/windowsserver/
Live Search
Windows Server Division WebLog
Page
Tools
Welcome to TechNet Blogs Sign in | Join | Help
HOME
EMAIL
RSS 2.0
ATOM 1.0
SEARCH
Go
TAGS
admin tools Author Profiles Branch Office certification Comic Data Protection Manager demo DNS DPM 2007 failover clustering high availability HPC IIS isv logo Lone Server Longhorn marketing waves Misc MySpace Networking NPS Partner solutions PowerShell R2 (Release Two) Recover Lost Data SBS Server Manager TechEd-ITForum Virtualization VPN Windows Server windows server 2008 Windows Server 2008 Longhorn Webcast Virtualization Cluster High Availability Launch Windows Storage X64 Editions
Trusted sites | Protected Mode: Off
100%

WINDOWS SERVER DIVISION WEBLOG

FOCUSING ON WINDOWS SERVER 2008 AND HYPER-V, STORAGE SERVER, HIGH PERFORMANCE COMPUTING, ESSENTIAL BUSINESS SERVER ,AND SMALL BUSINESS SERVER

GUI CONFIG UTILITY FOR SERVER CORE

Do you want to use the server core installation option of Windows Server 2008, but are put off by the command-line interface? Well, Guy Teverovsky may have what you need. He created a GUI for the GUI-less server core of WS08. Check it out here.

Patrick

Posted Monday, March 31, 2008 12:56 PM by WindowsServer | 1 Comments
Filed under: admin tools, PowerShell, Windows Server, windows server 2008, Windows Server 2008 Longhorn Webcast Virtualization Cluster High Availability Launch, Longhorn

RSAT: DOWNLOAD TODAY!

Announcing the Remote Server Administration Tools (RSAT) for Windows Vista SP1!
Do you remember ADMINPAK.MSI? This has been a staple for many IT Admins for many years. RSAT is an update to ADMINPAK.MSI for the WS08 administration tools to run on Windows Vista with SP1 and includes the new Group Policy Management Console (GPMC) which adds functionality like search, filter, commenting, and most importantly, Group Policy Preferences. RSAT is the collection of Windows Server management tools provided to enable IT Professionals to manage their Windows Server infrastructure from their PCs running Windows Vista SP1. RSAT is available as a separate download rather than as part of another product. RSAT is available to all customers as part of a supplemental EULA to Windows Vista licenses.

You can install RSAT from the following locations:

- Microsoft Remote Server Administration Tools for Windows Vista with SP1 (x64)
- Microsoft Remote Server Administration Tools for Windows Vista with SP1 (x86)

RSAT Questions and Answers

Figure 17-3: The Windows Server Team blog.

Windows Server 2008 Most Recent Knowledge Base Articles Feed

The next two Web sites really aren't Web sites but rather Really Simple Syndication (RSS) feeds. These are data feeds that can be consumed by IE 7.0 or any RSS feed aggregator, such as NewsGator. Microsoft has created an RSS feed that provides you a list of the most recent Knowledge Base articles on Window Server 2008. Therefore, with this feed, you're always up-to-date on the latest fixes and tweaks for Active Directory on Windows Server 2008. To get to this RSS feed, go to the following Web site:

```
http://support.microsoft.com/selectindex/?target=rss
```

Scroll to the Windows Server 2008 link. Click the link. On the new Web page, you find a link for subscribing to the feed.

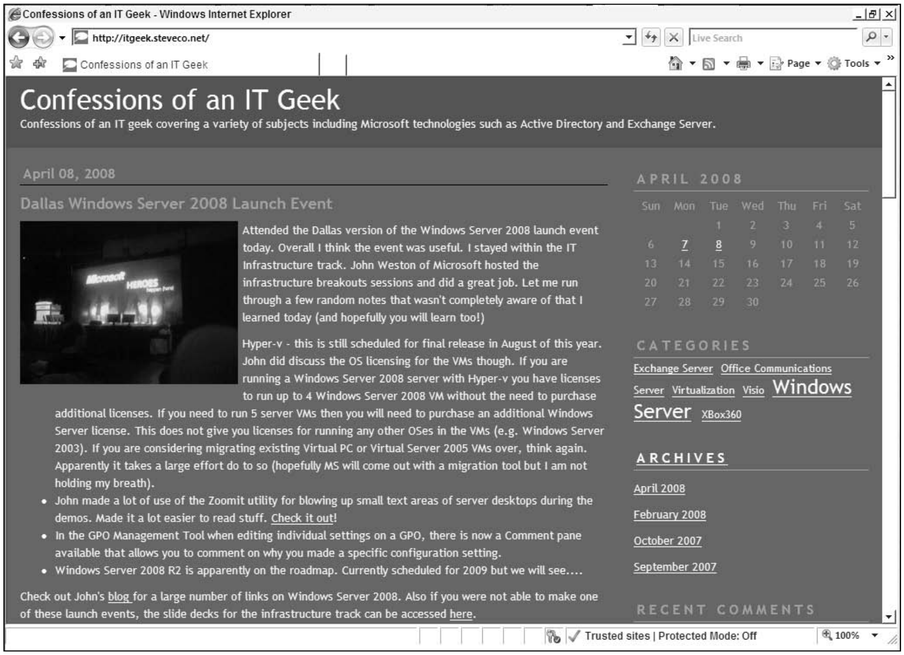
Windows Server 2008 Most Popular Downloads

The second RSS feed I want to provide you with is the one for the most popular downloads for Windows Server 2008. The idea being, if it's a popular download, there must be a good reason. This feed unfortunately is a bit harder to find. The best suggestion I have for getting this feed is to go to the earlier Windows Server 2008 TechCenter link and select Active Directory Domain Services from the Servers Role list. When you get to the AD DS page, you should see an RSS feed icon for the Windows Server 2008 Most Popular Downloads feed on the right.

My Blog

Last (but I hope not least), I provide you the link for my blog, Confessions of an IT Geek. (See Figure 17-4.) My intention is to cover topics related to Active Directory, Exchange, and other Microsoft products and technologies. Additionally, I consolidate a lot of current information from the above Web sites where possible. Therefore, I hope the site can cut down on you having to hop around to too many sites!

```
http://itgeek.steveco.net
```


Confessions of an IT Geek - Windows Internet Explorer
http://itgeek.steveco.net/
Live Search
Confessions of an IT Geek
Page
Tools
Trusted sites | Protected Mode: Off
100%

Confessions of an IT Geek

Confessions of an IT geek covering a variety of subjects including Microsoft technologies such as Active Directory and Exchange Server.

April 08, 2008

Dallas Windows Server 2008 Launch Event

Attended the Dallas version of the Windows Server 2008 launch event today. Overall I think the event was useful. I stayed within the IT Infrastructure track. John Weston of Microsoft hosted the infrastructure breakouts sessions and did a great job. Let me run through a few random notes that wasn't completely aware of that I learned today (and hopefully you will learn too!)

- Hyper-v - this is still scheduled for final release in August of this year. John did discuss the OS licensing for the VMs though. If you are running a Windows Server 2008 server with Hyper-v you have licenses to run up to 4 Windows Server 2008 VM without the need to purchase additional licenses. If you need to run 5 server VMs then you will need to purchase an additional Windows Server license. This does not give you licenses for running any other OSes in the VMs (e.g. Windows Server 2003). If you are considering migrating existing Virtual PC or Virtual Server 2005 VMs over, think again. Apparently it takes a large effort do to so (hopefully MS will come out with a migration tool but I am not holding my breath).
- John made a lot of use of the Zoomit utility for blowing up small text areas of server desktops during the demos. Made it a lot easier to read stuff. Check it out!
- In the GPO Management Tool when editing individual settings on a GPO, there is now a Comment pane available that allows you to comment on why you made a specific configuration setting.
- Windows Server 2008 R2 is apparently on the roadmap. Currently scheduled for 2009 but we will see....

Check out John's blog for a large number of links on Windows Server 2008. Also if you were not able to make one of these launch events, the slide decks for the infrastructure track can be accessed here.

APRIL 2008

Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

CATEGORIES

Exchange Server Office Communications Server Virtualization Visio Windows Server XBox360

ARCHIVES

April 2008

February 2008

October 2007

September 2007

RECENT COMMENTS

Figure 17-4: Confessions of an IT Geek blog.

Also if you want to reach me directly, you can e-mail me at

```
addummies@steveco.net
```

Chapter 18

Ten Troubleshooting Tips for Active Directory

In This Chapter

- Domain controller promotion issues
- Network issues
- Synchronizing time
- Can't log on to a domain
- Monitoring AD resources
- Can't modify the schema
- Replication issues
- Working with certificates
- Group policy issues
- Branch office users logging on for the first time

I would love to tell you that all my Active Directory efforts have been flawless. Alas, that's not true. I spent hours resolving some issues and needed help on others. So in the true spirit of cooperation with my fellow administrators — and because my editor tells me I should — I would like to share some troubleshooting tips for Active Directory.

Domain Controller Promotion Issues

Sometimes, when you're running the DCPROMO tool to promote a domain controller into an existing domain, the wizard fails, giving an error message that tells you that the domain isn't a valid Active Directory domain. More often than not, this error is a DNS problem, particularly when you're attempting to add a new domain controller to an existing domain or forest. Go back and make sure that your computer is configured to use an existing DNS server in the AD forest. Also, you can install DNS separately ahead of time and then run DCPROMO afterward.

Network Issues

Of course, Active Directory is a type of networked application. Therefore, Active Directory is only as reliable as the network that it's built on. If you're troubleshooting AD, verifying the operational functionality of the network between the domain controllers is a basic obligation. Start by verifying that the network between the domain controllers within the same AD site is working. PINGs by computer name are a good place to start, but keep in mind that firewalls sometimes are configured to prevent ICMP packets from being transmitted. Another good way to verify that the network is working is by using the `TELNET` command to connect to either the LDAP port (389) or the global catalog port (3269). If you get a blank screen immediately after connecting on either of these ports, then your network is most likely working well. If you aren't connected, then get on the phone with your network team and start working on the problem.

What Time Is It?

I cannot tell you how many times I've had to troubleshoot Active Directory issues that eventually were determined to be caused by a lack of time synchronization. Remember that Kerberos uses timestamps on the tokens used to access data and to log in with so that the data can't be captured and then replayed in the future as a way of hacking the infrastructure. If your domain controllers and member computers, including the user desktops and laptops aren't set to the same time, you are going to experience a wide range of odd problems, from not being able to log in to error messages when attempting to access files and applications. Also, keep in mind that time zones fall into this troubleshooting arena as well. 10 a.m. central standard time isn't the same as 10 a.m. eastern standard time. So make sure that you're using the correct time zone on your computers and that the actual time is correct on all the computers that you're troubleshooting as a first step in working out logon or data access issues. Also, if your server is having consistent problems with keeping time, you might consider replacing the clock battery on the server's motherboard. You can also use the `w32tm /resync` command to force an immediate timesync with the DC holding the PDC emulator role in the forest root domain of your forest.

Can't Log On to a Domain

Beyond time sync issues, any number of things can prevent a user from logging on to a domain. If you know that the user account and password are valid and the account isn't so new that replication may not have finished, consider these questions:

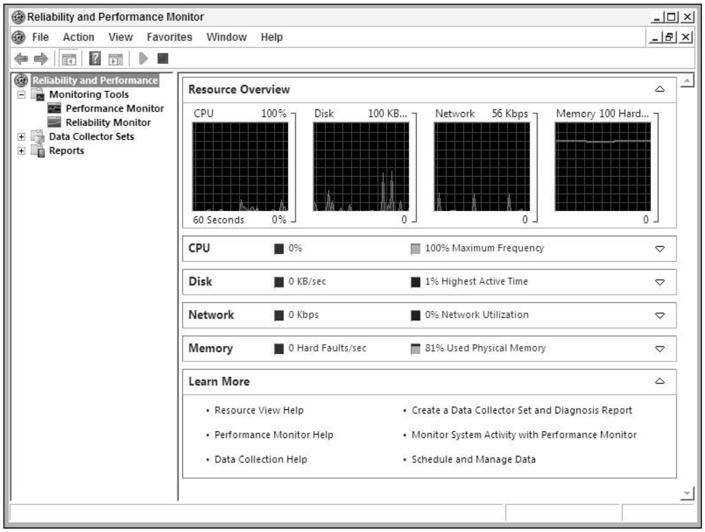
- Is TCP/IP configured correctly on the client computer?
- Can the user access DNS?
- Is a domain controller available?
- Is a global catalog server available?

Monitoring Active Directory Resources

In troubleshooting server and network problems, many administrators turn to performance analysis software to gather statistical data. The Reliability and Performance Monitor tool is provided with Windows Server 2008 as a window into how well your server is operating. The tool has three views into the server including the Resource Overview, Performance Monitor, and the Reliability Monitor. (See Figure 18-1.) If you think that you're having performance issues with your domain controller, this tool is a great place to start your troubleshooting.

TIP

Don't forget about the Event Viewer tool as well. If you're having actual errors on the server that are affecting performance, you'll most likely be seeing these errors logged in the Event Viewer.



CPU 0% 100% Maximum Frequency

Disk 0 KB/sec 1% Highest Active Time

Network 0 Kbps 0% Network Utilization

Memory 0 Hard Faults/sec 81% Used Physical Memory

Figure 18-1: The Resource Overview view in the Reliability and Performance Monitor.

Can't Modify the Schema

Make sure that you're a member of the Schema Administrators group. If you're not, open the Schema Administrators group in AD Users and Computers and add your ID to the group. If you still can't modify the Active Directory schema, the schema master server is probably unavailable. You need to test the network connections to the schema master.

Replication Issues

If you're seeing that changes on one domain controller aren't being replicated to another domain controller within a reasonable time, you're probably going to need to do some troubleshooting with AD replication. After you establish the basics (good network connectivity, time synchronization is good, DNS access), start using the REPADMIN tool to get information about the replication health. The best place to run this is on the DC that isn't getting the updates. Start by running the REPADMIN `/REPLSUMMARY` command; this provides a good overview of the replication health of the domain controller.

Working with Certificates

If you're implementing some of the AD components that make heavy use of certificates, such as AD Federation Services or AD Rights Management Services you might come across problems with using certificates. Most of the time, the issues are related to whether the certificate is trusted by the computers that use the certificates. If you're using an Enterprise CA via AD CS, you shouldn't come across this problem within your AD but if you have multiple environments using the certificates (which is typically the case with an AD FS deployment), make sure that all computers trust the certificates that are being used. One easy way of doing this is to modify the Default Domain Policy for each domain including the Enterprise CA certificate into the Trusted Root Certification Authorities policy.

TIP

As I mention in Chapter 10, the Enterprise PKI (PKIView) tool is an excellent troubleshooting tool when working with certificates.

Group Policy Issues

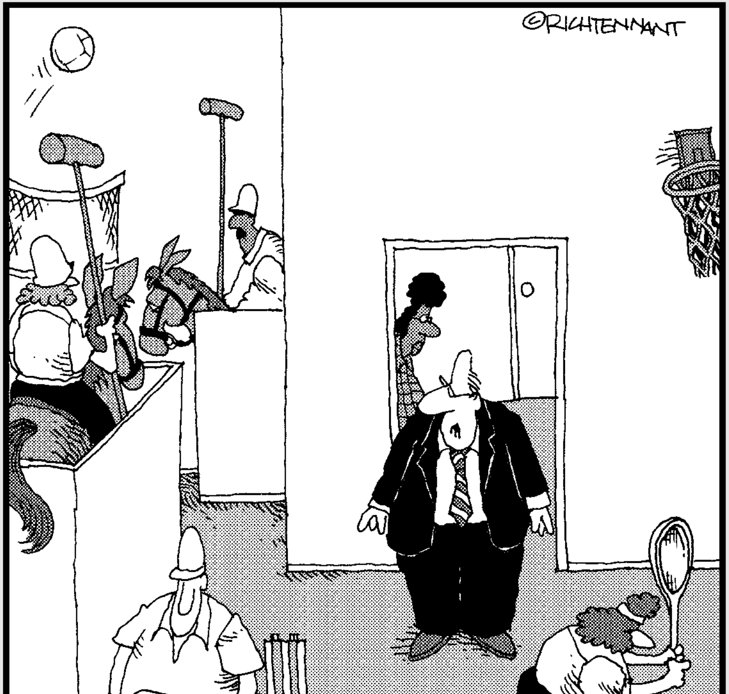
If you're having problems with users or computers not receiving the expected policy settings that you're attempting to enforce with Group Policy Objects, remember that you have the Resultant Set of Policy (RSoP) tool available to you to determine exactly what policy settings the user or computer is receiving from Active Directory. One other area of troubleshooting to look at is how well the DFSR service is working because policy files are replicated through this service. You can use both the DFSCMD and DFSRADMIN tools to help with troubleshooting in this area.

Branch Office Users Logging In for the First Time

If you're deploying read-only domain controllers in a branch office scenario, remember that RODCs don't normally store credential information locally. This requires that the network between the branch office and the location that includes the closest writable domain controller (as determined by the AD site topology) be functional any time a branch office user logs in. So, if branch office users are having difficulty logging in, check on the network connectivity. Keep in mind, though, that you have the option of enabling credential caching for the RODC, which helps with future logins. But this doesn't help with a first time user logging in. For that situation, the network must be functional before the credentials can be cached on the RODC.

Part VI
Appendixes

The 5th Wave By Rich Tennant



"Ms. Gretsky, tell the employees they can have internet games on their computers again."

In this part . . .

We've included the following appendixes for you to use as a reference:

- ✓ Appendix A is a Windows Server 2008 command line reference for some of the more popular tools for managing Active Directory.
- ✓ Appendix B is a glossary of terms to help you get familiar with the lingo.

Appendix A

Windows 2008 AD Command Line Tools

Microsoft has made available a wealth of command line tools in support of Active Directory. Although the GUI tools are easy to use and are great for a small number of operations, for real power and for the ability to run multiple commands in a script, you can't beat knowing how to use the various command line tools. In this appendix, I cover the more important commands you should be familiar with. Unfortunately, so many commands are useful that I can't cover all of them, but I provide a table listing of the other command line tools at the end of this appendix so that you're aware of their existence. So let's get going!

The assumption with all these commands is that you're running them within a command prompt window running with elevated administrative permissions. This is done by either logging into the domain controller with an administrative ID or using the Run As option to run the command prompt with an administrative ID context (done by right-clicking the command prompt icon in the start menu and selecting Run As).

DNSCMD

Although the DNS console is typically where you administrate the DNS service on your domain controllers, the DNSCMD tool can be used to perform a number of DNS operations.

The syntax of the command is

```
DNSCMD <servername> [command] [command arguments]
```

The available commands within DNSCMD are:

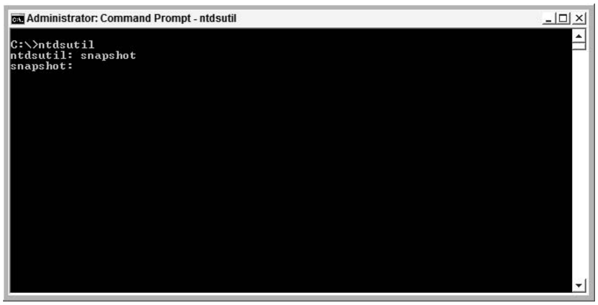
- **/clearcache:** Clears out the cached entries on the DNS server.
- **/config:** Allows for the configuration of the DNS server settings and the configuration of zones on the DNS server.

- **/enumzones:** Lists the zones that are stored on the DNS server.
- **/info:** Retrieves the DNS server information.
- **/ipvalidate:** Validates DNS server and DNS zone file availability by IP address.
- **/resetlistenaddress:** Defines which IP address on the server should listen for DNS requests.
- **/resetforwarders:** Allows for DNS forwarders to be either configured or clear.
- **/startscavenging:** Starts the scavenging of stale resource records.
- **/statistics:** Prints the statistics of the DNS server and allows for those statistics to be cleared.
- **/writebackfiles:** Writes back any updates to the zone in the DNS cache to the zone file(s).
- **/zoneinfo:** Views zone information.
- **/zoneadd:** Creates a new zone.
- **/zonedelete:** Deletes a zone.
- **/zonepause:** Temporarily prevents DNS resolution of a zone.
- **/zoneprint:** Displays the contents of a zone.
- **/zoneresume:** Resumes DNS resolution of a paused zone.
- **/zonereload:** Reloads into the DNS server application the contents of a zone file.
- **/zonerefresh:** Updates a secondary zone copy by pulling from the primary zone holder DNS server.
- **/zoneresettype:** Changes the zone type between an AD integrated, primary, secondary, stub, or forwarder zone.
- **/zoneresetsecondaries:** Configures secondary zone settings.
- **/zoneupdatefromds:** Updates an AD integrated zone from Active Directory.

NTDSUTIL

The NTDSUTIL tool is a very powerful and versatile utility that can be used with both AD DS and AD LDS directory instances to perform advanced administrative operations. The tool is initiated by running the `NTDSUTIL` command on any server where the AD DS or AD LDS role has been installed.

After you run the NTDSUTIL command, you're presented with the NTDSUTIL prompt. From this prompt you can enter into the various command areas of the tool. Each of these areas has its own unique prompt. The QUIT command can be used to exit the various areas of the tool as well as the tool itself. (See Figure A-1.)



```
C:\>ntdsutil
ntdsutil: snapshot
snapshot:
```

Figure A-1: Running the NTDSUTIL and then entering the SNAPSHOT command area of the tool.

The following sections cover the more commonly used command areas of NTDSUTIL.

NTDSUTIL has a built-in help facility that can be accessed simply by typing in the ? command at any level within the tool.

NTDSUTIL Activate Instance

Because NTDSUTIL can be used against multiple directory instances that may be running on the same server, including both an AD DS directory and possibly one or more AD LDS instances, you must specify which instance you want to work with before doing anything else within the NTDSUTIL tool. The Activate Instance command syntax is

```
ACTIVATE INSTANCE [NTDS | <AD LDS INSTANCE>]
```

If you're going to work with the AD DS instance on a domain controller, then the instance name will always be NTDS. To work with an AD LDS instance, you must specify the instance name that was defined at the time the instance was created.

With many of the command areas listed below you need to specify which instance you are working with before you can get into that area. You specify the instance you need to work with by using the Activate Instance command.

Also, keep in mind that most of the commands here have shortcut versions. The format of the shortcuts is typically just a minimal number of characters from the full command. For example, the `Activate Instance` command shortcut to select the NTDS instance is

```
AC I NTDS
```

NTDSUTIL Authoritative Restore

I discuss the use of this command in Chapter 15, but to reiterate, the NTDSUTIL `Authoritative Restore` command is used to mark one or more objects as authoritative after performing a restore of the database. This command is typically used to mark previously deleted objects to be replicated back to all the other domain controllers, which results in the objects being restored to the directory instance on all domain controllers. ***Note:*** Before you can use this command, the directory instance that you're working with must already be offline. This can be done either by switching to Directory Services Restore Mode, if working with AD DS, or by stopping the service for the AD LDS instance with which you want to work. The other alternative, if working with Windows Server 2008 AD DS, is to stop the AD DS service. After starting the NTDSUTIL tool, specify the instance you need to work with by using the `Activate Instance` command as I describe above. You can then enter the `Authoritative Restore` command area by entering in the following command at the NTDSUTIL prompt:

```
AUTHORITATIVE RESTORE
```

To define a single object to be marked authoritative you can use the following command:

```
RESTORE OBJECT [distinguished name of object]
```

If you need to mark a container, such as an organizational unit and all of its contents as authoritative, you can use the following command to mark those objects:

```
RESTORE SUBTREE [distinguished name of container]
```

NTDSUTIL Files

The `Files` command area of NTDSUTIL allows the administrator to perform management functions on the database and transaction log files on AD DS and AD LDS instances. As with the `Authoritative Restore` command, you need to activate the directory instance you want to work with and stop the service for that instance. A good number of commands in this area can be dangerous to use; in many cases, you will not use these commands unless you're on the phone with Microsoft resolving an issue. But, I do want to cover a few commands here.

First, to enter into the `Files` command area, type the command

```
FILES
```

The `Compact` command can be used to perform an offline defragmentation of the database file. Typically, this is done only to shrink the size of the file following a large number of object deletions from the instance. Here is the syntax of the command:

```
COMPACT to [target directory]
```

target directory is the file directory on the server on which you want to create the newly compressed database file. Note that after running this command, you must copy the file over the previous database file before it can be used.

The `Info` command provides information about the location and size of the database and log files. It's invoked simply by running

```
INFO
```

The `Move` command allows for the relocation of the database or log files to a new file directory on the server. The commands to move the database files and log files are

```
MOVE DB TO [target directory]
```

```
MOVE LOGS TO [target directory]
```

Note that this command not only moves the files but also updates the registry so that the directory service (AD DS or AD LDS) knows where the files were moved.

NTDSUTIL IFM

The `IFM` or `Install From Media` command is used when you're installing a new domain controller so that the new DC is immediately populated with data from another DC. This command is beneficial when you don't want to wait for normal AD replication to populate the new domain controller. The media that you're installing with is created by using this command. To get into the `IFM` command area, simply type in the command

```
IFM
```

The following are the various ways that you can use the `IFM` command. The first way is to create a full installation media:

```
CREATE FULL [target directory]
```

Note that this media can be used on a normal AD DS domain controller. You can also use this command to create a media copy for an AD LDS server if you've selected an AD LDS instance by using the `Activate Instance` command. The next command can be used to create install media for a read-only domain controller:

```
CREATE RODC [target directory]
```

`IFM` can also be used to create an install media that includes the contents of the SYSVOL by adding that word to the command:

```
CREATE SYSVOL FULL [target directory]
```

```
CREATE SYSVOL RODC [target directory]
```

NTDSUTIL Local Roles

When you're installing a read-only domain controller at a branch office site, you might need to depend on administrators at that local site to perform the installation and ongoing administration of that RODC. This possibility requires the branch office administrators to have administrative rights on that RODC. The `Local Roles` command is used to delegate administrative permissions to an individual RODC. This command actually modifies registry entries listing what accounts should be added to the members of the Built-In Security group (roles) like the Administrators group. The roles you can add members to with this command include the Administrators, Backup Operators, and Server Operators groups. You get into this command area from the NTDSUTIL prompt by entering the following:

```
LOCAL ROLES
```

The first command in this group provides a list of the defined roles that are on the RODC you're running the command on.

```
LIST ROLES
```

To add an account to one of these roles, use the following command:

```
ADD [account] [role]
```

To remove an account from a role:

```
REMOVE [account] [role]
```

To view the accounts you've added to the roles on this RODC:

```
SHOW ROLES
```

NTDSUTIL Roles

The `Roles` command is used to work with operations masters in AD DS. The two procedures you can perform with this tool are the transfer of an operations master role from one DC to another and, when an operations master DC is unavailable, assigning the role (known as *seizing*) to another DC. This command should be executed on the DC that you want to hold the role after the command is completed. To enter the `Roles` command area, at the `NTDSUTIL` prompt, type

```
ROLES
```

To transfer a role (you select from one of the five roles listed), the command is

```
TRANSFER <SCHEMA MASTER | NAMING MASTER | INFRASTRUCTURE
          MASTER | PDC | RID MASTER>
```

To seize a role (you select from one of the five roles listed), the command is

```
SEIZE <SCHEMA MASTER | NAMING MASTER | INFRASTRUCTURE
          MASTER | PDC | RID MASTER>
```

NTDSUTIL Set DSRM Password

I hope you won't need to use directory service restore mode frequently; however, there's a chance that when you do need to enter DSRM mode you might not remember the DSRM password for the DC. Never fear. With the `Set DSRM Password` command area, you can reset the password. To enter this area, type the following command:

```
SET DSRM PASSWORD
```

To reset the DSRM on the server type after entering this command area, type

```
RESET PASSWORD ON SERVER [server name]
```

NTDSUTIL Snapshot

The `Snapshot` command area of NTDSUTIL is used to create and manage snapshot backups of either AD DS or AD LDS directory instances on a server. This area is entered by typing

```
SNAPSHOT
```

The first step is to create a snapshot, but before you can do this, you need to specify the instance you are creating the snapshot of by using the `Activate Instance` command. After this is finished, you can create a snapshot of the active instance by typing

```
CREATE
```

When this snapshot is created, it's assigned both an index number and a long GUID number. You can use either number to refer to the snapshot with the other `Snapshot` commands. To view a list of the available snapshots (showing the index and GUID numbers), type

```
LIST ALL
```

To mount one of these instances so that you can use the DSAMAIN tool to access its contents (see Chapter 15), type the following command:

```
MOUNT [snapshot index or GUID]
```

To unmount a snapshot type

```
UNMOUNT [snapshot index or GUID]
```

REPADMIN

As I mention in Chapter 15, the REPADMIN tool is an indispensable utility when you're experiencing replication issues with an AD DS forest. With this tool, you can view the replication status of individual domain controllers and force various types of replications to occur.

The REPADMIN syntax is

```
REPADMIN [command][command arguments]
            [/u: {domain\user}][/pw:{password | *}]
```

The */u* and */pw* parameters can be used to specify an administrative ID that you want to run REPADMIN within. The available commands in REPADMIN include:

- **/REPLSUMMARY:** Provides a replication status report that you can use to identify any current replication failures.
- **/SHOWREPL:** Views replication status from the last replication cycle.
- **/KCC:** Initiates the Knowledge Consistency Checker (KCC) to perform a recalculation of the inbound replication topology.
- **/QUEUE:** Views the queued inbound replication requests that are pending.
- **/PRP:** Defines a password replication policy for read-only domain controllers.
- **/REPLICATE:** Forces an immediate replication cycle of a specific directory partition with a specified domain controller.
- **/RODCPWDREPL:** Initiates a password replication cycle to one or more RODCs for a specified set of users.
- **/SYNCALL:** Initiates a replication of a domain controller with all defined replication partner domain controllers.

DSAMAIN

I cover the Mountable AD tool (DSAMAIN) in Chapter 15, but to reiterate, this tool is used to mount an AD DS or AD LDS database on a server so that its contents can be viewed. Typically, you do this to determine the database's contents before restoring it over a production directory. The DSAMAIN syntax is

```
DSAMAIN -DBPATH [database file path] -LDAPPORT [LDAP
            port number] {-LOGPATH [log file path]
            -ADLDS        -SSLPORT [SSL port number]
            -GCPORT [global catalog port number] -GCSSLPORT
            [global catalog SSL port number] -ALLOWUPGRADE
            -ALLOWNONADMINACCESS}
```

The details on each of the options are

- **-DBPATH:** This is to specify the location of the database (DIT) file that you want to mount. This can be either an AD DS or AD LDS database file. This file can be either a local file on the server or a DIT file located on a snapshot taken with the NTDSUTIL `SNAPSHOT` command. This is a required option.
- **-LDAPPORT:** This option specifies what TCP port number the directory should be accessible from using LDAP. You use this option to specify a different LDAP port in the case where you are mounting this database on a server that is already running a production directory on the default LDAP port (389). This is a required option.
- **-LOGPATH:** With this option, you can specify a folder where transaction log files for the database will be created.
- **-ADLDS:** If the database you are to mount is from AD LDS, you need to specify this option.
- **-SSLPORT:** Specifies an SSL port number to be used if you want to provide LDAP over SSL access to the mounted database.
- **-GCPORT:** Defines a global catalog port number for an AD DS mounted database.
- **-GCSSLPORT:** Defines an SSL encrypted global catalog port number.
- **-ALLOWUPGRADE:** Allows for a down-level DIT file (Windows 2000 or 2003) file to be upgraded so that it can be mounted by using DSAMAIN.
- **-ALLOWNONADMINACCESS:** Normally, a mounted database is available only to members of the Domain Admins or Enterprise Admins groups. Specifying this option allows users without administrative rights to access the database.

Other Commands

You could literally fill an entire book with all the available commands related to supporting AD. Unfortunately, I can't go into detail on every command here, but I want you to know of these command's existence. Table A-1 includes all the available commands you should be familiar with if you're going to deploy or support Active Directory.

Table A-1 Windows 2008 AD Command Line Tools

Tool Name	*Description*
ADPREP.EXE	The AD Preparation Wizard. This tool is used to update the schema and permissions when upgrading AD from a previous version to Windows Server 2008 AD DS. This tool is also used to configure a preexisting forest to support RODCs.
AUDITPOL.EXE	Enables user to modify the audit policy of local or remote computers.
CERTREQ.EXE	Provides a method for requesting and retrieving certifications from AD CS.
CERTUTIL.EXE	Command line tool for viewing and managing the configuration of AD CS.
CSVDE.EXE	An import and export utility for AD DS. Supports the CSV file format so that you can view and edit the file information in programs that support CSV files, such as Microsoft Excel.
DCDIAG.EXE	A troubleshooting tool for AD DS domain controllers.
DCPROMO.EXE	The Active Directory Domain Services Installation Wizard. This tool is used to promote and demote servers into AD DS domain controllers.
DSACL.EXE	Command line tool for ACL management.
DSADD.EXE	Supports the creation of new objects into a directory.
DSDBUTIL.EXE	A tool that is similar to NTDSUTIL but more directed toward AD LDS support.
DSGET.EXE	Retrieves specified properties of existing directory objects.
DSMGMT.EXE	Management tool for AD LDS.
DSMOD.EXE	Modifies an existing directory object.
DSQUERY.EXE	Displays directory content based on the supplied parameters.
DSRM.EXE	Deletes a directory object.
GPFIXUP.EXE	Used to update GPOs in a domain that has been renamed by using RENDOM.

(continued)

Table A-1 *(continued)*

Tool Name	*Description*
GPRESULT.EXE	Provides Resultant Set of Policy functionality at the command prompt level.
GPUPDATE.EXE	Supports updating a group policy setting against a user or computer.
KSETUP.EXE	Kerberos setup that configures a Windows client for MIT Kerberos V5 interoperability.
KTPASS.EXE	Configure a non-Windows machine to become a security principal in AD DS.
LDIFDE.EXE	Similar to the CSVDC utility except that it works with LDIF text files, as defined in RFC 2849.
LDP.EXE	An LDAP administration tool. Can be used with both AD DS and AD LDS.
NET.EXE	A tool that provides the ability to perform a large number of network operations including creating computer and user accounts, start and stop services, and mapping network drives and printers.
NETDOM.EXE	Manage computer accounts and trust relationships.
NLTEST.EXE	A legacy tool that has been available since Windows NT 4.0. Can view and verify trust relationships and force remote shutdowns of servers.
NSLOOKUP.EXE	A DNS administration and troubleshooting tool.
RENDOM.EXE	Used to rename an AD DS domain.
W32TM.EXE	Manage and troubleshoot the Windows Time Service.

Appendix B

Glossary

Reading along about Active Directory and then suddenly running into a term with which you're not familiar can be quite distracting and frustrating. It can really put a crimp in your learning curve.

Use this glossary as a quick guide to the new terms that you run across in this book. By using this handy reference, you don't have to go digging around in various chapters to find the definition you're seeking.

Words appearing in *italics* within a definition refer to terms that I define elsewhere in this glossary.

Access Control Entry (ACE): The individual items of an *Access Control List (ACL).* Access Control Entries specify which users can access the resource and what access permissions (read/write, read-only, and so on) the user may exercise.

Access Control List (ACL): An ACL details which users may access a network resource. (*See also* Access Control Entry.)

Active Directory Certificate Services: A Windows Server 2008 server role that supports a public key infrastructure design by providing *digital certificate* enrollment and verification services.

Active Directory Domain Services: A Windows Server 2008 server role that provides an extensible directory service as well as an authentication and authorization service for other Windows-based applications.

Active Directory Federation Services: A Windows Server 2008 server role that supports the creation of *federations* in the support of Web Single Sign-On solutions.

Active Directory Lightweight Directory Services: A Windows Server 2008 server role that allows a server to support one or more *LDAP*-accessible *directory* instances.

Active Directory Rights Management Services: A Windows Server 2008 server role that provides control of what authorized users can do with documents.

Active Directory Services Interface (ADSI): The programming interface that applications use to access the Active Directory.

American National Standards Institute (ANSI): A nonprofit organization that coordinates voluntary standardization efforts within the United States.

Application programming interface (API): The programming interface that applications use to send requests to the operating system.

Attribute: A parameter describing an *object*. Objects consist of required attributes and optional attributes.

Authentication: The process of validating a user's ID and password and granting access to network resources.

Bandwidth: The amount of data that you transmit over a communications channel in a particular amount of time, usually one second. Available bandwidth on a network refers to the amount of bandwidth available for use, excluding normal network traffic.

Bridgehead server: One server within a *site* that handles all intersite *directory replication*. This server compresses and replicates to other sites all *directory* updates. In turn, the bridgehead server receives directory updates from all other sites. After the bridgehead server receives the updates, it replicates those updates to the other *domain controllers* in the site.

Canonical name: A DNS resource record identifying a nickname or alias given to a network host.

Child domain: Subordinate *domains* that branch from the *root,* or *parent,* domain.

Claim: Declarations that are made about your identity by a recognized security authority.

Container: An Active Directory *object* that holds other objects and containers. *Sites, domains,* and *organizational units (OUs)* are all containers.

Cost: In the context of Active Directory sites, *cost* is a measurement that the system administrator assigns to each *site link.* The cost value helps Active Directory determine which site link is a primary route (assigned a low cost, such as 1) and which is a secondary route (assigned a higher cost, such as 100) between *sites.*

Cross-link trust: A relationship that you establish to shorten the path between *domains* in the same *forest.* These relationships are *transitive trusts* that you manually create.

Datastore: A database file.

Data table: Stores all the specific data in Active Directory about users, groups, printers, and so on.

Defragmenting: The process by which the database file rearranges the pages of the file into a more compact format. This process is very similar to what you go through when you defragment the hard drive of your PC.

Digital certificate: An electronic ID card that verifies a user's credentials so that the user can communicate with network resources.

Directory: A network accessible application that acts as a hierarchical information store of data.

Distinguished name (DN): An X.500-based naming convention that uses particular abbreviations to define the path leading to an Active Directory *object.* The path `DC=com/DC=corp/CN=Users/CN=User1`, for example, is a distinguished name.

Distributed application: Applications that run on computers spread throughout the network.

DNS table: A listing of *resource records* that match a host's *IP* address to its name. Together, these resource records make up the DNS table that the *Domain Name Service (DNS)* references.

Domain: Within Active Directory, a boundary encompassing Active Directory *objects* for security or administrative purposes.

Domain controller (DC): A server that authenticates users seeking access to the *domain*.

Domain Name Service (DNS): A network's name-resolution service. A network client uses DNS while searching for a host's *IP* address. DNS can determine the IP address from the host's name.

Dynamic DNS (DDNS): Dynamic DNS enables hosts to update the *DNS table* with host names and addresses as they're added to the network.

Explicit trust: A one-way relationship, which means that Tree A can access the resources of Tree B, but Tree B can't access the resources of Tree A.

Extensible: An item you can add to or expand. The Active Directory database, for example, is extensible because it can be expanded to include new network *objects.*

Extensible Storage Engine (ESE): The Microsoft database engine on which Active Directory is based.

External trust: A relationship that the system administrator creates for *domain* users to access resources in a domain outside a *tree* or *forest*, between other Windows 2008 domain trees and forests, or between a Windows 2008 domain and a Windows NT domain. External trusts are always one-way, *nontransitive trusts*.

Fault tolerance: Redundant components that you configure to prevent lost data or services in the event of a system crash or network outage.

Federation: A trusted relationship between two security authorities that allows for identities to be projected from an account environment into a resource environment; therefore, allowing for an identity in the account environment to access data in the resource environment.

Flexible Single Master Operations (FSMO): *See* Operations Masters.

Forest: A grouping of domain *trees* that you join by *transitive trust* relationships. The domain trees are separate *namespaces* rather than a contiguous namespace. `Corp.com` and `xyz.com`, for example, are both separate namespaces, but you can join them by transitive trusts to form a forest.

Fragmentation: Just as disk drives become fragmented as you add and remove files, databases become fragmented as you move data in and out. A fragmented database takes more disk space and isn't as efficient. The system administrator then uses a database utility to *defragment* the database.

Fully Qualified Domain Name (FQDN): The entire path leading to a network *object*. `User1.namerica.corp.com`, for example, is an FQDN.

Functional model: A method that you use for designing an Active Directory structure that you can adapt to a variety of organization charts. By using a functional model, you may group *domains* by department, division, or project.

Garbage collection: Active Directory's automated database cleanup that occurs, by default, every 12 hours and takes care of deleting old log files, removing *tombstones,* and *defragmenting* the database file.

Geographic model: A popular method that you use for designing an Active Directory structure for an organization with specific geographic boundaries, such as a company with international divisions. Administrative functions within one location function separately from those in other locations.

Global catalog (GC): A searchable index that enables users to search for network *objects* without knowing their *domain* locations. The global catalog is a partial replica of the Active Directory.

Group policy: User and computer settings that apply to a specific group of users within a *site,* a *domain,* or an *OU* on the network.

Group Policy Object (GPO): A Group Policy Object is a set of user and computer configuration settings that you store as an *object* in the Active Directory. You apply the policy settings to computers within a *site,* a *domain,* or an *OU* to impose settings on the user or the computer.

Hierarchical: A logical, top-down structure.

Inheritance: In a *tree* hierarchy, *parent domains* and OUs pass along properties, such as GPOs and permissions, to their *child domains*. This process is known as inheritance.

Installation: In the context of Active Directory, loading Windows 2000 Server onto a partition that doesn't currently hold an operating system. Sometimes known as a fresh build.

Instance: A unique directory service running on a server. Each instance on a server utilizes a unique network port for *LDAP*.

Internet Engineering Task Force (IETF): An Internet standards governing organization.

Internet Protocol (IP): A network protocol that you use to address and route data packages between networked hosts.

Intersite replication: *Directory* updates between *domain controllers* in different *sites*.

Intrasite replication: *Directory* updates between *domain controllers* in the same *site*.

Kerberos: The default security authentication protocol in Active Directory Domain Services.

Key Distribution Center (KDC): A *Kerberos* security service running on *domain controllers*. It gives out tickets and keys to control resource access on the network.

Leaf: An Active Directory *object* that, unlike a container object, can't contain other objects. Printers and users are examples of leaf objects.

Lightweight Directory Access Protocol (LDAP): An Internet standard that enables Web browsers to find and access information in a *directory* service database. LDAP is based on the X.500 Directory Access Protocol (DAP) but is more efficient and more widely used.

Logical structure: The conceptual framework for Active Directory in which you match the Active Directory configuration to the business processes of a corporation or organization.

Mixed mode: A network operating with a mixture of Active Directory domain controllers that are running varying versions of the Windows Server operating system.

Namespace: A *logically structured* naming convention in which all *objects* are contiguous, or connected in an unbroken sequence. All the names within a *namespace* share the same *root domain*.

Native mode: A network in which all *domain* servers are running the same version of the Windows Server operating system.

Nested containers: A *container* inside a container is a nested container.

Nontransitive Trust: A trust relationship between two *domains* that does not extend to other trusted domains.

Object: A user, group, printer, or any other Active Directory component with descriptive parameters or *attributes*.

Object class: A set of mandatory *attributes* and optional attributes that combine to define a particular class of Active Directory *objects*. A user is one object class and a printer is another. Object classes are sometimes known as *schema objects* or metadata.

Object identifiers: Dotted decimal numbers that the *American National Standards Institute (ANSI)* assigns to each *object class* and *attribute.* ANSI assigns a specific root identifier to a U.S. corporation, and the corporation then assigns variations of its root identifier to the *objects* and attributes that it creates.

Operations Masters: Five unique roles that are defined within both an Active Directory Domain Service (AD DS) forest and an AD DS domain. The first two roles (Schema Master and Domain Naming Master) are unique for the forest. The remaining roles (PDC Emulator, RID Master, and Infrastructure Master) are unique for each domain in the forest.

Organizational unit (OU): A logical *container* within a *domain* that you use to organize *objects* for easier administration and access. A domain contains OUs. They can't span multiple domains nor can they contain objects from other domains.

Parent-child trust: A transitive trust relationship between a parent domain and its child domain.

Parent domain: A *domain* with subordinate domains. The *root* of the *tree* is a parent domain. *See also* Root domain.

Partition: A separate portion of a directory database that has its own unique replication topology. In AD DS, four partitions are available: Schema, Configuration, Domain, and Application.

Physical structure: A framework for Active Directory that encompasses the network configuration, network devices, and network *bandwidth*.

Propagating updates: The process of forcing *replication* — the exchange of database information between the *domain controllers* within a *domain* — to occur.

Read-only Domain Controller: An AD DS domain controller that contains a read-only copy of the directory.

Relative distinguished name (RDN): The portion of an object's name that is distinct from the object's path. The RDN is an X.500-based convention that's a kind of subset to the *distinguished name (DN)* convention. A relative distinguished name is sometimes known simply as a relative name. `User1.namerica.corp.com`, for example, is a distinguished name. The relative distinguished name is simply `user1`.

Replication: The periodic exchange of database information between the *domain controllers* within a *domain,* which ensures that all domain controllers contain updated, consistent data.

Replication latency: The period between the beginning and end of *replication* to all *domain controllers,* this can result in the *directory* information between domain controllers being temporarily out of synch.

Resource record: Resource records are individual *DNS* entries in the DNS database that clients use to access *domain* services. An Active Directory server, for example, has an SRV (service) record in its DNS entry. Clients use the SRV record to locate an Active Directory server.

Root domain: The first *domain* that you create in your Active Directory structure. The root domain is the topmost level of your domain.

Safe mode: A method of starting a Windows Server with only the basic drivers. You use this mode for troubleshooting the computer.

Schema: Definitions of all *object classes* or *object* categories and their *attributes* that are stored in the Active Directory.

Schema cache: The copy of Active Directory that loads into memory each time a computer running Active Directory boots up. The schema cache serves as a performance enhancement by storing the information in RAM, which provides faster access than reading from disk.

Security Accounts Manager (SAM): The Windows NT database that contains user accounts and related security information.

Security Identifier (SID): A unique identification number for an *object* on the network. Users, groups, and resources all have a unique SID that distinguishes them from all other objects.

Server Core: This is an installation option of Windows Server 2008 where only the minimal services and executables are installed. The server interface supports only a command line prompt. The benefit of deploying a server in this mode is the reduction of network exposure because of the reduction in services as well as the reduction in the frequency of patches to be applied.

Session ticket: Encrypted information that the server examines to validate the client as an authenticated *domain* member. If a domain client requests access to a server resource, the *Key Distribution Center* returns a session ticket to the client computer. The server then determines whether to validate the client for access to that resource.

Site: A grouping of *IP subnets* connected by high-speed or high-*bandwidth* links. Sites are part of your network's physical *topology*.

Site link: A connection between two *sites* over which *replication* occurs.

Site link bridge: A connector between two *site links*. A site link bridge essentially creates a *replication* path among available site links.

Subdomain: *See* Child domain.

Subnet: A portion of a segmented network.

Synchronization: Adjusting disparate *directory* services so that they're in tune or their database information matches.

Ticket Granting Ticket (TGT): A *Kerberos object* that contains authentication information and key information for accessing resources. Whenever a user authenticates, the *Key Distribution Center* sends a Ticket Granting Ticket (TGT) with authentication information. The client caches the TGT until it needs to access a *domain* resource and then presents the TGT and requests access to the resource.

Token: Tokens are objects that are issued by a recognized security authority that can be used to prove your identity to another security system.

Tombstone: An *object* that you mark for removal from the database. Suppose, for example, that you remove several computers from the *domain*. You tombstone each computer object to show that it's obsolete. Each tombstoned object remains in the database for 60 days from when you mark it.

Topology: The physical shape or design of a network. Bus, ring, and star are all network topologies.

Transitive trust: Bidirectional trust relationship between two *domains* that extends to other trusted domains.

Tree: *Hierarchical* grouping of *domains* within a contiguous *namespace*.

Tree-root trust: The *transitive trust* relationship between two *trees* in a *forest*.

Trust relationship (or trust): The method that Active Directory uses to join separate *domains* into an administrative model. An Active Directory trust relationship enables users in one domain to access resources in another domain without merging administrative control of the two domains.

Trust relationships can be transitive or nontransitive. Nontransitive trusts are one-way only, meaning that the trust works in only one direction. One domain, the trusted domain, holds the users who require access. The second, or trusting domain, holds resources that the users in the trusted domain want to access. For both domains to trust each other, you must create a second trust relationship.

Active Directory, however, also supports *transitive trusts* that are bidirectional between domains rather than one way, so the system administrator no longer has to create them.

User Principal Name (UPN): The portion of an object's name that's generally recognized as an e-mail address. The UPN consists of the user's logon name and the name of the *domain* in which the user object resides.

Index