

# Microsoft® Active Directory Administration

---

Kevin Kocis

**SAMS**

*800 East 96th St., Indianapolis, Indiana, 46240 USA*

# Microsoft® Active Directory Administration

## Copyright © 2001 by Sams Publishing

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher. No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein.

International Standard Book Number: 0-672-31975-6

Library of Congress Catalog Card Number: 00-102736

Printed in the United States of America

First Printing: December 2000

03 02 01 00 4 3 2 1

## Trademarks

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Sams Publishing cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

## Warning and Disclaimer

Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an “as is” basis. The author and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of programs accompanying it.

### ACQUISITIONS EDITOR

*Neil Rowe*

### DEVELOPMENT EDITOR

*Steve Rowe*

### TECHNICAL EDITOR

*Michael Schene*

### MANAGING EDITOR

*Charlotte Clapp*

### PROJECT EDITORS

*Paul Schneider*

*Elizabeth Finney*

### COPY EDITORS

*Geneil Breeze*

*Chuck Hutchinson*

### INDEXER

*Erika Millen*

### PROOFREADER

*Katherin Bidwell*

### TEAM COORDINATOR

*Vicki Harding*

### MEDIA DEVELOPER

*JG Moore*

### INTERIOR DESIGNER

*Anne Jones*

### COVER DESIGNER

*Anne Jones*

### PRODUCTION

*Ayanna Lacey*

*Heather Hiatt Miller*

*Stacey Richwine-DeRome*

# Overview

	Introduction	1
1	Understanding Active Directory	3
2	Active Directory Architecture	25
3	Managing Domains, Trusts, and DNS	59
4	Managing Users, Groups, and Computers	107
5	Active Directory Security	139
6	Administering Group Policy	171
7	Managing and Modifying Active Directory Schema	203
8	Managing Sites, Replication, and Network Traffic	225
9	Managing Updates with Flexible Single-Master Operations	251
10	Active Directory Reliability and Optimization	279
Appendix	Common Active Directory Utilities	313
	Index	321

# Contents

## **Introduction 1**

### **1 Understanding Active Directory 3**

Major Technical Features of Active Directory .....	4
Manageability .....	4
Security .....	10
Interoperability .....	12
Major Components of Active Directory .....	14
Namespace.....	14
Tree .....	15
Forest .....	16
Domain .....	16
Organizational Unit (OU) .....	17
Site .....	17
NT 4.0 Versus Windows 2000 .....	17
Logical Differences .....	17
Physical Differences .....	18
Administrative Differences .....	18
Microsoft's View of Meta-directory Services.....	19
Active Directory Compared to Novell 5.....	19
Partitions .....	20
Catalogs .....	21
Internet Standards Support .....	22
Summary .....	23

### **2 Active Directory Architecture 25**

Subsystem Architecture .....	26
Security Subsystem .....	26
Directory Service Architecture .....	28
Directory System Agent .....	30
Database Layer .....	30
Extensible Storage Engine.....	31
Protocols, Interfaces, and Services to Active Directory .....	31
X.500 Directory Service .....	31
Lightweight Directory Access Protocol (LDAP) .....	32
Active Directory Services Interface (ADSI) .....	32
Active Directory Replication.....	33

Logical Structure Fundamentals .....	33
Domain Hierarchy .....	33
Active Directory Domain Names .....	35
Tree and Forest Structure .....	36
Physical Structure Fundamentals .....	44
Directory Contents.....	45
Directory Partitions .....	46
Sites .....	54
Sites Versus Domains .....	55
Data Storage .....	55
Summary .....	56

### **3 Managing Domains, Trusts, and DNS 59**

Domain Fundamentals .....	60
Managing Domains .....	61
Adding Domains .....	62
Domain Models .....	63
Managing Trusts .....	64
Trust Relationships .....	64
Adding Trusts .....	66
Modifying Trusts .....	67
Name Resolution in Active Directory.....	68
Naming Standards .....	68
Name Restrictions .....	70
DNS Server Roles .....	71
Resource Records .....	72
Zones and Zone Files .....	75
Lookup Zones .....	76
Dynamic DNS and Zone Transfers .....	76
Zone Transfer.....	78
Integrating DNS and Active Directory .....	80
DNS Installation Wizard .....	81
Configuring Zones.....	83
Heterogeneous Environments .....	87
Using WINS and WINSR Records .....	87
Using UTF-8 Characters Format.....	87
Receiving Non-RFC-Compliant Data.....	88
UNIX/BIND .....	88
DNS Considerations for Active Directory.....	94
DNS and WINS .....	97
DHCP in Active Directory .....	98
Benefits .....	98
New Windows 2000 DHCP Features .....	99

	DHCP Lease Process .....	100
	Integration of DHCP with Dynamic DNS .....	101
	Configuring DHCP .....	103
	Setting DDNS Update on a Scope .....	104
	Summary .....	105
<b>4</b>	<b>Managing Users, Groups, and Computers</b>	<b>107</b>
	Object Management Fundamentals .....	108
	Managing Users .....	110
	User Accounts .....	110
	Predefined User Accounts .....	110
	Adding and Deleting Users .....	111
	Modifying Users .....	113
	Locating Users.....	116
	Moving User Accounts .....	116
	Managing User Profiles and Home Directories .....	117
	Advantages of User Profiles .....	117
	Profile Types .....	117
	Advantages of Home Directories .....	119
	Managing Groups .....	120
	Group Types .....	120
	Group Scope and Replication Traffic .....	126
	How Domain Mode Affects Groups .....	127
	Modifying Groups .....	128
	Converting Group Type .....	129
	Replication Conflicts .....	132
	Managing Computer Accounts .....	133
	Creating Computer Accounts .....	133
	Locating Computers .....	135
	Editing Computer Accounts .....	135
	Resetting Computer Accounts .....	136
	Enabling and Disabling Computer Accounts .....	136
	Summary .....	138
<b>5</b>	<b>Active Directory Security</b>	<b>139</b>
	The Active Directory Security Model.....	140
	Authentication in Active Directory .....	140
	Kerberos Authentication .....	140
	Kerberos Preauthentication .....	143
	Public Key Infrastructure .....	146
	IPSec .....	148

Object-Oriented Security .....	155
Access Control Lists (ACLs) .....	155
Rights and Permissions .....	157
Security Descriptor .....	158
Active Directory Object Security .....	160
Active Directory Objects .....	160
Publishing Active Directory Resources .....	164
Publishing Shared Folders .....	164
Publishing Printers .....	166
Publishing Guidelines .....	167
Ownership and Delegation .....	167
Permission Inheritance .....	168
Best Practices for Access Control .....	168
Summary .....	169
<b>6 Administering Group Policy 171</b>	
Group Policy Fundamentals .....	172
Windows NT 4.0 and Windows 2000 Policy Comparison .....	172
Group Policy Administrative Requirements .....	173
Group Policy Objects (GPOs) .....	174
Creating Group Policy Objects (GPOs) .....	176
Configuring Group Policy .....	177
MMC Snap-in Extension Model .....	180
Settings and Templates .....	182
Special Policies (Account Policies) .....	194
Linking GPOs .....	194
Inheritance .....	195
Delegation of GPO Administration .....	197
Mixed Mode Group Policy .....	199
Filtering and Delegating Group Policy with Security Groups .....	200
Multiple Group Policy Objects .....	201
Trust Relationships with Previous Versions of Windows .....	202
Summary .....	202
<b>7 Managing and Modifying Active Directory Schema 203</b>	
Active Directory Schema Fundamentals .....	204
Schema Structure: Exploring the Directory Information Tree .....	206
Starting the Active Directory Schema Snap-In .....	207
Active Directory Schema Objects .....	209
Schema Modification .....	212
Planning to Extend the Schema .....	212
Adding a Class .....	219
Verifying Schema Modifications .....	221

Issues with Extending Schema .....	222
System Checks for Schema Modifications .....	222
Deactivating Schema Objects .....	223
Deactivating Existing Classes and Attributes .....	224
Summary .....	224
<b>8 Managing Sites, Replication, and Network Traffic</b>	<b>225</b>
Site Topology Fundamentals .....	226
Sites .....	227
When to Create a New Site .....	228
Subnets.....	230
Connections .....	231
Site Links.....	233
Bridgehead Servers .....	235
Site Link Bridges.....	237
Active Directory Replication Model .....	239
Directory Partition Replicas .....	240
Benefits .....	240
Replication Components .....	241
Updates .....	243
Replication Topology .....	244
Transports and Protocols .....	244
Intrasite Replication.....	246
Intersite Replication.....	247
Replication Tools.....	248
Summary .....	249
<b>9 Managing Updates with Flexible Single-Master Operations</b>	<b>251</b>
Flexible Single-Master Fundamentals .....	252
FSMO and Directory Schema Updates .....	252
Operations Master Roles and Placement .....	253
Schema Master .....	256
Domain Naming Master .....	258
Relative Identifier Master .....	260
Primary Domain Controller Emulator (PDCE).....	261
Infrastructure Master .....	263
Placing Flexible Single-Master Operations.....	265
Performing Operations Master Role Transfers .....	266
Using the ntdsutil Tool for Role Placement .....	267
Using the ntdsutil Tool for Role Location.....	269
Schema Master Permission Changes.....	270
Domain Naming Master Permission Changes .....	270
PDCE Permission Changes .....	270



Infrastructure Master Permission Changes .....	270
RID Master Permission Changes .....	271
Controlling Role Transfers .....	271
Controlling Role Seizures.....	272
Operations Masters Troubleshooting .....	273
Responding to Operations Master Failures .....	273
Primary Domain Controller Emulator Failures .....	274
Infrastructure Master Failures .....	274
Other Operations Master Failures .....	274
Troubleshooting and Technical Details .....	276
Other FSMO Errors and Clarifications .....	277
Summary .....	278
<b>10 Active Directory Reliability and Optimization 279</b>	
Utilities .....	280
Active Directory Backup .....	280
Performing an Active Directory Backup .....	281
Using the Backup Utility's Backup Wizard .....	282
Scheduling Backups .....	283
Active Directory Restore .....	284
Restoring Active Directory Through Reinstallation and Replication .....	285
Restoring Active Directory .....	285
Restoring Active Directory to Dissimilar Hardware .....	290
Performing an Authoritative Restore.....	290
Monitoring Active Directory Performance.....	295
Monitoring Domain Controller Performance .....	295
System Monitor .....	295
Performance Logs and Alerts .....	301
Task Manager .....	307
Event Logs.....	309
Network Monitor .....	310
Summary .....	312
<b>A Common Active Directory Utilities 313</b>	
ADSI (Active Directory Service Interface) .....	314
Comma-Separated Value Directory Exchange (CSVDE) .....	314
LDAP Data Interchange Format Directory Exchange (LDIFDE) .....	316
Movetree .....	319
Movetree Syntax .....	319
<b>Index 321</b>	

## About the Author

**Kevin Kocis**, MCSE has been working in the Information Technology field for over 10 years. He is currently the Manager of Information Technology for a division of a Fortune 100 company, where he strategizes network and server infrastructure, Oracle implementation, and platform and interoperability issues (Windows, UNIX, and Macintosh). He has written several articles and reviews for Microsoft Certified Professional magazine. In addition, he has presented on Windows 2000 Server at TechMentor conferences in Atlanta and San Francisco.

# Dedication

*This book is dedicated to my loving family, who tolerated the endless piles of drafts, the miles of cat5 cable coiling through the house, the endless supply of MP3 files blaring until midmorning—only a loving family could tolerate such nonsense. Mucho thanks to Patto mommy, J-man, JordaBean, and Baby Jen. I love you guys dearly. Thanks for your understanding.*

# Acknowledgments

Nobody ever writes a book alone. First, doors are opened (thank you, Linda and Dian at MCP magazine), friendships are created, and your arm is twisted to write (thanks to Harry Brelsford for introducing me to long nights of solitude). Then publishers confide in you and convince you that you're something you're not—like a writer (thanks to Neil Rowe, Steve Rowe, and all the editors at Macmillan USA for their encouragement—and their Starbucks). A special thanks to Michael Schene for his keen eye and great late night conversations. You are truly a diamond in the technical editing rough, my man!

Thanks to my friends and supporters: Ronnie G, Don Jordan, JJF, and Connie (for the Grand Canyon). My great vendor contacts at EMC (Jay, Pete, Debra, and who could forget JB!), InfoTech Systems (Bob and Susan), and TekSystems (Mr. Bill, Nick, and Alicia).

There's also the music (thanks to Geddy, Neil, and Alex) and the NASCAR races that kept me in the reality loop (thanks to Jeff, Bobby, and Tony—you guys rock!).

I wish to thank my friends and co-workers who provide me with endless inspiration and workload. Thanks, Stango!

A special acknowledgment to Rich and Kayme, my brother and sister, whom I love more than I tell them and more than they'll ever know.

A final thanks to my parents, Frank and Mellody Kocis, who are still afraid of computers and continue to love me unconditionally, regardless of the hell I put them through so many years ago...

# Tell Us What You Think!

As the reader of this book, *you* are our most important critic and commentator. We value your opinion and want to know what we're doing right, what we could do better, what areas you'd like to see us publish in, and any other words of wisdom you're willing to pass our way.

As an Associate Publisher for Sams Publishing, I welcome your comments. You can fax, email, or write me directly to let me know what you did or didn't like about this book—as well as what we can do to make our books stronger.

*Please note that I cannot help you with technical problems related to the topic of this book, and that due to the high volume of mail I receive, I might not be able to reply to every message.*

When you write, please be sure to include this book's title and author as well as your name and phone or fax number. I will carefully review your comments and share them with the author and editors who worked on the book.

Fax: 317-581-4770

Email: [feedback@sampublishing.com](mailto:feedback@sampublishing.com)

# Introduction

It's time for a confession. I completely realize that no book will ever serve as the ultimate Windows 2000 Active Directory guide. Active Directory's extensive functionality demands the capacity of an encyclopedia set, with a matching set to assist in interpreting Microsoft's language. Active Directory is, by far, the most demanding facet of Windows 2000 Server.

Planning for Active Directory is a serious task that should not be taken lightly. Incorporating planning into this book could have easily doubled its length. Many companies will struggle with how to get Active Directory just right, others will fight its integration into their current structure, and many will wait to see who actually jumps on this Active Directory thing. For these reasons, I chose to focus predominantly on Active Directory administration.

When I was asked to write this book, I set out with some major objectives:

- To write an understandable guide
- To cover the major administrative tasks and issues
- To break down the complexity of Active Directory

As of the writing of this book, the acceptance and sales of Windows 2000 in the enterprise is not as prolific as Microsoft had hoped. Although I don't read too much into this, it could mean any number of things:

Active Directory is too new and unproven.

Active Directory is too difficult to manage.

Test labs are amuck with avid Windows 2000 administrators.

I would like to think the latter.

Despite my long hours buried in endless piles of technical resource papers, reviews, journals, and correspondence, I realize that I could never give Active Directory justice in a book of this size. Hopefully, I have provided you with a solid step toward technically mastering Active Directory.

Because I've always been poor at introductions, let's get on with the show.

## Conventions Used in This Book

This book uses different typefaces to differentiate between code and regular English, and also to help you identify important concepts.

Text that you type and text that should appear on your screen is presented in monospace type.

It will look like this to mimic the way text looks on your screen.

Placeholders for variables and expressions appear in *monospace italic* font. You should replace the placeholder with the specific value it represents.

This arrow (➡) at the beginning of a line of code means that a single line of code is too long to fit on the printed page. Continue typing all characters after the ➡ as though they were part of the preceding line.

**NOTE**

A Note presents interesting pieces of information related to the surrounding discussion.

**TIP**

A Tip offers advice or teaches an easier way to do something.

**CAUTION**

A Caution advises you about potential problems and helps you steer clear of disaster.

# Understanding Active Directory

CHAPTER

**1**

## IN THIS CHAPTER

- Major Technical Features of Active Directory 4
- Major Components of Active Directory 14
- NT 4.0 Versus Windows 2000 17
- Microsoft's View of Meta-directory Services 19
- Active Directory Compared to Novell 5 19

Active Directory (AD) is Microsoft's solution to the enterprise-level shortcomings of Windows NT 4.0 and is a revolution for administrators in contrast to NT 4.0.

Although Active Directory has a very intricate architecture, its purpose is to ease the lives of system administrators at the enterprise level. From Microsoft's perspective, Active Directory ranks as an enterprise-class directory service based on Internet standards, providing information and the necessary services to users. Active Directory is provided with all Windows 2000 server products and is Microsoft's implementation of an existing model (X.500), an existing communication protocol (LDAP), and an existing location technology (DNS).

This chapter focuses on the major technical features and components of Active Directory, as well its growth and differences from Windows NT 4.0 in the enterprise.

In simple terms, Active Directory is a distributed database, expanding across multiple Windows 2000 servers in a network. Active Directory is built on Internet standards and is quite scalable in contrast to Windows NT 4.0, despite its centralized management structure. Active Directory updates and modifications are circulated via multi-master replication throughout the network. The Active Directory database schema is expandable and object-oriented, meaning it represents each account and resource as an object. Because the database is structured in a hierarchical tree, it allows for easier organization, permission assignment, and delegation of authority. Active Directory is efficient, and its data is both redundant and load-balanced. Active Directory replaces the weak domain functionality from NT 4.0.

Although this book focuses on the administration of Active Directory, I will present some of the key features and components that enhance administration.

## Major Technical Features of Active Directory

Active Directory has many advanced technical features. According to Microsoft, most of the features can be categorized into three key areas:

- Manageability
- Security
- Interoperability

These issues evolved from experience and shortcomings with Windows NT 4.0 in the enterprise. Let's take a closer look at these key areas.

### Manageability

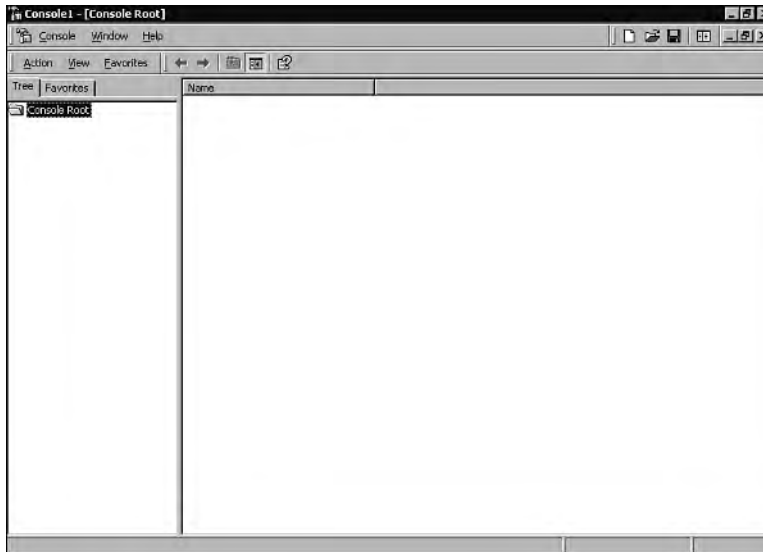
Active Directory has many features that can be categorized into the area of manageability. These features are important because previous versions of Windows NT were quite lacking in this area, particularly at the enterprise level. The following sections describe some of the key manageability enhancements in Active Directory.



## Centralized Management

Active Directory centrally manages Windows users, clients, and servers through a single consistent management interface, reducing redundancy and maintenance costs. The Microsoft Management Console (MMC) is a tool for hosting various administrative consoles, which may include folders or Web pages. MMC was introduced in Internet Information Services 4.0 (IIS) but not widely implemented.

Simply put, the MMC is a tool acting as an interface to other tools (called snap-ins). The main MMC window provides commands and tools for authoring consoles, as shown in Figure 1.1. The console itself performs no exclusive function. Additional MMC components (snap-ins) are added for custom administration. Microsoft and third-party companies develop these MMC snap-ins. Snap-ins always reside in a console and do not function independently.



**FIGURE 1.1**

*The MMC console without snap-ins.*

The MMC runs in two modes: author and user. Author mode allows for customization, whereas user mode restricts a user from saving or modifying the console configuration.

To set the MMC mode, perform the following steps:

1. Go to Start, Run, and type **mmc**. Then press the Return key (this launches the console). You can launch the console from a command prompt by typing **mmc**.
2. Under the Console menu, select Options.

3. Assign a name to the console. (Console1 is the default.)
4. In the Console Mode window, select the desired mode.
5. In the User Mode window, you can modify restrictions by selecting the check boxes at the bottom of the Options window.
6. Click Apply, then OK.

The Active Directory administrative tools can be used only from a computer with access to a Windows 2000 domain. Some tools are available also as snap-ins to the MMC console. The three core Active Directory administrative tools available through the MMC on Windows 2000 domain controllers are

- Active Directory Users and Computers
- Active Directory Domains and Trusts
- Active Directory Sites and Services

The Active Directory Users and Computers MMC snap-in is one of the most useful tools for administering your Active Directory. It replaces and improves upon the Server Manager and User Manager from Windows NT 4.0.

To add these snap-ins to the MMC console, perform the following steps:

1. From the Console menu, select Add/Remove snap-in (or press Ctrl+M).
2. Click Add in the Standalone panel of the Add/Remove snap-in window.
3. Select the desired snap-in, and click Add. (Repeat this procedure for all the snap-ins.) Then click Close.
4. Click OK in the Add/Remove snap-in window.

The Active Directory administrative tools included with Windows 2000 Server simplify directory service administration and can be customized to individual administrators with specific administrative responsibilities. You can also create scripts that use Active Directory Service Interfaces (ADSI). ADSI will be addressed in Chapter 2, “Active Directory Architecture.”

#### **NOTE**

Intellimirror is a Microsoft term that may disappear in the future as AD matures. The Intellimirror role has changed since the initial concept of NT5 and now actually addresses concepts such as Group Policy and software distribution.

## Group Policy

Group policy allows you to define and control the policies for users, groups, and computers. Group policy can be set at the site, domain, or organizational unit (OU) level in Active Directory.

You can use policies to define the permitted actions and settings for users and computers, such as logon/logoff scripts, security, and desktop and Registry settings. Policy-based management simplifies tasks such as operating system updates, installing applications, managing user profiles, and locking down desktop systems.

Group policy is powerful and somewhat complicated. Despite its complex nature, most companies still benefit by implementing some form of Group policy.

### NOTE

Group policy can be regarded as a combination of NT 4.0 policies and zero administration. For many larger enterprises, NT 4.0 laptops with zero administration proved restrictive if sites were configured differently (such as no DHCP), and the need arose to configure the machine locally.

See Chapter 6, “Administering Group Policy,” for more information.

## Software Distribution

With Active Directory, you can automatically distribute applications to users. For example, all engineers can automatically receive CAD or CAM design software on their Windows 2000 client machines.

The Software Installation snap-in is an MMC snap-in that extends Group policy. This snap-in allows you to manage the installation of software on users’ computers. It does so by two methods: publishing and assigning software. Assigned software is available on either a per-user or per-computer basis from the Start menu. Users gain access to published software by using Add/Remove Programs in Control Panel.

## Global Catalog

The Global Catalog (GC) is a subset of Active Directory—an index containing all the objects in the forest, as well as a subset of each object’s properties. The Global Catalog allows users to search by selected attributes to find an object easily anywhere in the forest.

The Global Catalog allows users to locate any objects they have appropriate access to. In contrast to the Find Computer feature in NT 4.0, users can search for any object within Active Directory such as servers and printers. You must have proper access to an object you are searching for in the GC.

**NOTE**

The Global Catalog offers a faster search of AD objects because it contains all the objects in the forest. However, it holds only certain selected attributes of each object, allowing for faster searches.

Every domain controller (DC) in a forest stores three full, writable directory partitions: a domain directory partition, a schema directory partition, and a configuration directory partition. A Global Catalog server is a domain controller that stores these writable directory partitions, as well as a partial, read-only copy of all other domain directory partitions in the forest.

**Transitive Trusts**

In Windows 2000, transitive trust relationships are always two-way trust relationships. A trust relationship between a Windows 2000 domain and a Windows NT 4.0 domain is always a non-transitive trust relationship that must be established manually.

A transitive trust works in the following manner: If domain A trusts domain B, and domain B trusts domain C, then domain A implicitly trusts domain C. A two-way trust implies that domain C also trusts domain A.

**NOTE**

In NT 4.0, trust relationships were created explicitly in one direction. Establishing two one-way trusts created a two-way trust relationship.

For Active Directory to automatically configure a replication topology, all adjacent domain trust relationships within the forest are two-way and transitive.

In Windows 2000, domains can be joined to a domain tree or forest, and each sub domain (called a child domain) has an automatic two-way transitive trust established with its parent domain. Transitive trusts are also applied automatically for all domains that are members of the domain tree or forest.

Transitive trust agreements greatly reduce the number of trust relationships to manage between Windows domains. However, these trusts can inhibit authentication response as the domain tree or forest expands. For more information, see Chapter 3, “Managing Domains, Trusts, and DNS.”

## Flexible Single-Master Operations (FSMOs)

Despite the elimination of Primary Domain Controllers (PDCs) and Backup Domain Controllers (BDCs) from the Windows 2000 Server structure and the equality of Windows 2000 domain controllers, certain operations must occur at only one domain controller. They are called single-master operations. The domain controllers hosting these operations are called operation master roles.

The domain controllers assigned to manage single-master operations are called role owners for the operations. The five single-master operations include the following:

- Domain Naming Master
- Schema Modification Master
- Primary Domain Controller Emulation Master
- Relative ID Master
- Infrastructure Master

For more information about managing flexible single-master roles, see Chapter 9, “Managing Updates with Flexible Single-Master Operations.”

## Active Directory Services Interfaces (ADSI)

Active Directory Services Interfaces (ADSI), which is a tool for browsing Active Directory, reveals the internal structure of the database and performs a regedit-like role. ADSI greatly simplifies the development of directory-enabled applications, as well as the administration of distributed systems. ADSI should not be used for common administrative tasks for which MMC snap-ins would suffice (because they’re simpler and more fail-safe).

With ADSI, you can script applications to interact with other directory services without knowing all the details of the underlying protocols. You can write programs and scripts that make use of ADSI to read or write to various legacy directories.

For more information on ADSI, see Chapter 2.

## Backward Compatibility

Windows 2000 Domain Controllers support a mixed environment of client computers. Pre-Windows 2000 computers respond as though they are accessing an NT 4.0 domain controller.

WINS name resolution is still supported for down-level clients. For clarification

- Windows 2000-based clients use the DNS name.
- All down-level Windows 9x and Windows NT 4.0-based clients use the NetBIOS name for backward compatibility (which is created when the domain controller is first installed).

An Active Directory domain can consist of heterogeneous clients, including Windows NT 4.0, Windows 95 and 98, Macintosh, and UNIX workstations. Although clients have full access to shared resources within the domain (Macintosh and UNIX may require additional configuration), only Windows 2000 clients and Windows 9x clients (with the Active Directory client software installed) can use Active Directory to query information about these shared resources.

**NOTE**

An Active Directory client for NT 4.0 machines is planned for a future release.

## Multi-Master Replication

With multi-master replication, any update or modification made to any domain controller is copied to all the other domain controllers in the same domain. This process ensures that the directory is available even if a DC fails and assists with controlling bandwidth by providing multiple copies of the directory across multiple servers.

## Security

Security is one of the enhanced features in Active Directory. Windows 2000 and Active Directory security are completely integrated and more robust than previous versions of NT. Many new standards and components contribute to this effort; they're described in the following sections.

### Kerberos Authentication

Active Directory supports the Kerberos V5 protocol, which provides quick, single sign-on to Windows 2000-based resources. Additional benefits include mutual and delegated authentication.

The Kerberos protocol allows negotiation of the encryption algorithm. Most Kerberos implementations default to the Data Encryption Standard (DES), which has a 56-bit key length. Although Windows 2000 Kerberos does support DES for interoperability purposes, its default choice for an encryption algorithm is RC4. In North America, 128-bit RC4 keys are used, whereas the international version supports only 56-bit keys. A proprietary Windows 2000 environment uses only the RC4 algorithm.

For more information on Kerberos, see Chapter 5, "Active Directory Security."

### Public Key Infrastructure and X.509 Certificates

Active Directory supports public key infrastructure (PKI) and X.509 certificates to interoperate with extranet and e-commerce application deployment.

The Windows 2000 PKI provides the framework of services, technology, protocols, and standards that enable you to deploy and manage a strong information security system.

The major components of the Windows 2000 public key infrastructure include the following:

- Windows 2000 Certificate Services—issues and manages digital certificates
- Microsoft CryptoAPI and cryptographic service providers (CSPs)—provides cryptographic operations and manages private keys
- Certificate stores—store and manage certificates

For more information on PKI and its related services, see Chapter 5.

## LDAP over SSL and ACL Support

According to Microsoft, Active Directory supports Lightweight Directory Access Protocol (LDAP) over the Secure Sockets Layer (SSL) for secure directory transactions for extranet and e-commerce applications.

SSL is a proposed open standard developed by Netscape Communications. SSL allows you to establish a secure communications channel, preventing the interception of critical information, such as credit card information.

Access Control Lists (ACLs) through LDAP provides secure connections and interoperability for extranets and e-commerce applications.

## Required Authentication

Required authentication allows administrators to determine and require the specific type of logon needed for user authentication, including Kerberos, x.509 certificate, or NT LAN Manager (NTLM) (for situations involving backward compatibility).

## Attribute-Level Security

All objects in the Active Directory are protected by Access Control Lists (ACLs). ACLs determine who can see the objects and what actions each user can perform on the objects. An object's existence is never revealed to a user without permission to view it. The Global Catalog (GC) enforces object- and attribute-level security for detailed control of access to information stored in the directory.

An ACL is a list of Access Control Entries (ACEs) stored with the object it protects. In Windows 2000, an ACL is stored as a binary value called a security descriptor. Each ACE contains a security identifier (SID), which identifies the principal (user or group) to whom the ACE applies and information on what type of access the ACE permits or denies.

ACLs on directory objects contain ACEs that apply to the object and ACEs that apply to the individual attributes of the object. With this new feature, you can control not just who can view an object, but also what properties the user can see.

For more information about ACLs and ACEs, see Chapter 5.

## **Spanning Security Groups**

Because of the transitive trusts established in Active Directory, no restrictions are placed on security groups that span domains. This means that groups throughout the forest can be managed centrally (if the appropriate permissions are assigned).

## **Delegated Administration**

Windows 2000 allows you to delegate privileges and rights to different containers and objects in the directory to certain other lower-level administrators or approved users.

With delegation, you can grant certain rights for containers and subtrees to users and groups, which eliminates the need for a group of domain administrators to have full control over large segments of the user population (such as the Enterprise Administrators group). Allowing a designated user or group in a particular domain or OU to perform administrative tasks alleviates the administrative burden on higher-level administrators in the enterprise.

## **Interoperability**

A key area of concern for administrators in heterogeneous environments is Active Directory's capability to integrate with current and legacy business-critical systems. Although Active Directory does not provide a client solution allowing UNIX, Novell, and Macintosh to utilize Active Directory searches, such a solution may be possible in the future through the use of a third-party utility.

Despite its apparent lacking with Active Directory, Windows 2000 Server still delivers UNIX, Novell NetWare, Windows NT 4.0, and Macintosh interoperability, allowing you to integrate Windows 2000 Server into an existing environment. You can introduce Windows 2000 Server into your environment as it provides migration paths from different systems, devices, and applications.

## **Domain Name System (DNS) Standards**

Windows 2000 uses Domain Name System (DNS) standards for hierarchical naming of Active Directory domains and computers. Domain and computer objects are part of both the DNS domain hierarchy and the Active Directory domain hierarchy. Even though the objects have identical names, these domain hierarchies represent separate namespaces. Chapter 4, "Managing Users, Groups, and Computers," addresses DNS and namespaces in greater detail.

## **Native LDAP**

Active Directory is implemented as a native LDAP server that eases interoperability in extranet environments and e-commerce applications.



**NOTE**

Although Active Directory attempts to maintain itself as a strong meta-directory candidate, many heterogeneous environments may refrain from integrating as such because of its newness in the directory services arena. AD may need time to prove itself first, particularly in the areas of reliability and interoperability.

Active Directory reflects Microsoft's trend toward relying on standard protocols. The Lightweight Directory Access Protocol (LDAP) is a product of the Internet Engineering Task Force (IETF). It defines how clients and servers exchange information about a directory. LDAP versions 2 and 3 are used by Active Directory.

Active Directory servers provide the LDAP service for object location, and LDAP relies on TCP as the underlying transport layer protocol. Therefore, a client searching for an Active Directory server within the kevinkocis.com domain, for example, would look up the DNS record for ldap.tcp.kevinkocis.com.

**NOTE**

Active Directory uses both DNS and LDAP services to locate objects in Active Directory.

**Extensible Schema**

In Active Directory, the schema is the set of attributes used to describe a particular object class. Different types of information must be tracked for different object classes. The schema is stored within Active Directory just like other objects and can be automatically replicated throughout your enterprise. It also uses Active Directory security, so you can delegate authority over the schema to different users and groups. By changing the ACLs on a schema object, an administrator can allow any user to add or modify attributes for an object class.

Active Directory allows you to extend the directory schema and create new properties and objects.

For more information on modifying the schema, see Chapter 7, "Managing and Modifying Active Directory Schema."

**Microsoft Directory Synchronization Services (MSDSS)**

Microsoft Directory Synchronization Services (MSDSS) is an add-on service that permits two-way synchronization between Active Directory and Novell Directory Services (NDS). MSDSS also synchronizes directory information stored in Active Directory with all versions of NetWare 3.x bindery services on a one-way basis.

With MSDSS, you can deploy Active Directory and not have to replace your existing NDS directory or manage two separate directories. You can manage accounts from either directory and use directory-enabled applications such as Microsoft Exchange 2000.

**NOTE**

MSDSS is based on DirSync, a proposed IETF standard.

## Active Directory Connectors (ADC)

Active Directory Connector (ADC) consists of a replication and mapping engine that provides directory synchronization and import/export tools. ADC allows you to replicate an Exchange 5.5 directory with Active Directory and offers connectivity to NDS. You can then manage the information using the Active Directory snap-ins.

## Open APIs

Active Directory can integrate with other applications, directories, and devices through LDAP, ADSI, and Messaging API (MAPI). ADSI provides an object-oriented interface to the Active Directory. Administrators who frequently utilize scripts will benefit from ADSI. The LDAP C API is a lower-level interface. MAPI is supported for backward compatibility only.

## DEA Platform

Active Directory provides a directory-enabled application (DEA) platform that lets applications use the directory to automate installation, distribution, and maintenance.

## DEN Platform

Active Directory, combined with hardware and software support from Cisco Systems, introduces a directory-enabled networking (DEN) platform that allows administrators to allocate network bandwidth and quality of service.

# Major Components of Active Directory

As you'll notice, Active Directory has many new technical features. Let's briefly review the major components.

## Namespace

A namespace is a defined area where standardized names can be used to symbolically represent some type of information or data (such as an IP address) and that can be resolved to the object itself. The domain hierarchy (which will be addressed in a moment) defines a namespace. In each namespace, specific rules determine how names can be created and used. In the

case of the DNS namespace (and the Active Directory namespace for that matter), the namespace is hierarchically structured and offers rules that allow partitioning of the namespace. The Active Directory namespace is directly related to DNS. An example of such a namespace is kevinkocis.com.

**NOTE**

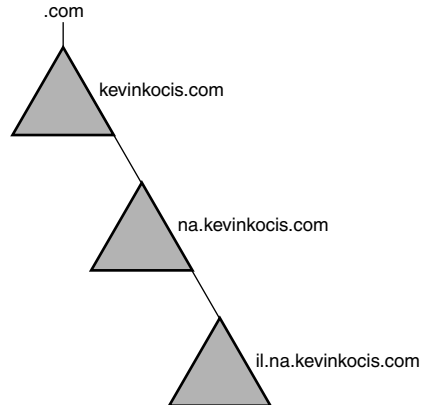
An Active Directory forest can have one or more namespaces.

Other namespaces, such as the NetBIOS namespace, are flat (non-hierarchical) and cannot be partitioned.

Chapter 2 addresses naming conventions in more detail.

## Tree

The Active Directory domain tree, like the DNS tree, is created in an inverse fashion with the root at the top (see Figure 1.2).



**FIGURE 1.2**

*An Active Directory tree.*

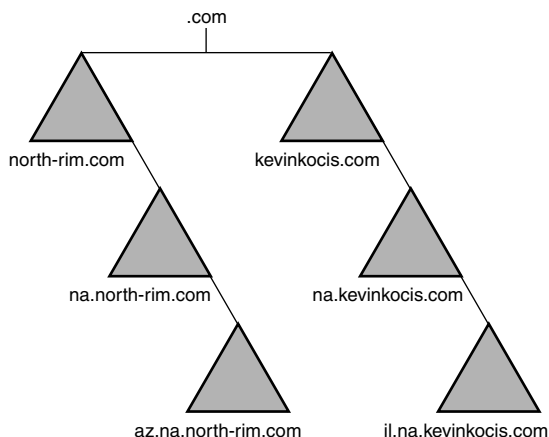
A tree is a hierarchy of objects and containers. A tree logically displays how objects are connected, or the paths between objects. A tree usually consists of containers, which serve as repositories for domain objects. A simple directory is a container. A computer network or domain is also a container. The endpoints on the tree are usually objects called *leaf nodes*. These nodes are considered non-containers because they cannot contain other objects.

A contiguous subtree is any unbroken path in the tree, including all members of any container in that path.

A domain tree (a tree) is made up of one or more domains that form a contiguous namespace and share a common schema and configuration. Domains in a tree are linked together by two-way transitive trust relationships. The Active Directory is a set of one or more trees.

## Forest

A forest is a collection of one or more domain trees that do not form a contiguous namespace (referred to as *noncontiguous* or *disjointed*), even though they share a common schema, configuration, and Global Catalog (see Figure 1.3). While trees in a given forest do not share a common root, they trust each other via transitive hierarchical Kerberos trust relationships. Unlike a tree, a forest does not need a distinct name. Domain trees are linked together in a forest via two-way, transitive trusts, allowing users with appropriate permissions to access resources in any domain in the forest.



**FIGURE 1.3**

*An Active Directory forest.*

## Domain

A domain is a logical security boundary of a Windows NT or Windows 2000 computer network. The Active Directory is made up of one or more domains. A domain can span more than one physical location, and hosts its own security policies and security relationships with other domains. When multiple domains are connected by trust relationships and share a common schema, configuration, and Global Catalog, you have a domain tree.

## Organizational Unit (OU)

If you are familiar with Exchange 5.5 and higher, you understand the concept of the organizational unit. The OU in Active Directory provides another level of partitioning to the logical namespace. OUs act as containers that can contain other OUs and objects, including users, groups, and computers. You can organize your objects into logical containers depending on variables such as geography or organizational structure. Group policy is based at the OU level, making this a critical and important feature of Active Directory.

For more information about Group policy, see Chapter 6.

## Site

Unlike domains, sites are related to physical areas. They are not logical, but based on connectivity and proximity.

A site is a location in a network that contains Active Directory servers, and is defined as one or more well-connected TCP/IP subnets. Microsoft defines “well-connected” as a network where connectivity is highly reliable and fast (for example, LAN speeds of 10 million bits per second or greater). Defining a site as a set of subnets allows administrators to quickly and easily configure the Active Directory access and replication topology to take advantage of the physical network. When a user logs on, the Active Directory client finds DCs in the user’s site. Because computers in the same site have high network proximity, communication is reliable, fast, and efficient. Determining the local site at logon time is accomplished easily because the user’s workstation already knows what TCP/IP subnet it is on, and subnets translate directly to Active Directory sites.

## NT 4.0 Versus Windows 2000

Enterprise administration and centralization were quite challenging with Windows NT 4.0. The NT 4.0 domain model was restrictive in terms of logical and physical models, and limited in terms of centralized administration. Let’s take a closer look at these differences.

### Logical Differences

Windows NT 4.0 provided four domain models: single domain, master domain, multiple master domain, and complete trust domain. These domain models didn’t scale well and as a result required significant administrative overhead. Enterprise administrators were required to establish multiple trusts, creating a web of redundant trusts. Administrators who denied trusts where users required access were forced to set up share permissions with secondary logons. This process grew quite complex (and it remains so in many large-scale NT 4.0 enterprise implementations).

In Windows 2000, the four domain models have evolved to a complete trust model in which all trusts are inherently two-way and transitive (except in the case of trusts with NT 4.0 domains, where they still need to be created manually). This eliminates the requirements of any secondary logon or creation of “web” trusts and can reduce administrative overhead.

## Physical Differences

Windows NT 4.0 used Primary and Backup Domain Controllers (PDCs and BDCs) for authentication. Domains required a single PDC and as many BDCs as necessary for local authentication. When a PDC failed, one of the BDCs needed to be promoted to the role of PDC.

In Windows 2000, all domain servers are considered domain controllers (DCs). Domain controllers use multi-master replication and automatically “back up” each other in the event of unavailability.

Windows NT 4.0 required either the NetBIOS service for browsing or authentication in a routed environment, whereas Windows 2000 Active Directory relies on DNS (WINS for backward compatibility).

The smallest unit of replication in Windows NT 4.0 was the object itself. In Windows 2000, attributes of objects are the smallest units of replication. This is critical to reduce replication traffic.

## Administrative Differences

In Windows NT 4.0, the SAM database was limited to 40MB (although many enterprises exceeded this limit successfully). The Windows 2000 Active Directory database size limit is 70TB (that’s a significant difference!). This increase brings the maximum user count from 40,000 or so to one to two million in a single domain.

In Windows NT 4.0, the domain was the smallest unit of authentication and policy. In Windows 2000, the smallest unit of authentication is the organizational unit (OU). The smallest unit of security delegation in Windows NT 4.0 was the domain. In Windows 2000, security can be delegated at the object property level.

In Windows NT 4.0, there was no central repository for information. Each domain was responsible for centralizing its data but was not searchable. To locate an object, such as a share or computer, you were required to know the name of the domain and the server where the object resided. With Active Directory, you can search the entire forest for a computer without knowing its exact location.

In Windows NT 4.0, sites were nonexistent unless you had installed Microsoft Exchange, whereas in Windows 2000, sites are integral parts of the physical structure.

## Microsoft's View of Meta-directory Services

Microsoft views Active Directory as a major component of its Meta-directory Services, which enhances Active Directory by providing these services:

- Directory synchronization
- Unification of object views
- The assignment of authoritative directory sources for an attribute
- A simple and flexible environment

According to Microsoft, future releases of MMS will further integrate with Active Directory and customer needs by including the following:

- An optimized Active Directory Management Agent to better utilize Active Directory's advanced replication protocol.
- All authorized access to the meta-directory namespaces will use Active Directory for authentication.
- Integration of Microsoft's Gateway Service for NetWare.
- Integration with Windows 2000 Server.

Data and directory management in today's enterprises holds many challenges because many enterprises are managed in a very decentralized manner. A meta-directory collects all the identity data in one place and provides tools for managing the data, despite its format. This process substantially reduces administration costs and data duplication.

### NOTE

Microsoft Meta-directory Services evolved from ZOOMIT VIA.

## Active Directory Compared to Novell 5

Although the real impact or acceptance of Active Directory remains to be seen, a recent Gartner Group survey indicates that many enterprises will adopt Windows 2000 and implement Active Directory to some degree within the year 2001. The following sections explore the differences between Active Directory and its most competitive service, Novell Directory Service(NDS), which ships with Novell 5.0 and higher.

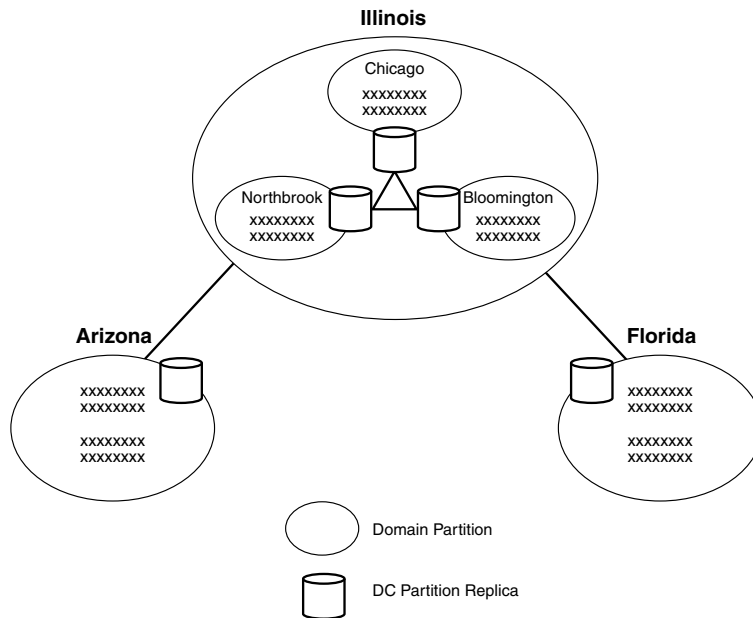
## Partitions

Partitions are an integral part of the data replication in Active Directory and Novell Directory Services. AD and NDS handle directory partitions quite differently, as you'll see in the next few sections.

### Active Directory Partitions

In Active Directory, the boundary of a partition is a domain. Active Directory supports multi-master replication; therefore, a full replica of the partition is available on all domain controllers within the domain, even if the domain spans multiple sites.

In Figure 1.4, the Illinois domain consists of three sites: Chicago, Northbrook, and Bloomington. A full replica of the domain partition is available on all DCs.



**FIGURE 1.4**

*An Active Directory partition design.*

With Windows NT 4.0, few companies created domains that spanned sites. With Active Directory, the domain model expands dramatically because it can effectively manage more information about every object in a domain. It can also allow administrative delegation to local administrators and approved users. Lastly, Active Directory's replication between sites is optimized to reduce the need to create domain boundaries based solely on network bandwidth.

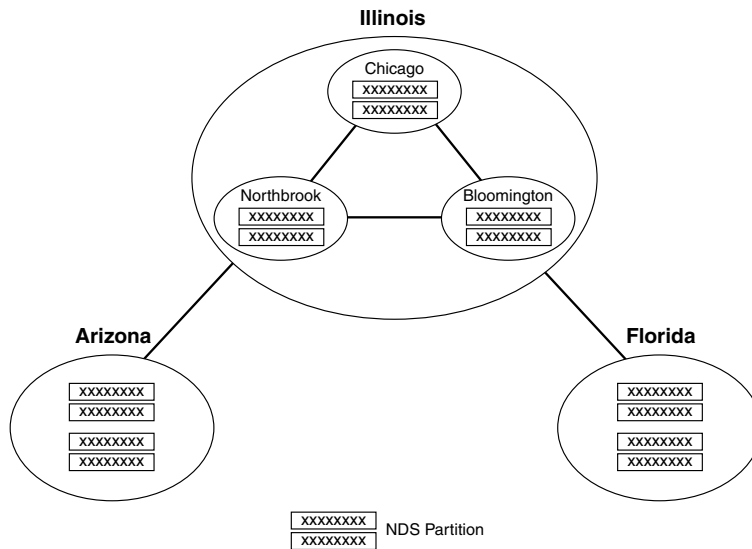


For these reasons, Microsoft expects that most companies will have fewer (but larger) domains when they deploy Windows 2000.

## NDS Partitions

NDS's partitioning technology takes an object storage approach to dividing the NDS namespace. Objects are the units of replication within NDS. The latest version of NDS (NDS8) supports indexed files, but does not recommend exceeding 1,500 objects in a given partition for performance reasons.

Figure 1.5 shows how NDS might be deployed based on the same customer scenario. In this scenario, each territory and site would require multiple partitions to store all the objects corresponding to that location.



**FIGURE 1.5**

*An NDS partition design.*

From Microsoft's perspective, NDS's partitioning scheme can be complex because the contents of container objects cannot span partitions.

## Catalogs

For both Active Directory and NDS, in some situations, storing all possible objects in a single directory partition is impractical. For this reason, both directory services provide a catalog feature.

## Active Directory Catalogs

Active Directory uses the Global Catalog (GC). The GC contains a subset of object attributes that are of interest beyond the scope of a single domain. You can assign the GC privilege to DCs, which then hold certain object attributes that are automatically replicated to other GC servers throughout the forest.

## NDS Catalogs

NDS provides a mechanism called Catalog Services to create catalogs. Administrators use the Catalog Service Manager utility to create and manage catalogs. With this utility, you can define the cataloged partitions and the associated object attributes. The Catalog Service Manager uses the Dredger to periodically rebuild the catalogs.

Microsoft believes that NDS catalogs have several significant weaknesses. Unlike ADS, NDS catalogs are not based on incremental updates. The Dredger starts with a blank, flat file and proceeds to individually query each partition. This process can be time-consuming and may risk dated information. Also, the “dredged” objects do not retain their access control properties.

## Internet Standards Support

In the directory services arena, Microsoft feels two Internet standards are very important: LDAP and DNS.

Although both directory services use LDAP, only NDS requires LDAP Services for Novell to be installed on every NDS server supporting LDAP access. Although some people may argue that Microsoft’s service tends to “manipulate” standards like TCP/IP and DNS, Novell Directory Service and LDAP use different naming syntaxes, which may complicate some Internet querying (because access control rights are computed differently when access to NDS occurs via LDAP as opposed to directly through NDS interfaces).

## Active Directory and DNS

Active Directory domain names are based on DNS names. Because domains maintain a one-to-one relationship with Active Directory partitions, an Active Directory namespace can be located directly via DNS. Further, an Active Directory object’s fully distinguished name contains the DNS name of its partition, is globally unique, and completely describes how to locate the object in a company’s intranet or across the entire Internet.

## NDS and DNS

The NDS object naming style does not use DNS names. Therefore, when you are locating an object in NDS, you need to know how to first find the appropriate NDS server. Also, because NDS and LDAP use different naming syntaxes, intranet applications written to access NDS directly use different object names than Internet applications that use LDAP.

## Summary

Active Directory is a directory service developed by Microsoft to provide information and services to enterprise users. The main features are manageability, security, and interoperability. Active Directory Service consists of both the logical and physical structure of the enterprise, and it attempts to consolidate administrative efforts with its array of new technical features.

It is clear that companies expect to use directory services in a growing number of roles. Further, companies want to eliminate corporate and geographic boundaries as barriers to doing business over the Internet, reverse the trend toward proliferation of role-specific directories, and realize synergy from elements in their network computing environments. Most important, companies need to make informed decisions now about selecting the directory services that will support their business needs well into the future. This chapter addressed some of Microsoft's issues with Novell's directory services (NDS) as well.

You will begin exploring more of the technical structure of Active Directory in the following chapter.



# Active Directory Architecture

CHAPTER

# 2

## IN THIS CHAPTER

- Subsystem Architecture 26
- Directory Service Architecture 28
- Protocols, Interfaces, and Services to Active Directory 31
- Logical Structure Fundamentals 33
- Physical Structure Fundamentals 44

Understanding the interactions of Active Directory's architectural components provides the basis for understanding how Active Directory manages data. The first part of this chapter addresses the relationship of Active Directory to the rest of the Microsoft Windows 2000 Server operating system. Later, we will look into the logical and physical structures of the Active Directory database, its components, and their characteristics.

## Subsystem Architecture

In Windows 2000, the two processor access modes are kernel and user. They protect applications from platform variations by dividing the low-level, platform-specific processes from the upper-level processes. They also prevent direct access to system code and data.

Applications and services run in user mode where they request system services through an application programming interface (API) that obtains limited access to system data. The process is transferred to kernel mode to perform its role in a protected environment. The process is then transferred back to user mode.

### NOTE

The Local Security Authority (LSA), part of the security subsystem in user mode, is the module where Active Directory runs.

The security reference monitor, which runs in kernel mode, enforces the security rules of the security subsystem. Access Control Lists (ACLs) protect objects in the Active Directory structure. Figure 2.1 shows the location of Active Directory within Windows 2000.

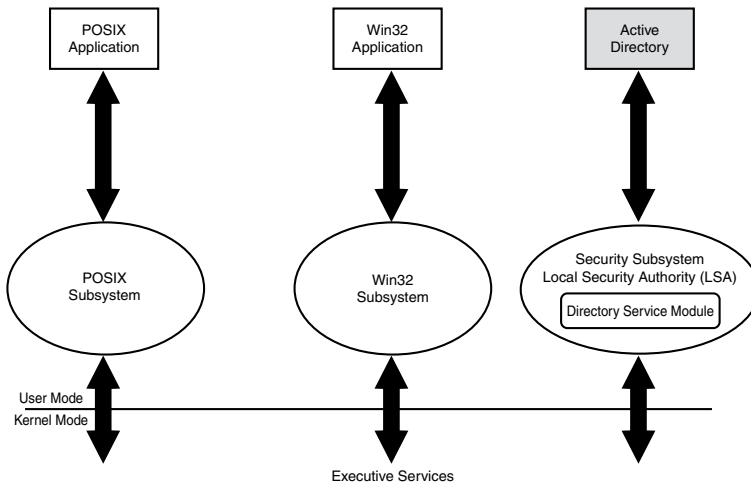
The integration of Active Directory and the security subsystem services is critical to the success of Windows 2000. First, all directory objects being accessed require authentication (performed by the security subsystem) and then validation of access permissions (performed by the security subsystem and security reference monitor). The security reference monitor, which sits in kernel mode, enforces the access control applied to Active Directory objects.

## Security Subsystem

As mentioned previously, Active Directory is a subcomponent of the Local Security Authority. Components of the security subsystem run in the context of the Lsass.exe process and include the following:

- Local Security Authority
- Net Logon service
- Security Accounts Manager service

- LSA Server service
- Secure Sockets Layer
- Kerberos V5 authentication protocol and NTLM authentication protocol

**FIGURE 2.1**

*Active Directory location within Windows 2000 system.*

The security subsystem monitors security policies and accounts in effect on the computer system.

## Local Security Authority (LSA)

The Local Security Authority (LSA) is a protected module that maintains the local security on a system (known as the local security policy).

In general, the LSA performs four primary functions:

- It manages local security policy.
- It provides interactive user logon services.
- It generates tokens containing user and group information about security privileges for the user.
- It manages the audit policy and settings, and it writes alerts to the appropriate system log file.

The local security policy identifies the following:

- Domains trusted to authenticate user logons
- Users who can access the system and their respective method (locally, remotely, or as a service)
- Users who are assigned privileges
- The level of security auditing
- Default memory quotas

The LSA has the following components:

- Netlogon.dll Net Logon service securely passes the user's logon credentials to a domain controller and returns the domain security identifiers (SIDs) and user rights. In Windows 2000, the Net Logon service uses DNS to locate domain controllers. For Windows NT 4.0 domain controllers, this service also acts as the replication protocol.
- Msv1\_0.dll NT Lan Manager (NTLM) authentication protocol authenticates clients that do not use Kerberos authentication.
- Schannel.dll Secure Sockets Layer (SSL) authentication protocol provides authentication over an encrypted channel.
- Kerberos.dll Kerberos V5 authentication protocol providing Windows 2000 authentication.
- Kdcsvc.dll Kerberos Key Distribution Center (KDC) service is responsible for granting tickets to clients.
- Lsasrv.dll LSA server service, which enforces security policies.
- Samsrv.dll Security Accounts Manager (SAM) stores local security accounts, enforces locally stored policies, and supports APIs.
- Ntdsa.dll Directory service module supports the Windows 2000 replication protocol and Lightweight Directory Access Protocol (LDAP), and manages partitions of data.
- Secur32.dll Multiple authentication provider that holds all the components together.

## Directory Service Architecture

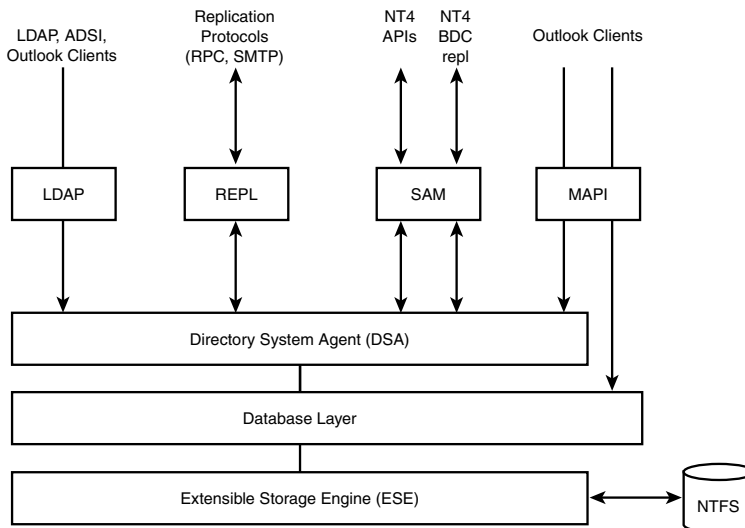
Active Directory functionality is a layered architecture where the layers represent server processes that provide directory services to client applications. Active Directory consists of three service layers and several interfaces and protocols that work together to provide directory services. The three service layers are



- The Directory System Agent (DSA)
- The Database Layer
- The Extensible Storage Engine (ESE)

These layers host the different information types required to locate records in the directory database. Above the service layers in this architecture are the protocols and APIs (APIs are on the clients only) that enable communication between clients and directory services, or between two directory services in the case of replication.

Figure 2.2 shows the Active Directory service layers and their respective interfaces and protocols. The direction of the arrows reveals how different clients gain access to Active Directory through the interfaces. LDAP and Messaging API (MAPI) clients gain access to the directory by calling functions, indicated by one-way arrows into the directory system agent. The SAM exists as a separate dynamic-link library (DLL) and can call only entry points exported by the directory system agent. All other components except the extensible storage engine (Esent.dll) are in Ntdsa.dll itself and are linked to the functions that they want to call. Because of this scenario, a three-way interaction is required between the three DLLs.



**FIGURE 2.2**

*Active Directory service layers and their corresponding interfaces.*

The key service components include the following:

- **Directory system agent (DSA)** This constructs a hierarchy from the domain structure stored in the directory, and provides APIs for directory access calls.

- **Database layer** This provides an abstraction layer between applications and the database. Calls from applications go through the database layer. They are never made directly to the database.
- **Extensible storage engine(ESE)** This engine communicates with records in the directory data store based on the object's relative distinguished name (RDN) attribute.
- **Data store (Ntds.dit)** This item is the Active Directory database and can be modified only by the ESE database engine. You can administer the file by using the Ntdsutil command-line tool (see Appendix A for more information on the ntdsutil.exe tool).

Clients obtain access to Active Directory by using one of the following interfaces supported by Active Directory:

- **LDAP/ADSI** Lightweight Directory Access Protocol (LDAP) and Active Directory Service Interfaces (ADSI).
- **MAPI** Messaging API used by Microsoft Outlook. Outlook clients connect to the DSA using the MAPI remote procedure call (RPC) Address Book provider interface.
- **SAM** Pre-Windows 2000 NT versions rely on the SAM interface to connect to the DSA. Mixed-mode backup domain controllers also use the SAM interface for replication.
- **REPL** Active Directory DSAs connect to each other by using a proprietary RPC interface during directory replication.
- **RPC** Remote Procedure Call used over TCP/IP to transmit and synchronize information of fast, reliable networks. Also acts as an interface allowing remote system processing.

## Directory System Agent

The directory system agent (DSA) is the server-side process that creates an instance of a directory service and provides access to the directory data. Clients use one of the supported interfaces to connect to the DSA to reference and manage Active Directory objects and their attributes.

The DSA layer provides the object identification and replication support as well as referrals and schema enforcement.

## Database Layer

The database layer provides an object view of Active Directory database information and prevents the upper layers of the directory service from directly accessing the underlying database system. The database layer is an internal interface. No database access is direct to the extensible storage engine. All calls and requests are routed through the database layer.

## Extensible Storage Engine

The Extensible Storage Engine (ESE) stores all the Active Directory data in the `ntds.dit` file, and is built on the ESE database from Microsoft Exchange. The Windows 2000 version of this database is `Esent.dll`. According to Microsoft, the ESE can manage a database up to 16TB, and has been tested to maintain 40 million objects per domain.

Active Directory comes with a predefined schema that defines all the attributes that are required and allowed for a given object. The ESE reserves storage only for space being used (attributes with values, not just all attributes), and it expands as attributes are added. The ESE can be backed up while online.

For more information on the data store, see the “Data Storage” section near the end of this chapter.

## Protocols, Interfaces, and Services to Active Directory

The Active Directory data model is derived from the X.500 model of objects and attributes (or properties). For example, attributes of a user object could include the user’s name, phone number, and email address. Note that Active Directory is not an X.500 directory. It does not implement the X.500 protocols—which include Directory Access Protocol (DAP), Directory System Protocol (DSP), Directory Information Shadowing Protocol (DISP), and Directory Operational Binding Management Protocol (DOP). LDAP provides the most important functions offered by DAP and is designed to work over TCP/IP without the overhead of “enveloping” OSI protocols over TCP/IP.

### X.500 Directory Service

The Active Directory information model is derived from the X.500 information model. X.500 defines several wire protocols that Active Directory does not implement. These protocols are

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

The Active Directory does not implement these protocols because there is little interest in them. This lack of interest stems from the fact that these protocols are dependent on OSI networking, an alternative to TCP/IP, which is not widely implemented (due to lack of transport efficiency). LDAP provides the most important functions offered by DAP and DSP and is designed to work over TCP/IP without the overhead.

## Lightweight Directory Access Protocol (LDAP)

In Active Directory, LDAP serves as both a protocol and an API. LDAP is the Active Directory core protocol, meaning that it is the only wire protocol supported by Active Directory. The LDAP API provides access to the LDAP protocol, and ADSI uses LDAP.

### NOTE

LDAP is a wire protocol, which means that it manages client and server encapsulation and requests transmissions.

LDAP was initially used with X.500 directories. LDAPv3 is an industry standard that can be used with any directory service that implements the LDAP protocol. Active Directory supports LDAPv2 and LDAPv3.

### NOTE

LDAPv3 is backward compatible with LDAPv2. A requirement of an LDAPv3 server is that an LDAPv2 client can connect to it.

## Active Directory Services Interface (ADSI)

The primary and recommended API for Active Directory is Active Directory Service Interfaces (ADSI). ADSI provides a simple, powerful, object-oriented interface to Active Directory and enables access to Active Directory by exposing objects stored in the directory as COM objects. ADSI is comprised of COM programming interfaces. A directory object is manipulated using the methods on one or more COM interfaces. By implementing the required interfaces, ADSI providers translate these interfaces to the API calls of a particular directory service.

ADSI makes it easy for programmers and administrators to create directory programs by using high-level tools such as Microsoft Visual Basic, Java, C, or Visual C++ without having to worry about the underlying differences between the different namespaces.

ADSI enables you to build or buy programs that give you a single point of access to multiple directories in your network environment, whether those directories are based on LDAP or another protocol. ADSI is fully scriptable for ease of use by administrators.

ADSI hides LDAP details from users. It is a simpler interface to the directory structure than the LDAP API.

## Active Directory Replication

Active Directory replication is performed over replication transport protocols. Active Directory handles replication differently depending on whether it is within a site (intrasite) or between sites (intersite). For intrasite replication, Active Directory replication uses RPC-over-IP transport protocols. For intersite replication (between sites), Active Directory replication uses your choice of two replication transport protocols: IP (RPC over IP) and Simple Mail Transfer Protocol (SMTP over IP).

### NOTE

RPC over IP is always used with one exception. The exception is for intersite replication traffic between domain controllers in different domains, and the administrator has opted to use SMTP.

For more information about Active Directory replication, see Chapter 8, “Managing Sites, Replication, and Network Traffic.”

## Logical Structure Fundamentals

In Active Directory, you organize resources in a logical structure. The logical structure allows you to define and group resources so that they can be located by name instead of by physical location. In previous versions of Exchange, resources were traditionally organized by “sites” and “servers,” based on convenience of administration. Active Directory resources include objects, organizational units, domains, trees, and forests.

## Domain Hierarchy

In Windows 2000, a domain defines both an administrative boundary and a security boundary for a collection of objects that are relevant to a specific group of users on a network. Administrative privileges do not extend from one domain to other domains, and a domain’s security policy applies only to security accounts within the domain.

Active Directory is made up of one or more domains. A *domain* is a security boundary of a Windows NT or Windows 2000 computer network, where privileges given in one domain do not carry over to other domains. All objects and organizational units exist within a domain. Therefore, a Windows 2000 domain, similar to a Windows NT 4.0 domain, may contain computers, user accounts, groups, and contacts.

The advantages of setting up your organization using domains include the following:

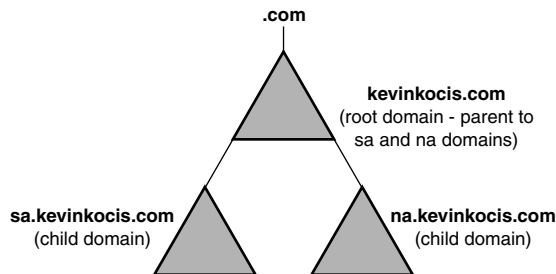
- The capability to set security policies (administrative rights and permissions) that don't cross domain boundaries.
- The capability to delegate administrative authority by domain or organizational unit to reduce the number of administrators with enterprise-level authority.
- The capability to store object information within a specific domain.

Domains are units of replication and can receive changes to the Active Directory and replicate them to all other domain controllers in the domain. All domain controllers host a writable copy of Active Directory.

#### NOTE

There are some drawbacks to enterprises with multiple domains. See Chapter 5, "Active Directory Security," for more information.

Domains can be structured hierarchically in parent-child relationships. As mentioned earlier, a parent domain is the domain directly superior in the hierarchy to one or more subordinate, or child, domains.



**FIGURE 2.3**

*Windows 2000 domain hierarchy.*

This hierarchical structure is different from the flat domain structure in previous NT versions. The Windows 2000 domain hierarchy allows you to search multiple domains in one query because each domain contains information about parent and child domains. This eliminates the need to know the exact location of an object to locate it. In previous versions of NT, you were required to know both the domain and the server where the object was located to find it. This proved to be inefficient in a large enterprise.

## Active Directory Domain Names

Active Directory uses DNS hierarchical naming standards for domains and computers. Domain and computer objects exist in both the DNS domain hierarchy and the Active Directory domain hierarchy.

### NOTE

Although these domain hierarchies have identical names, they represent separate namespaces.

## DNS Naming Conventions

Active Directory uses DNS naming standards to provide support for the mapping of DNS domain names to IP addresses.

Active Directory domain controllers are identified by their specific services, such as LDAP servers, domain controllers, and Global Catalog servers.

A DNS hierarchy is enforced by the following requirements:

- A child domain can have exactly one parent domain.
- Two children of the same parent cannot have the same name.

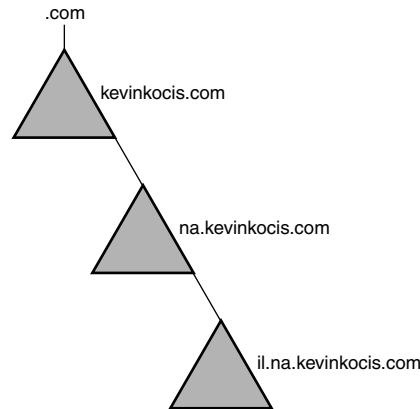
### NOTE

Because Active Directory domains use DNS names, these two standards also apply to Active Directory domains.

In the DNS naming convention, a period (.) separates each portion of a DNS name, as well as the domain name in the Active Directory's hierarchical tree structure.

For example, in the DNS domain name `il.na.kevinkocis.com`, “il,” “na,” “kevinkocis,” and “com” each correspond to a DNS domain. As illustrated in Figure 2.4, in Active Directory, the domain name `il.na.kevinkocis.com` represents a hierarchy in which `kevinkocis.com` is the root (topmost) domain, `na` is a child domain of `kevinkocis.com` (`na.kevinkocis.com`), and `il` is a child domain of `na.kevinkocis.com`.

The `.com` domain is outside Active Directory, although it appears as part of the domain name. Domains such as `.com`, `.org`, and `.edu`, are top-level domains used on the Internet to classify organizations by type.

**FIGURE 2.4**

*Active Directory hierarchy with DNS names.*

The hierarchy of domains is created as a result of contiguous naming, where each subordinate level appends to the preceding level.

Because every Windows 2000 domain has a DNS name (kevinkocis.com), and every Windows 2000–based computer has a DNS name (dc1.kevinkocis.com), domains and computers are represented both as objects in Active Directory and as nodes in DNS.

However, the Active Directory domain computer account object is in a different namespace from the DNS host record that represents the same computer in the DNS zone.

## Tree and Forest Structure

Multiple domains can be combined into structures called domain trees and forests. Active Directory domains are created in an inverted tree structure (similar to DNS), with the root at the top. Windows 2000 domain hierarchies are connected by two-way, transitive trusts.

### NOTE

The number of trust relationships that are required to connect domains is  $n-1$  where  $n$  is the total number of domains.

If domains in the same organization require different namespaces (for example, because of acquisition, merger, or strong business independence), create a separate tree for each namespace. These noncontiguous trees that are linked by trust relationships form a forest. A single tree with no relationships to other trees is a forest of one tree.



Windows 2000 tree structure relationships for the entire forest are stored in Active Directory as trust account objects in the System container within a specific domain directory partition. Information about a domain's connections to a parent domain is added to the configuration data that is replicated to every domain in the forest. This way, every domain controller in the forest knows the tree structure for the entire forest, including knowledge of the links between trees.

## Trees

A tree is a hierarchical arrangement of one or more Windows 2000 domains that share a common hierarchical naming structure. Endpoints on the tree are usually objects. Nodes in the tree (points at which the tree branches) are containers that hold a group of objects or other containers. A tree shows how objects are connected or the path from one object to another (refer to Figure 2.3).

Standard DNS domain names are used to represent the tree structure (for example, na.kevinkocis.com). The first domain in a domain tree is called the root domain (kevinkocis, in this example). Additional domains in the same domain tree are called child domains. Na.kevinkocis.com is a child domain to kevinkocis.com.

Active Directory is considered a namespace, and all domains sharing a common root domain form a contiguous namespace.

The Windows 2000 domain tree is the enterprise-wide Active Directory. All Windows 2000 domains in a given enterprise should belong to the enterprise domain tree. Enterprises that need to support non-contiguous DNS names for their domains will need to form a forest.

## Forests

A forest consists of one or more trees that do not form a contiguous namespace. Forests allow organizations to group divisions that operate independently but still need to communicate. Forests have a root domain, which is the first domain in a forest, and which is necessary for establishing trust relationships across the domain trees. A forest exists as a set of cross-reference objects and trust relationships known to the member trees. Domains in trees and forests also share a common schema and common configuration information. Therefore, an Exchange organization can span an entire forest, but cannot span multiple forests.

### NOTE

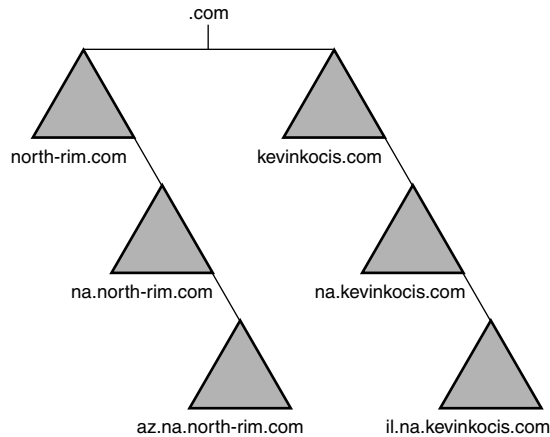
Forests will evolve mostly from corporate partnerships and mergers. This will depend on the current directory structure of each company.

You can create multiple forests and trust relationships between specific domains in the various forests. This lets you grant access to resources and accounts that are outside a particular forest. However, an Exchange infrastructure cannot span multiple forests.

### NOTE

Trust relationships created between domains in different forests are one-way and nontransitive by default.

Figure 2.5 illustrates forest relationships. Forests are at the top of the hierarchy made up of trees that contain domains. Domains are made up of other domains or organizational units.



**FIGURE 2.5**

*Active Directory forest structure.*

### Distinguished Name

The distinguished name is unique and only identifies a single object in the enterprise. LDAP uniquely identifies every object in the enterprise through a comma-separated list of name values. The full path to the object is defined by the distinguished name (also known as a DN).

### NOTE

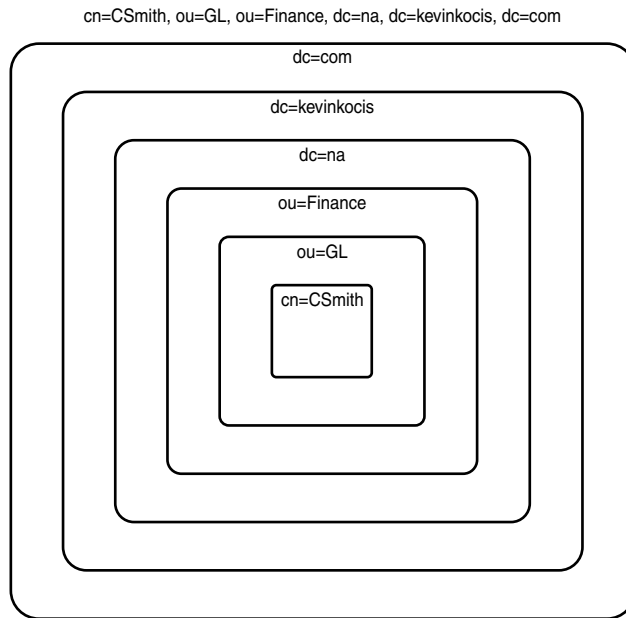
No two identical distinguished names can exist in Active Directory. Should two identical names populate the same group (for example, two users named Chris Smith work in the marketing department at the same location), you might need to alter your naming conventions for this particular situation.

The distinguished name contains information for an LDAP client to retrieve the object's information from the directory.

For example, a user named Chris Smith works in the finance department at my pseudo company. His user account is created in an organizational unit that stores the accounts for finance department employees working with the general ledger (we'll abbreviate general ledger as GL). Chris Smith's user identifier is CSmith, and he works in the North American branch of the company. The root domain of the company is kevinkocis.com, and the local domain is na.kevinkocis.com. The DN for this account would be as follows:

Cn=CSmith,ou=GL,ou=Finance,dc=na,dc=kevinkocis,dc=com

Figure 2.6 shows the layers determining the DN.



**FIGURE 2.6**

*The distinguished name for the Chris Smith user object.*

Note that each part of the DN is associated with an object class, a naming scheme adopted from LDAP. There are three important rules in terms of class attributes:

- DC is assigned to DNS components.
- OU is assigned to organizational units.
- CN is assigned to all other attributes.

The DN reads from most specific to most general, left to right. In the LDAP distinguished name, the relative distinguished names begin at the left and end at the right with the root name.

MMC snap-in tools for Active Directory do not display the LDAP abbreviations for the naming attributes domain component (dc=), organizational unit (ou=), or common name (cn=). These abbreviations are shown only to illustrate how LDAP recognizes the portions of the distinguished name. Most Active Directory tools display object names in canonical form (because distinguished names are difficult to remember), which we'll address later in this chapter.

#### NOTE

Each portion of the distinguished name is expressed as *attribute\_type=value* (for example, cn=CSmith).

### Relative Distinguished Name

The relative distinguished name (RDN) of an object is the part of the name that identifies this object as unique from other objects in its naming hierarchy. The name of the object itself (the CN attribute), which is separate from the object's path, is defined by the relative distinguished name.

In Figure 2.6 from the preceding section, the relative distinguished name of the object is CSmith. The maximum length allowed for a relative distinguished name is 255 characters, but the schema imposes more specific limits. For example, the cn attribute type, which is often used for naming the relative distinguished name, is restricted to a maximum of 64 characters.

Active Directory relative distinguished names are unique within a specific parent container. Active Directory does not allow two identical RDNs in the same parent container, but they can exist in different hierarchies.

#### NOTE

Two objects can have identical relative distinguished names but still be unique in the directory because within their respective parent containers, their distinguished names are not the same. In terms of LDAP, the object cn=CSmith,dc=na,dc=kevinkocis,dc=com is identified as being different from cn=CSmith,dc=kevinkocis,dc=com.

The relative distinguished name for each object is stored in the Active Directory database and contains a parent reference of the object.

Naming Attributes

Active Directory follows the attribute naming standards as proposed in Request For Comments (RFC) 2253, but does not implement all the standards. For example, Active Directory does not use certain nomenclature, which I'll address in a moment.

In Active Directory, the attribute type used to describe the object's relative distinguished name (in this case, cn=) is called the naming attribute. For example, part of the definition of the class User is the attribute cn (CommonName) as the naming attribute. For this reason, the relative distinguished name for user CSmith is expressed as cn=CSmith.

The naming attributes shown in Table 2.1 are used in Active Directory, as described in RFC 2253.

TABLE 2.1 Active Directory Naming Attributes (Default)

Object Class	Display Name	Naming Attribute LDAP Name
User	Common-Name	cn
Organizational Unit	Organizational-Unit-Name	ou
Domain	Domain-Component	dc

Other naming attributes described in RFC 2253 (o= for organization and c= for country/region) are not implemented in Active Directory, even though LDAP recognizes them.

You will use naming attributes (such as DN and RDN) only when you are programming for LDAP and using ADSI or other scripting or programming languages.

Object Identity and Uniqueness

Every Active Directory object has a unique identity, even if it is moved or renamed. The identity of an object is defined by a globally unique identifier (GUID), a 128-bit number assigned by the directory system agent (DSA) when the object is created. The GUID is permanently stored in an attribute, *objectGUID*, that is present on every object. The *objectGUID* attribute is secured and cannot be altered or deleted.

NOTE

The common pronunciation of GUID is gwid.

## Active Directory Name Formats

Active Directory supports several object name formats. These formats depend on the way the object was created. Active Directory displays name strings in the canonical name format illustrated in the following list. The following formats are supported by Active Directory and are based on the LDAP distinguished name:

- **LDAP Distinguished Name** LDAP v2 and LDAP v3 recognize the standard naming conventions, which represent cn=common name, ou=organizational unit, o=organization, c=country/region. Active Directory implements the domain component (dc) instead of o=organization and does not support c=country/region. For example  
cn=csmith,ou=gl,ou=finance,dc=na,dc=kevinkocis,dc=com
- **LDAP Uniform Resource Locator (URL)** Active Directory supports LDAP access from any LDAP-enabled client. An LDAP URL names the server holding Active Directory services and the attributed name of the object (DN). For example  
LDAP://server1.na.kevinkocis.com/cn=csmith,ou=gl,  
ou=finance,dc=na,dc=kevinkocis,dc=com
- **Active Directory Canonical Name** By default, display of object names in Windows 2000 user interfaces is the canonical name, which uses the DNS domain name (separated by periods). The respective canonical name of the previous example, would appear as follows:

na.kevinkocis.com/finance/gl/csmith

### CAUTION

If the name of an organizational unit contains a forward slash character (/) (for example, if the OU name was "finance/gl"), the system requires a special escape character in the form of a backslash (\). This is done to distinguish between forward slashes that separate parts of the canonical name and the forward slash that is part of the organizational unit name.

The canonical name that appears in Active Directory Users and Computers properties pages displays the escape character immediately preceding the forward slash in the name of the organizational unit. For example, if the name of an organizational unit is Finance/GL and the name of the domain is Kevinkocis.com, the canonical name is displayed as Kevinkocis.com/Finance\GL.

## DNS-to-LDAP Distinguished Name Mapping

Because DNS domain names mirror Active Directory domain names, there may be confusion over the actual namespace. Active Directory names have a different format, required by LDAP to identify directory objects. DNS domain names are therefore mapped to Active Directory domain names and vice versa.

Active Directory uses an algorithm to automatically assign an LDAP distinguished name for each DNS domain name. The algorithm provides a domain component (dc) attribute-type label for each DNS label in the DNS domain name. For example, the DNS domain na.kevinkocis.com is translated to the LDAP distinguished name that has the form dc=na,dc=kevinkocis,dc=com.

## Logon Names

Users gaining access to a domain and its resources require a unique logon name. User accounts are security principals (objects to which Windows security is applied in the form of authentication and authorization), and they are authenticated when they log on to the domain or local computer. They are authorized when they access resources.

User security principals have two types of logon names:

- SAM account name
- User principal name

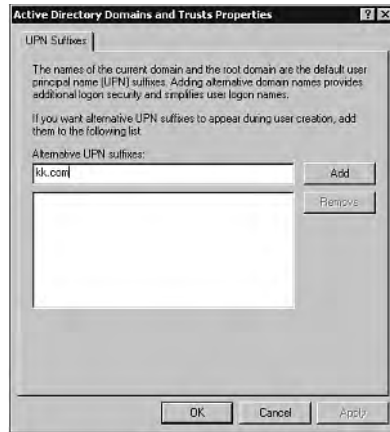
A SAM account name is required for compatibility with previous Windows NT domains. SAM account names are flat names compared to DNS hierarchical names.

A user principal name (UPN) is a “friendly” name that is shorter and easier to remember than the distinguished name. It consists of a shortened name that usually represents the user and the DNS name of the domain where the user object resides. For example, the user Chris Smith, who has a user account in the kevinkocis.com domain, might have the user principal name CSmith@kevinkocis.com. Because the user’s principal name is independent of the user’s distinguished name, a user object can be moved or renamed without affecting the user’s logon name.

### NOTE

The user principal name is an attribute (*userPrincipalName*) of the security principal object. If this attribute has no value, the default user principal name <userName>@<DnsDomainName>.

You can create additional user principal name suffixes and assign them if you don't want to use the default domain name (one example might be a very long DNS domain name). Chris Smith might want to use csmith@kk.com (instead of csmith@na.kevinkocis.com); see Figure 2.7. For more information on creating additional UPN suffixes, see Chapter 4, “Managing Users, Groups, and Computers.”



**FIGURE 2.7**

*Adding a UPN suffix in the Active Directory Domains and Trusts Properties window.*

## Physical Structure Fundamentals

Active Directory separates the logical structure of the domain hierarchy from the physical structure. By grouping resources logically, you can locate a resource by name rather than its physical location. Because you group resources logically, Active Directory allows the network's physical structure to be transparent to users.

The physical structure of Active Directory is based on sites. A site is a combination of one or more IP subnets, connected by high-speed connectivity.

### NOTE

Microsoft considers a high-speed link to be 10 million bits per second or faster. For administrators, a T1 link may be high-speed if its bandwidth is not saturated. The administrator will also need to decide whether additional sites need to be created to control traffic.



Active Directory uses replication to ensure that domain controller changes are updated to all the other domain controllers in the domain. Within a site, Active Directory generates a ring topology for replication among domain controllers in a domain. This ensures at least two replication paths from one domain controller to another. Replication can still take place even if a domain controller is down or offline. If you add or remove a domain controller from the network or a site, Active Directory automatically reconfigures the topology to reflect the change.

For more information, see Chapter 8.

## Directory Contents

Active Directory contains a variety of objects structured with the use of containers (in the form of organizational units—OUs). The directory is also broken into partitions known as directory partitions or Naming Contexts.

### Objects

An object is the basic unit of the Active Directory. It is a distinct, named set of attributes that represents something concrete, such as a user, printer, computer, or application. Attributes are the characteristics of the object identified in the directory. Object attributes include information such as its location and features. A user is considered an object. In Exchange, a user's attributes can include its first name (Chris), last name (Smith), and email address (csmith@kevinkocis.com), as well as the user's ability to receive email, which types of email he can receive, and where he can receive email. Objects are assigned permissions for access and can be collected and organized in items called containers.

### Object Naming

All objects in Active Directory follow naming conventions. There are various naming conventions in Active Directory. Although DNS is an effective tool for Internet name resolution, it does not accommodate the granularity required for Active Directory naming. LDAP is more adequate because it can uniquely identify objects in Active Directory.

### Containers

A container is any item in the directory tree where objects are added. An obvious example of a container is a folder. However, you can also use MMC to add tools to items other than folders, which then also become containers.

The predominant container in Active Directory is the Organizational Unit.

### Organizational Units (OUs)

An organizational unit is an Active Directory container used for storing objects. An organizational unit is the smallest scope or unit to which you can assign or delegate administrative authority. You can use organizational units to create containers within a domain that represent the hierarchical, the logical, or the departmental (business) structures within your organization.

**NOTE**

Many Windows NT 4.0 domains migrating to Windows 2000 Active Directory can be converted to OUs in their new domain model and have delegated authority.

Organizational units can contain other organizational units, which means you can create a hierarchy of containers that can be extended as necessary to model your organization's hierarchy within a domain. OUs are also the smallest unit of group policy object application available in Active Directory.

The three main reasons for using OUs are

- Organizing accounts
- Delegation of authority
- Group policy application

**NOTE**

An organizational unit cannot contain objects from other domains.

## Directory Partitions

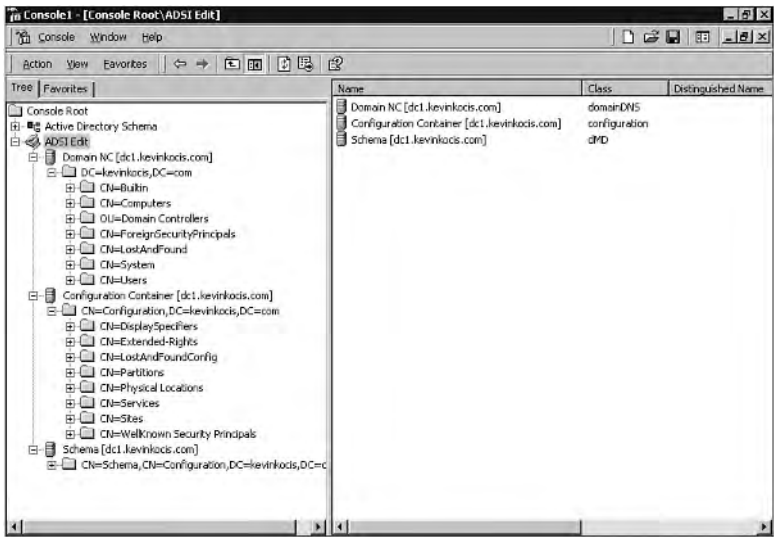
In Active Directory, a forest is partitioned into domains. Domain controllers within the same domain share the same information. Domain controllers from different domains share the same configuration, schema, and GC information, but do not share the same domain data. The directory partition, also called a naming context, allows for storage distribution. This enhances the scalability of the Active Directory database to millions of objects.

Each directory partition contains a subdirectory of objects in the tree. Copies of this partition can be stored across domain controllers, which are updated through directory replication.

The information stored in Active Directory on every domain controller (whether or not it is a global catalog server) is partitioned into three categories: domain, schema, and configuration data. These directory partitions are the units of replication. The three directory partitions hosted on each domain controller are as follows:

- Configuration partition Contains site and replication topology information, as well as service and directory partition information. This data is common to all domains in the domain tree or forest. Configuration data is replicated to all domain controllers in the forest.

- **Schema partition** Contains all object types (and their attributes) that can be created in Active Directory. This data is common to all domains in the domain tree or forest. It stores class and attribute definitions for all existing and possible Active Directory objects. Schema partition data is replicated to all domain controllers in the forest.
- **Domain NC partition** Contains all the objects in the directory for this domain. Updates to this container are replicated to only domain controllers within the domain and to GC servers if the update is made to an attribute configured for replication to the Global Catalog. See Figure 2.8 for the ADSI view of domain partitions.



**FIGURE 2.8**  
*The domain partitions as viewed in ADSI Edit.*

**Using ADSI Edit**

To use ADSI Edit, install the Support Tools located in the Support\Tools folder on the Windows 2000 Server operating system CD. Double-click the Setup icon in that folder.

To view or change attribute values by using ADSI Edit, follow these steps:

1. Go to Start, Programs, Windows 2000 Support Tools, Tools, ADSI Edit.
2. If the directory partition whose attributes you want to change or view is not displayed, right-click the ADSI Edit icon and then click Connect to.
3. If the current computer is not the domain controller on which you want to change attributes, under Computer, click Select or type a domain controller, and then select or enter the computer name.

4. To select the directory partition, under Connection Point, click Naming Context.
5. In the Naming Context list, click a directory partition and then click OK.

**NOTE**

In the Name box, the name of the directory partition that you selected is displayed. You can replace this name with a name that better identifies the specific connection.

6. Select the object whose property values you want to view or change.
7. In the Properties dialog box, in the Select Which Properties to View box, click one of these alternatives: Optional, Mandatory, or Both.
8. In the Select a Property to View box, click the property that you want to view.
9. To change a property value, type the value in the Edit Attribute box.
10. Click Set, and then click OK.

If the domain controller is a global catalog server, it also holds a fourth category of information, partial replica of domain data directory partition.

**Partial Replica of Domain Data Directory Partition for All Domains**

A global catalog server stores and replicates a partial replica of the domain directory partition for all other domains in the forest. This partial replica contains a subset of the properties for all objects in all domains in the forest, and it is read-only. A complete replica is read/write.

Each domain is mapped to a different directory partition so objects belonging to two different domains can be maintained and replicated independently. Information relevant to the entire forest is replicated separately.

**NOTE**

You cannot rename the root object in a directory partition, which means that you cannot rename a Domain, Schema, or Configuration container.

**Configuring Directory Partitions**

The Active Directory Installation Wizard copies the directory database file (Ntds.dit) from its location in the %SystemRoot\System32 directory to the destination that you have specified, after which the wizard configures the local server to host the directory service. This process includes creating the directory partitions and the default domain security principals.

The following directory partitions are created as default partitions on the first domain controller in a forest and are updated through replication on every subsequent domain controller created in the forest:

- The schema directory partition is created as `cn=schema,cn=configuration,dc=forestRootDomain`. `Schema.ini` is used to create default directory objects and display specifiers and to implement default security on the directory database.
- The configuration directory partition is created as `cn=configuration,dc=forestRootDomain`.
- The domain directory partition is created as `dc=domainName` and contains the security principals for the domain.
- When you create a new domain, the wizard creates a new directory partition that contains all the default domain objects.
- When you create an additional domain controller in an existing domain, the objects are updated through replication. The wizard does not create the default domain directory partition objects.
- When you upgrade a primary domain controller in Windows NT 4.0, the wizard creates domain security principals and local security principals. It also migrates LSA memberships and existing accounts.

## Configuration Partition

The configuration directory partition is created when the first Windows 2000 domain is. On future creations of child domains or new tree-root domains or when an additional domain controller is added to the domain, the configuration directory partition is replicated to the new domain controller.

The following objects are child containers within the Configuration container:

- DisplaySpecifiers
- Extended-Rights
- LostAndFoundConfig
- Partitions
- Physical Locations
- Sites
- Services
- Well-Known Security Principals

Although other information can be stored in the Configuration container, it is recommended that the following criteria apply to this data:

- The information is defined as global interest (for example, the default configuration and policy information for all instances of a given service in the enterprise).
- The information is highly available, such that referencing the information stored in another domain is not sufficient.
- The volatility of the information is low.
- The volume of information is small.

Global information should be stored in one of two places: in a child of the Services container or in a child of a site object.

### Managing Configuration Data

You can manage different portions of the Configuration container with Windows 2000 administrative tools. The following tools are available on the Start, Programs, Administrative Tools (which are also available as snap-ins in the MMC):

- Active Directory Sites and Services

#### NOTE

The Services container in Active Directory Sites and Services is hidden by default. To reveal the Services container, right-click Active Directory Sites and Services, point to View, and then click Show Services Node.

- Active Directory Domains and Trusts
- Active Directory Schema

#### NOTE

The schema snap-in requires a special installation. See Chapter 9, "Managing Updates with Flexible Single-Master Operations," for details on installing the schema container.

### Schema Directory Partition

The schema for Active Directory consists of a set of object classes, attributes, and syntaxes. The schema sets rules for consistency regarding object creation and modification. Although Active Directory contains a default set of classes and attributes you cannot modify, you can extend the schema by adding new attributes and classes from a qualified domain controller

(the schema master FSMO). These changes must be targeted at the domain controller that holds the schema master role for the forest.

For more information about enabling schema modifications and extending the schema, see Chapter 7, “Managing and Modifying Active Directory Schema.” For more information about single-master roles, see Chapter 9.

You can also use ADSI Edit to view the schema directory partition objects and properties. When you open ADSI Edit, the Schema container is displayed by default. Expand the container to view the attributes and classes.

## Domain Directory Partitions

When you create a new domain, a domain directory partition is created in Active Directory.

The root object in each domain directory partition is a container object that is named for the DNS domain. The child containers of the domain container can be viewed in the Active Directory Users and Computers console.

A domain container has the following child containers:

- Builtin
- Computers
- Deleted Objects
- Domain Controllers
- ForeignSecurityPrincipals
- Infrastructure
- LostAndFound (Advanced Features)
- System (Advanced Features)
- Users

By default, only some containers appear in the Active Directory Users and Computers. To view all the containers in Active Directory Users and Computers, click on the View menu and select Advanced Features.

### NOTE

Unlike the configuration and schema directory partitions, a full copy of the domain directory partition is replicated only among domain controllers within the same domain, not to other domains in the forest. A partial copy of domain objects (all objects, but a limited set of attributes that have been configured to replicate to the global catalog) is also replicated to all domain controllers that are configured to be Global Catalog servers.

You can use Active Directory Users and Computers to manage the contents of the domain directory partition. You can use ADSI Edit to manage properties not displayed in Active Directory Users and Computers. When you open ADSI Edit, the domain directory partition for the domain to which you are logged on is displayed by default.

### System Container Contents

The System container (located in the domain partition) stores per-domain operational information, such as local security policy, file link tracking, network meetings, objects representing other trusted domains, and containers for RPC and Winsock connection points.

The System container has the following child containers:

- AdminSDHolder
- Default Domain Policy
- Dfs Configuration
- File Replication Service
- FileLinks
- IP Security
- Meetings
- MicrosoftDNS
- Policies

### CAUTION

It is highly recommended that you do not alter or modify the Policies container. Instead, use the Group Policy MMC snap-in to specify a desktop configuration for a particular Group Policy object.

- RpcServices
- WinsockServices

During the installation of Windows 2000 Server, the default Active Directory database file (Ntds.dit) is placed in the %SystemRoot%\System32 directory. In this location, the file does not function as the directory database; it exists as a distribution copy so that you do not have to use the operating system CD to install Active Directory.

Ntds.dit includes the default copy of the schema and configuration directory partitions, as well as a default domain directory partition. During the installation of Active Directory, the default copy of the schema and configuration directory partitions (along with the domain directory



partition if the domain controller is an additional domain controller in the domain) are synchronized with existing domain controllers for that domain. At the completion of the installation process, Active Directory is fully synchronized and available for updates on the new server.

During the installation of Active Directory, you can stop the replication process. To stop the replication process, click the Finish Replication Later button when it appears. Replication then continues after the computer is restarted. The domain controller does not advertise itself until replication is complete.

### NOTE

If the AD database to be replicated is small in size, the Finish Replication Later button may appear for only a brief period of time or not at all.

## 2

## Domain Controllers

Active Directory must reside on a domain controller, which stores a complete copy of all Active Directory information for that domain. It also manages changes to that information, and replicates those changes to other domain controllers in the same and other domains. Schema and infrastructure information is replicated between all domain controllers in a forest.

## Member Server

Although only domain controllers contain Active Directory objects, other Windows 2000 servers can add functionality to your Windows 2000 implementation.

A member server is a Windows 2000 server that is a member of an Active Directory domain but is not a domain controller and does not contain any Active Directory objects. Member servers share common security features such as domain policies and user rights.

Member servers can act as the following:

- File servers
- Print servers
- Web servers
- Proxy servers
- Routing and Remote Access Services (RRAS) Servers
- Application servers, which include component servers, terminal servers, certificate servers, database servers, and email servers

Because it is not a domain controller, a member server does not handle the account logon process, participate in Active Directory replication, or store domain security policy information.

These member servers have a common set of security-related features:

- Member servers adhere to Group Policy settings defined for the site, domain, or organizational unit.
- Resources available on a member server are configured for access control.
- Member server users have user rights assigned to them.
- Member servers contain a local security account database, the Security Account Manager (SAM).

## Changing Roles

A server within an Active Directory domain can function in one of two roles: either as a domain controller or a member server.

As the needs of your computing environment change, you might want to change the role of a server. Using the Active Directory Installation Wizard or `dcpromo.exe`, you can promote a member server to a domain controller, or you can demote a domain controller to a member server (as mentioned earlier).

## Sites

A Windows 2000 site is a group of Active Directory servers that can communicate over highly reliable, high-bandwidth, permanent and synchronous connections. Setting up Windows 2000 sites allows you to configure Active Directory access and a replication topology to take advantage of the physical network. When a user logs on, the Active Directory client finds Active Directory servers in the same site as the user. Because computers in the same site are proximal in network terms, communication among computers is reliable, fast, and efficient.

The two primary reasons for creating sites in Active Directory are to control replication traffic and to control logon and authentication traffic. You create sites with Active Directory Sites and Services. No direct relationship exists between domains and sites, so a single domain can span multiple sites, and a site can span multiple domains. Typically, a site has the same boundaries as a local area network.

One of the benefits of Active Directory is that domains can span physical locations with different topologies connected by WAN links and still remain transparent to the user. However, available WAN bandwidth is always a consideration.

## Sites Versus Domains

It is important to understand that sites are independent of domains. Sites map the physical structure of your network, whereas domains (if you use more than one) typically map the logical structure of your organization. Logical and physical structures are independent of each other, which has the following consequences:

- There is no necessary connection between sites and domain namespaces.
- There is no necessary correlation between your network's physical structure and its domain structure. However, in many organizations, domains are set up to reflect physical network structure. This is because domains are partitions, and partitioning influences replication—partitioning the forest into multiple, smaller domains can reduce the amount of replication traffic.
- Active Directory lets multiple domains appear in a single site and a single domain appear in multiple sites.

## Data Storage

Active Directory data is stored in the Ntds.dit ESE database file. Two copies of Ntds.dit are present in separate locations on a given domain controller:

- `%SystemRoot%\NTDS\Ntds.dit` Stores the database which contains domain information and a forest data replica.
- `%SystemRoot%\System32\Ntds.dit` A distribution copy of the default directory that is used during promotion of a Windows 2000–based computer to a domain controller. During the promotion process, the Ntds.dit file is copied from the `%SystemRoot%\System32` directory into the default `%SystemRoot%\NTDS` directory.

Active Directory stores data for an entire forest. “Directory” and “forest” can be considered synonymous. Although there is a single directory, data storage is distributed among one or more domains while consistent data is maintained throughout the forest that applies to all domains. Active Directory is partitioned and replicated. So that it can support tens of millions of objects, Active Directory is partitioned into logical segments. To provide support and availability for thousands of clients, each logical partition replicates its changes separately among those domain controllers in the forest that store copies of the same directory partitions.

Some directory partitions store forest-wide configuration information and schema information; other directory partitions store information specific to individual domains, such as users, groups, and organizational units. The directory partitions that store domain information are replicated to domain controllers in that domain only. The directory partitions that store configuration and schema information are replicated to domain controllers in all domains. Domain controllers configured as Global Catalog servers store a full replica of one domain directory

partition plus a partial replica of every other domain in the forest. A Global Catalog domain controller can be queried to find any object in the forest.

### NOTE

There is a distinction between a directory partition and a database partition. The Active Directory database is not partitioned. Only the directory tree, which is the logical representation of the data held by a domain controller, is partitioned.

The distribution of Active Directory data in the directory tree can be summarized as follows:

- Domain-wide Data Distribution
  - Domain-specific data is stored in a domain directory partition.
  - A full, writable replica of the domain directory partition is replicated to every domain controller in the domain, including any Global Catalog servers in the domain.
- Forest-wide Data Distribution
  - Forest-wide data is stored in two directory partitions: the configuration directory partition and the schema directory partition. The Configuration container is the topmost object of the configuration directory partition; the Schema container is the topmost object of the schema directory partition.
  - Full, writable replicas of the configuration directory partition and the schema directory partition are replicated to every domain controller in the forest.
  - In addition to a full, writable replica of a single domain (the domain for which the domain controller is authoritative), partial, read-only replicas of every other domain directory partition in the forest are stored on domain controllers designated as Global Catalog servers. The read-only replicas in the Global Catalog are “partial” because they store only some of the attributes for each object.

When Active Directory is first installed on a computer running Windows 2000 Server, the entire full replicas or partial replicas are replicated to create the directory. Thereafter, only changes to directory objects (attribute changes and the creation and deletion of objects) are replicated.

## Summary

Active Directory architecture, although too complex for this book, has been enhanced in many ways from previous Windows NT versions. The logical structure of Active Directory mirrors

DNS, and shares many similarities and conforms to its standards. Active Directory is composed logically of objects, domains, and trees. Physically, it is comprised of domain controllers and sites.



# Managing Domains, Trusts, and DNS

CHAPTER

3

## IN THIS CHAPTER

- Domain Fundamentals 60
- Managing Trusts 64
- Name Resolution in Active Directory 68
- Integrating DNS and Active Directory 80
- Heterogeneous Environments 87
- DNS and WINS 97
- DHCP in Active Directory 98

One of the most improved and enhanced features of Windows 2000 is the administrative overhead reduction in terms of domains, trusts, and DNS. This chapter focuses on those benefits as well as step-by-step examples and exercises.

## Domain Fundamentals

Windows 2000 domains and Active Directory depend on one another and even are defined by each other's characteristics. Let's start with an explanation of the Windows 2000 domain model and examine why that model is so different from the Windows NT domain model.

As you'll recall, Windows NT 4.0 domains didn't scale well. Using one-way non-transitive trusts in enterprise implementations required significant administrative overhead. Windows 2000 has a new approach to trusts and is now in coordination with industry standards such as the Lightweight Directory Access Protocol (LDAP) and Domain Name Service (DNS).

As you learned in Chapter 1, "Understanding Active Directory," Windows 2000 domains are organized in a hierarchy (including trees and forests), as opposed to manually trusted, non-contiguous domain namespaces. The first domain created in a Windows 2000 deployment is called the root domain. This domain serves as the root for all domain trees created in the forest. Each domain tree has its respective subroot. Because Windows 2000 domain structures share a direct relationship with DNS domain hierarchies, the structure of Windows 2000 domains is similar to the familiar structure of DNS domain hierarchies. Examples of root domains are `kevinkocis.com` or `north-rim.com`. They serve as the roots of their DNS hierarchies and roots of their respective Windows 2000 domain structure.

Domains subsequently created in a given Windows 2000 domain hierarchy become child domains of the root domain. For example, if `sales` is a child domain of `north-rim.com`, the `sales` domain becomes `sales.north-rim.com`.

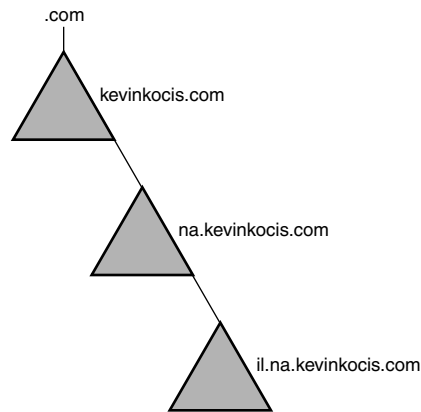
Windows 2000 requires that domains be either a root domain or a child domain in a domain hierarchy, and must be unique in respect to their parent domain. Refer to Figure 2.3 in Chapter 2, "Active Directory Architecture," for a visual guide.

### NOTE

You cannot have two domains called `sales` that are direct child domains of a root domain called `north-rim.com`, for example. However, you can have two domains called `sales` in the overall domain hierarchy. You could have `sales.north-rim.com` as well as `sales.az.north-rim.com`.



Because the Windows 2000 domain is an administrative boundary, administrative privileges do not flow across domain boundaries or down through a Windows 2000 domain tree. For example, in Figure 3.1, kevinkocis.com is the root domain and the parent domain of na.kevinkocis.com, and na.kevinkocis.com acts as the parent domain of il.na.kevinkocis.com. Users with administrative rights in domain kevinkocis.com do not have administrative rights in na.kevinkocis.com, nor do users with administrative rights in na.kevinkocis.com have administrative rights in domain il.na.kevinkocis.com. Only Enterprise Administrators, located in the root domain, had administrative powers over the entire forest. They can delegate authority to domain administrators to have administrative authority over a remote domain.



**FIGURE 3.1**

*A domain tree.*

## Managing Domains

The Microsoft Windows 2000 domain structure and its associated objects have changed significantly from their Windows NT 4.0 incarnations. Two significant changes are the domain scalability and the transition to a two-way transitive trust relationship mode.

Windows 2000 is more scalable than Windows NT 4.0. Windows NT 4.0 has a limit of 40,000 user accounts in a single domain. This limit comes from the maximum recommended size of the Security Accounts Manager (SAM) database file of 40 MB. Based on this, organizations were forced to create additional account domains to be able to support the number of user accounts expected in the organization. Another reason for creating additional domains was for administrative delegation. Because the domain is the most granular administrative unit, creating additional domains was one way to delegate administrative roles. Business groups also forced enterprises to allow them to create additional domains that were—outside of political realms—unnecessary.

With a Windows 2000 Active Directory implementation, organizations should avoid creating additional domains. Windows 2000 supports significantly larger numbers of objects in its database. Within a domain, organizational units are used to create very granular administrative roles. Therefore, very large organizations do not need to create additional domains to support their large user account requirements. You can implement a strong organizational unit (OU) model as opposed to multiple domains.

The main recommendation for planning domains and DNS is to delegate a separate DNS zone per each Active Directory domain—in other words, mirroring the AD structure. You should have two DNS servers running on domain controllers in the domain. Remember that when a domain is implemented, you cannot change its name or split it into two domains. You also cannot combine two domains. However, you can use an import/export tool called `ldifde.exe` to transport objects outside the forest. For object transfers within the forest but between domains, use the `movetree.exe` tool. Both tools are covered in more detail in Appendix A.

In Windows NT 4.0, all trusts were configured via one-way, non-transitive trusts. To establish a two-way trust, administrators from the two domains were required to coordinate two separate one-way trusts. These trusts were also non-transitive.

In Windows 2000, all trusts are two-way, transitive trusts. The exceptions are explained in Chapters 1 and 2. This enhancement eliminates administrative overhead in terms of trust configuration and management.

## Adding Domains

Basing domain creation on stable criteria such as geography is the best way to ensure migration will be as simple as possible. Other criteria, such as business groups or suborganizations, are much less stable and are likely to result in the need to move security principal accounts between domains.

You can migrate Windows NT 4.0 domains to Windows 2000 domains in one of the following ways:

- Create a new Windows 2000 domain and join an existing tree.
- Create a new Windows 2000 domain and create a new tree.
- Merge into an existing Windows 2000 domain.

### NOTE

You cannot move the security principals from a Windows NT 4.0 domain into more than one Windows 2000 domain at upgrade time. You need to use the Active Directory Services Interface (ADSI) and other tools to move the user accounts between domains after the upgrade is completed.

## Domain Models

The two domain modes are mixed mode and native mode. Mixed mode is the default mode setting for domains on Windows 2000 domain controllers. Mixed mode allows Windows 2000 domain controllers and Windows NT backup domain controllers to cohabitate in a domain. Mixed mode does not support the universal and nested group enhancements of Windows 2000.

You can change the domain mode setting to Windows 2000 native mode only after all Windows NT domain controllers are either removed from the domain or upgraded to Windows 2000. After that, if you do not plan to add any more down-level domain controllers to the domain, you can switch the domain from mixed mode to native mode. Also, native mode does not support down-level replication.

Several things happen during the conversion from mixed mode to native mode:

- Support for down-level replication and down-level domain controllers stops.
- You can no longer add new down-level domain controllers to the domain.
- All domain controllers are equal, even the Primary Domain Controller (PDC) during the migration process.

### CAUTION

Do not change the domain mode if you have or will have any Windows NT 4.0 domain controllers! The change from mixed mode to native mode is one way only. You cannot change from native mode to mixed mode.

## Changing Domain Mode

To switch the domain mode, perform the following:

1. Start the Active Directory Users and Computers snap-in or the Active Directory Domains and Trusts snap-in.
2. Right-click the domain name, and then click Properties.
3. On the General tab, click Change To Native Mode.
4. In the Warning dialog box, click Yes and then click OK.

### NOTE

It may take up to 15 minutes for a domain mode change to impact all Windows 2000 domain controllers.

## Managing Trusts

As you learned earlier, one of the most important differences between Windows NT 4.0 domains and Windows 2000 domains is the way trust relationships are created and maintained between domains within the organization. Rather than establish a web of one-way trusts as required in Windows NT 4.0, Windows 2000 implements transitive trusts that span the domain tree and forest structure. This model greatly simplifies administration.

### Trust Relationships

Trust relationships in Windows NT 4.0 can be represented in the following equation (with  $n$  equaling the number of domains):

$$\text{Windows NT 4.0 domains} = (n * (n-1))$$

Therefore a company with 6 domains needs to establish 30 trust relationships ( $6 * (6-1)$ ).

Trust relationships among Windows 2000 domains can be represented in the following equation:

$$\text{Windows 2000 domains} = (n-1)$$

Therefore, a company with 6 domains needs to establish 5 trust relationships ( $6-1$ ).

That's a significant difference in the number of trust relationships that must be managed, particularly when you're in a corporation with hundreds of NT 4.0 domains!

Another trust feature of Windows 2000 domains is that they are created and implemented by default. As you install domain controllers, trusts are automatically created. This process is tied to the fact that Windows 2000 domains are hierarchically created. That enables Windows 2000 to automatically know which domains are included in a given domain tree, and when trust relationships are established between root domains, to automatically know which domain trees are included in the forest.

In contrast, administrators had to create (and subsequently manage) trust relationships between Windows NT domains, and they had to remember which way the trust relationships flowed (and how that affected user rights and permissions in either domain). The difference is significant, the management overhead is sliced to a fraction, and the implementation of such trusts is more intuitive—all due to the new trust model and the hierarchical approach to domains and domain trees.

Windows 2000 incorporates three types of trust relationships. The trust relationships available to Windows 2000 domains are the following:

- One-way trusts
- Transitive trusts
- Cross-link trusts

## One-Way Trusts

One-way trusts are obviously not two-way, nor are they transitive. You can still create one-way trusts just like in a Windows NT 4.0 environment. However, creating multiple one-way trusts does not create a transitive trust.

One-way trusts can be used when creating trust relationships with Windows NT 4.0 domains.

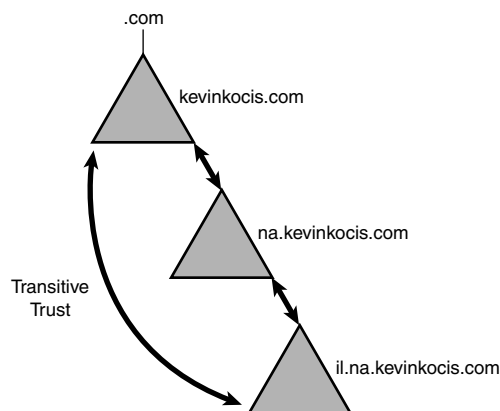
### NOTE

Because down-level domains cannot participate in Windows 2000 transitive trust environments, you must create one-way trusts for interoperability with down-level Windows NT domains.

You can also implement one-way trust relationships between domains in different Windows 2000 forests. This capability allows you to isolate the trust relationship to the domain where the relationship is created and maintained rather than create a trust relationship that affects the entire forest. These one-way trusts are called explicit trusts.

## Transitive Trusts

Transitive trusts establish a trust relationship between two domains that is able to flow through to other domains. If you assume that domain A trusts domain B, and domain B trusts domain C, then domain A inherently trusts domain C and vice versa. Let's look at the Windows 2000 domain example in Figure 3.2.



**FIGURE 3.2**

*Transitive trusts in a Windows 2000 domain tree.*

In this example, kevinkocis.com trusts na.kevinkocis.com, and na.kevinkocis.com trusts il.na.kevinkocis.com. Therefore, kevinkocis.com trusts il.na.kevinkocis.com.

Transitive trusts reduce the administrative overhead traditionally associated with the domain trust maintenance. In Windows 2000, transitive trust relationships between parent and child domains are automatically established whenever new domains are created in the domain tree.

#### NOTE

Transitive trusts are limited to Windows 2000 domains and to domains within the same domain tree or forest. You cannot create a transitive trust relationship with Windows NT 4.0 domains or between two Windows 2000 domains from different forests.

## Cross-Link Trusts (Shortcut Trusts)

Cross-link trusts (or shortcut trusts, as they are sometimes referred to) can increase authentication performance by establishing one-way transitive trusts between two domains. With cross-link trusts, a virtual link is created within the tree or forest hierarchy, enabling faster trust relationship confirmations.

Cross-link trusts are established between nonadjacent domains that are logically distant from each other in a forest or domain tree. You should implement cross-link trusts only if your network is experiencing heavy authentication traffic along the path between the domains. In Figure 3.3, if users in the bz domain of the tree are continually accessing resources in the il domain in the other branch, the authentication traffic can affect network and authentication performance.

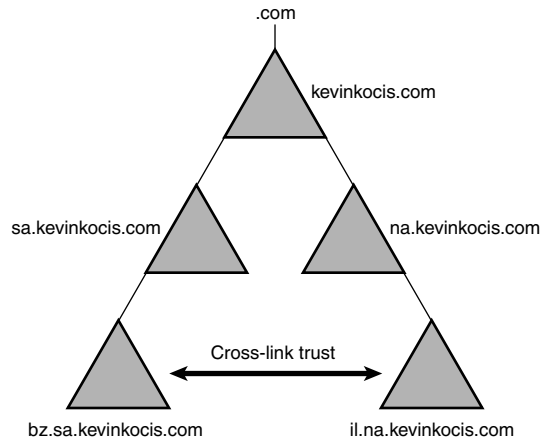
A better approach is to create a cross-link trust between domains bz and il, which enables authentications between the domains to occur without traversing the domain tree back to the root and down the other branch. The result is better performance in terms of authentication and less traffic to domains and DCs not directly involved in the process.

## Adding Trusts

Two-way transitive trusts are created by default when additional Windows 2000 domains are added to the tree or forest. In the case of down-level domains, explicit trusts must be created.

To create an explicit domain trust, do the following:

1. Open Active Directory Domains and Trusts.
2. In the console tree, right-click the domain node for the domain you want to administer, and then click Properties.

**FIGURE 3.3**

*A cross-link trust.*

3. Click the Trusts tab.
4. Depending on your requirements, click either Domains Trusted By This Domain or Domains That Trust This Domain, and then click Add.
5. If the domain to be added is a Windows 2000 domain, type the full DNS name of the domain.  
Or, if the domain is running an earlier version of Windows NT, type the domain name.
6. Optionally, you can type and confirm the password for this trust.
7. Repeat this procedure on the domain that forms the other part of the explicit trust relationship.

**NOTE**

The password must be accepted in both the trusting and trusted domains. Remember to use the Run As feature to administer a domain to which you are not currently logged on.

## Modifying Trusts

Even though trusts are created by default, if your enterprise consists of multiple down-level domains, you may need to modify these trusts. Cross-link trusts may also require verification at certain timed intervals (such as in the event of a down-level domain upgrading to Windows 2000, or the separation of a previous trust collaboration).

To verify a trust, follow these steps:

1. Open Active Directory Domains and Trusts.
2. In the console tree, right-click one of the domains involved in the trust you want to verify, and then click Properties.
3. Click the Trusts tab.
4. In either Domains Trusted By This Domain or Domains That Trust This Domain, click the trust to be verified, and then click Edit.
5. Click Verify/Reset.

To revoke a trust, follow these steps:

1. Open Active Directory Domains and Trusts.
2. In the console tree, right-click one of the domain nodes involved in the trust you want to verify, and then click Properties.
3. Click the Trusts tab.
4. In either Domains Trusted By This Domain or Domains That Trust This Domain, click the trust to be revoked, and then click Remove.
5. Repeat this procedure for the other domain involved in the trust.

#### NOTE

You cannot revoke the default two-way transitive trusts between domains in a forest. Only manual trusts can be removed.

## Name Resolution in Active Directory

Windows 2000 prefers the Domain Name System (DNS) as its main name resolution method. DNS is an integral part of Active Directory and must be installed on your network. Active Directory clients use DNS servers to locate Active Directory domain controllers as well as other services on the network.

Explaining DNS fully would require a book in itself; therefore, this section covers only the specifics.

## Naming Standards

Your naming convention should follow the Internet standard character set permitted for use in DNS host naming. Standard characters, which are defined in RFC 1123, include all uppercase letters (A–Z), lowercase letters (a–z), numbers (0–9), and the hyphen (-). If you implemented



NetBIOS with more unconventional names, existing computer names might not conform to the DNS naming standard. If this is the case, consider revising your computer names.

**NOTE**

You cannot change the NetBIOS name of the machine after upgrading to Windows 2000. You must edit the Registry because there is no GUI tool to make this change at the present time.

To ease the transition from NetBIOS names to DNS domain names, the Windows 2000 DNS service includes support for extended ASCII and Unicode characters.

**NOTE**

ASCII and Unicode character support can be used only in a pure Windows 2000 network environment because most other DNS client software is based on RFC 1123, the specification that standardizes Internet host naming requirements.

If a non-standard DNS domain name is entered during Windows 2000 setup, a warning message appears recommending the use of a standard DNS name.

In earlier versions of Windows NT, a NetBIOS name was used to identify a Windows computer on the network. In Windows 2000, a computer can be identified by

- Its NetBIOS computer name, which is optional and used for NT 4.0 interoperability. For example, dc1.
- The fully qualified domain name (FQDN) for the computer. For example, dc1.kevinko-cis.com.
- Its primary (or default) Windows 2000 name, which would also be dc1 for this example.

The full computer name is a combination of both the computer name and the DNS domain name for the computer. The DNS domain name for the computer is part of the system properties for the computer and is not related to any specifically installed networking components.

The NetBIOS computer name is implemented to ensure interoperability between NetBIOS and DNS naming in Windows 2000. The value of this parameter, which is not required in a pure Windows 2000 environment, is derived from the first 15 characters of the DNS full computer name.

When the full computer name is a combination of the computer name and the DNS domain name for the computer, the impact of renaming and making the transition from a NetBIOS namespace to a DNS namespace can be minimal. Users continue to focus on the short computer name. If this name is 15 or fewer characters, you can keep the name identical to the NetBIOS computer name. You can then also assign a DNS domain name for each computer by using remote administration tools.

## Name Restrictions

Different DNS implementations impose different character and length restrictions. Table 6.1 shows the restrictions for each implementation.

**TABLE 6.1** DNS Name Restrictions

<i>Restriction</i>	<i>Standard DNS (Including Windows NT 4.0)</i>	<i>DNS in Windows 2000</i>	<i>NetBIOS</i>
Characters	Supports RFC 1123, which permits <i>A</i> to <i>Z</i> , <i>a</i> to <i>z</i> , 0 to 9, and the hyphen (-)	Several different configurations are possible, as described at the end of this section.	Unicode characters, numbers, whitespace, symbols: !@#\$ % ^ & ' )( . - _ { } ~
Fully qualified domain name length	63 bytes per label and 255 bytes for an FQDN	63 bytes per label and 255 bytes for an FQDN; domain controllers are limited to 155 bytes for an FQDN.	15 bytes

**NOTE**

Although you can create long, complex DNS names, creating shorter, user-friendly names is recommended.

Microsoft has proposed that the DNS name specification be readjusted to accommodate a larger character set: UTF-character encoding, which is a superset of ASCII and a translation of the UCS (also known as Unicode) character encoding. The UTF-character set includes characters from most languages.

You can configure the Windows 2000 DNS server to allow or disallow the use of UTF-characters on your Windows 2000 server on a per-server basis with the DNS console. From the Advanced tab of the server properties page, set Name Checking to one of the following:

- Strict RFC (ANSI)—Allows A to Z, a to z, the hyphen (-), the asterisk (\*) as a first label; and the underscore (\_) as the first character in a label.
- Non RFC (ANSI)—Allows all Strict RFC (ANSI) characters as well as placement of the underscore (\_) anywhere in a name.
- Multibyte (UTF-8)—Allows all Non RFC (ANSI) characters, as well as UTF-8 characters.
- Any character—Allows any character, including UTF-8 characters.

## DNS Server Roles

For Active Directory to function properly, DNS servers must maintain high availability. To ensure this, you might not want to rely only on AD domain controllers for DNS. You should provide at least a primary and a secondary name server per domain. This way, you can load balance between servers and provide quicker access and redundancy. Clients should be configured to query both a primary and secondary DNS server. You can configure this with DHCP, which is covered in the next chapter.

### NOTE

Clients use an application called a resolver to query DNS servers.

Let's take a closer look at the various DNS server roles.

### Primary Name Server

The primary name server is the original source of address data (zone files for zone transfers) for the domain. It is also the Start of Authority (SOA) for the domain and holds the master files. There can be only one primary name server per DNS zone.

### Secondary Name Server

The secondary name server contains authoritative address data for the domain, which it receives from the primary server in the form of zone transfers. Secondary name servers relieve the primary name server's load by answering queries. As you learned earlier, this server provides redundancy and load balancing for the domain.

## Forwarders and Slaves

When a DNS server receives a query, it first searches its local zones and cache. If it does not find the requested information and is not authoritative for the requested information, it queries other servers to resolve the request.

To solve this problem, DNS allows for the use of forwarders, which are DNS servers that provide forwarding of offsite queries for other DNS servers. A forwarder is basically a designated machine for handling queries. It waits for a response, and if it doesn't receive one, it searches for an answer itself.

A slave performs similarly to the forwarder, except that it does not attempt to resolve the address itself. When a slave receives a DNS query that it cannot resolve through its own zones, it passes the query to a forwarder. If the forwarder cannot resolve the request, the slave returns a query failure to its requestor. Slaves do not resolve queries on their own.

## Caching-Only Servers

All DNS servers perform caching when they receive information from other servers and store the information for a certain amount of time. This capability enhances DNS resolution and reduces traffic associated with DNS queries.

Caching-only servers simply perform queries and cache the answers, therefore they do not generate zone transfer network traffic because they do not contain any zones.

Caching-only servers are ideal for companies with an Internet service provider (ISP) providing primary DNS services.

## Resource Records

Resource records (RRs) are sets of information in the DNS database for processing client queries. Each DNS server hosts the resource records it needs to handle authoritative queries.

### NOTE

A DNS server is authoritative for a contiguous portion of the DNS namespace if it contains information about that portion of the namespace.

In DNS, resource records are represented in binary form for queries and replies. Resource records are represented as text entries in Active Directory database files.

## Resource Record Types

Different types of resource records can be used to provide DNS-based data about computers on a TCP/IP network. This section describes the following resource records:

- SOA
- NS
- A
- PTR
- CNAME
- MX
- SRV

### SOA Resource Records

Every zone contains a Start of Authority (SOA) resource record at the beginning of the zone. They contain general zone information, including the authoritative DNS server for the zone and various expiration parameters.

### NS Resource Records

The name server (NS) resource record indicates the servers authoritative for the zone, including primary and secondary servers specified in the SOA resource record. Every zone must contain at least one NS record at the zone root.

### A Resource Records

The address (A) resource record maps an FQDN or host name to an IP address, so clients can obtain an IP address for an FQDN.

### PTR Records

The pointer (PTR) resource record performs tasks opposite from those performed by the A resource record. This record maps an IP address to an FQDN or host name, and is used in reverse lookup zones.

### CNAME Resource Records

The canonical name (CNAME) resource record creates an alias (synonymous name) for the specified FQDN or host name. CNAME records can hide the implementation details of your network from the clients that connect to it. This can be useful in a proxy situation where you want to hide the actual proxy FQDN from your user population.

**NOTE**

According to RFC 2181, each alias can have only one canonical name.

**MX Resource Records**

The mail exchange (MX) resource record specifies a mail exchange server for a DNS domain name. A mail exchange server is a host that either processes or forwards mail for the DNS domain name.

**NOTE**

Only mail exchange servers use MX records.

You can have multiple MX resource records for multiple mail exchange servers in a domain and assign them different weight preferences. The lower-weighted mail server will be contacted first.

**SRV Records**

Service (SRV) resource records specify the location of the servers for a specific service, protocol, and DNS domain.

The format of an SRV record is as follows:

`_Service._Proto.Name TTL Class SRV Priority Weight Port Target`

- The **Service** field specifies the name of the service, such as `http` or `telnet`.
- The **Proto** field specifies the protocol, which is usually `TCP` or `UDP`.
- The **Name** field specifies the DNS domain.
- The **TTL** field is the Time to Live (optional).
- The **Class** field often represented by `IN` (for Internet). This field is optional.
- The **Priority** field specifies the priority of the host (a number from 0 to 65,535). The host with the lowest number has priority.
- The **Weight** field is used for load-balancing when hosts have the same priority (a number from 0 to 65,535).
- The **Port** field shows the port of the service on this host (a number from 0 to 65,535).
- The **Target** field shows the FQDN for the host providing the service.

If a computer needs to locate a Web server in the kevinkocis.com DNS domain, the client sends the following query:

```
_http._tcp.www.kevinkocis.com
```

The DNS server replies with the SRV records listed earlier. The client then chooses between servers by looking at their priority values.

#### NOTE

If the priority values are the same but the weight values are different, the client would randomly choose a Web server, except that the server with the highest weight value would have a higher probability of being chosen.

Next, the client requests the A record for web1.kevinkocis.com, and the DNS server sends the A record. Finally, the client attempts to contact the Web server.

## Zones and Zone Files

A zone is a contiguous portion of the DNS namespace hosted on a specific domain node. A zone is a portion of the DNS namespace generally stored in a file, and can contain multiple domains. A domain can be subdivided into several partitions or zones, which can be controlled by a separate DNS server. Using the zone, the DNS server answers queries about hosts in its zone, and is authoritative for that zone.

Zones fall into three classifications:

- Standard primary
- Standard secondary
- Active Directory–integrated

These zone types are created in the DNS Wizard, and are stored either in files or in Active Directory. A primary zone is the copy of the zone to which the updates are made. It is stored in a text file. A secondary zone is a read-only copy of the zone that is replicated from a master server, and is also a text file.

Standard primary and secondary zones are stored as zone files on the server's hard drive. Some secondary servers store them in memory and perform a zone transfer whenever they are restarted. Active Directory–integrated zones are stored and replicated in the Active Directory.

**NOTE**

Only one server can manage the primary zone for each DNS domain. You cannot configure two different servers to manage the same primary zones.

There is one exception, however: Multiple computers can manage Windows 2000 Active Directory–integrated zones. You can configure a single DNS server to manage one zone or multiple zones, depending on your needs. You can create multiple zones to distribute administrative tasks to different groups and provide efficient data distribution. You can also store the same zone on multiple servers to provide load balancing and fault tolerance.

## Lookup Zones

Lookup zones store the information required to resolve host names and IP addresses within the domain. Depending on which variable you have (host name or address), you'll use forward or reverse lookup zones, respectively.

### Forward Lookup Zone

Forward lookup zones contain information needed to resolve names within the DNS domain. They must include SOA and NS records and can include any type of resource record except the PTR resource record. With most queries, the client supplies a name and requests the IP address that corresponds to that name. This type of query is typically described as a forward lookup.

### Reverse Lookup Zone

Reverse lookup zones contain information needed to perform reverse lookups. They usually include SOA, NS, PTR, and CNAME records.

If you already have the IP address but need the client name, you would use a reverse lookup.

## Dynamic DNS and Zone Transfers

Windows 2000 supports both dynamic and secure dynamic updates. With dynamic updates, clients can automatically send updates to the domain's authoritative name server. The authoritative name server then verifies that certain prerequisites have been met. If the prerequisites have been met, the authoritative name server makes the change.

**NOTE**

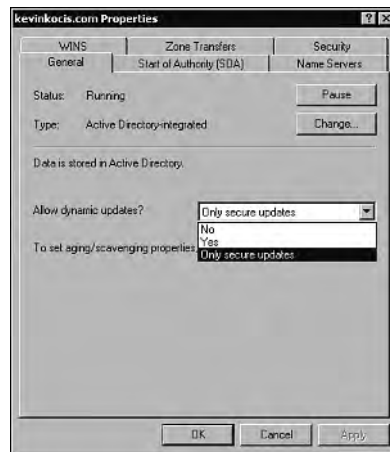
Both clients and servers can send dynamic updates.



Dynamic update provides the following benefits:

- Reduces administrative overhead by allowing clients (including DHCP clients), to dynamically register A and PTR resource records with a primary server.
- Allows DHCP servers to register A and PTR resource records on behalf of DHCP clients (Windows NT and 9x clients).
- Allows domain controllers to dynamically register their SRV records.

Secure dynamic update works like dynamic update, but requires authentication to update the dnsZone and dnsNode objects. See Figure 3.4 for setting dynamic updates.



**FIGURE 3.4**

*Setting dynamic updates.*

Secure dynamic update allows you to protect zones and resource records from being modified without authorization and enables you to specify exactly which users and groups can modify zones and resource records.

#### NOTE

Although any primary zone can be configured for dynamic update, only Active Directory-integrated zones can be configured for secure dynamic update. If you disable secure dynamic update, the client cannot perform updates on zones that have been configured for secure dynamic update.

By default, the dynamic update client automatically deregisters name-to-IP address mappings whenever the DHCP lease expires. You can configure the client not to register its name and IP address in DNS. If you configure the client not to automatically register name-to-IP address mappings, the DHCP server is running Windows 2000, and it is configured to register DNS resource records on behalf of clients that are running versions of Windows earlier than Windows 2000, the DHCP server attempts to update the mappings instead.

To prevent the client from registering name-to-IP address mappings, follow these steps:

1. Double-click the Network icon in Control Panel.
2. Right-click the icon for the connection on which you want to disable registration of name-to-IP address mappings, and then click Properties.
3. Click Internet Protocol (TCP/IP), and then click Properties.
4. Click Advanced, and then click the DNS tab.
5. Clear the Register This Connection's Address in DNS check box.

You can force a re-registration by using the command-line tool Ipconfig. For Windows 2000-based clients, type the following at the command prompt:

```
ipconfig /registerdns
```

For Windows NT 4.0-based clients, type the following:

```
ipconfig /release
```

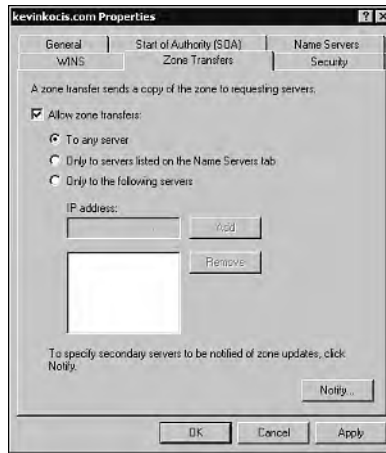
```
ipconfig /renew
```

For Microsoft Windows 9x-based clients, type the following:

```
winipcfg /renew
```

## Zone Transfer

Zone changes made to a master server must be replicated to all the secondary servers for that zone. This process is called a *zone transfer*. Traditionally, only one form of zone transfer—known as full zone transfer (AXFR)—was available. Active Directory incorporates a new type of zone transfer, the incremental zone transfer (IXFR). Let's take a closer look at these zone transfers, as shown in Figure 3.5.

**FIGURE 3.5**

*The zone transfer options.*

## Full Zone Transfer

In a full zone transfer, the zone's master server transmits the entire zone database to the secondary server for that zone. Secondary servers initiate full zone transfers using the following process:

1. The secondary server polls the master server at the time interval set in the Refresh field of the State Of Authority (SOA) resource record tab.
2. The master server responds with the SOA resource record.
3. The secondary server compares the serial numbers of the SOA records. If the master server's serial number for the zone is higher than the secondary server's serial number, its zone database is out of date, and the secondary server sends an AXFR request (a request for a full zone transfer).
4. The master server then sends the full zone database to the secondary server.

If the master server for the zone does not respond to polling by the secondary server, the secondary server continues to poll based on the interval specified in the Retry field of the SOA resource record. If there is still no answer after the interval specified in the Expire field since the last successful zone transfer, it discards its zone and stops responding to client requests.

**NOTE**

Name servers running versions of Berkeley Internet Name Domain (BIND) earlier than 4.9.4 can send and receive only one resource record per message during a full zone transfer. Windows 2000 and later versions of BIND can send and receive multiple resource records per message. This capability improves the performance of full zone transfers. For more information about BIND, visit the Internet Software Consortium Web site at <http://isc.org>.

## Incremental Transfer

Full zone transfers can affect network bandwidth. Because of this, a new standard was defined, which is called the incremental zone transfer (IXFR).

Incremental zone transfer works much the same as full zone transfer except that it transfers only the modified parts of the zone. In this situation, if a zone transfer is required, it sends an incremental zone transfer (IXFR) query instead of a full zone transfer (AXFR) query, requesting that the master server for the zone perform an incremental zone transfer.

The master server sends the oldest updates first and the newest updates last to the secondary server. When it receives an incremental zone transfer, the secondary server creates a new version of the zone and begins replacing its resource records with the updated resource records, starting with the oldest updates. After all the updates have been made, the secondary server replaces its old version of the zone with the new version of the zone.

## Integrating DNS and Active Directory

The integration of DNS and Active Directory is a key feature of Windows 2000. Like DNS, Active Directory is a distributed database that can be partitioned and replicated. Active Directory domains and DNS domains use identical names for different namespaces. Active Directory uses DNS as its location service, enabling computers to find the location of domain controllers and other services on the network. LDAP is the protocol used to query and update Active Directory, and all domain controllers run an LDAP server.

You cannot install Active Directory without having DNS on your network because Active Directory uses DNS as its location service. However, you can install DNS separately, without Active Directory. If you install DNS on a domain controller, you can also choose whether or not to use Active Directory to provide storage and replication for DNS. Using Active Directory for storage and replication provides the following benefits:

- Increased fault tolerance
- Security
- Easier management
- More efficient replication of large zones

For DNS to function as a location service for Active Directory, you must have a DNS server to host the locator records (A, SRV, and CNAME).

You can configure your Windows 2000 DNS server automatically by using the Active Directory Installation Wizard, which performs all the installation and configuration necessary for DNS, and the Netlogon service adds the necessary locator records.

You can manually configure DNS if you want to set up a configuration other than the Active Directory default configuration (such as BIND).

For information about issues related to configuring DNS when you're using a third-party DNS server, see "Heterogeneous Environments" later in this chapter.

## DNS Installation Wizard

The Active Directory Installation Wizard promotes the computer to the role of domain controller, installs Active Directory, and can install and configure the DNS server.

When you start the Active Directory Installation Wizard and choose to create a new domain, the wizard finds the DNS server that is authoritative for the name of the new Active Directory domain and then checks whether that server is going to accept dynamic updates. If the test is positive, the wizard does not install and configure a local DNS server.

If the Active Directory Installation Wizard cannot find the DNS server that is authoritative for the name, or if the server it finds does not support dynamic updates or is not configured to accept dynamic updates, the wizard asks whether you want it to automatically install and configure a local DNS server. If you answer yes, the wizard automatically installs and configures the DNS Server service.

During automatic configuration, the Active Directory Installation Wizard adds to the DNS server the forward lookup zone that will host the locator records and configures the DNS server to accept dynamic updates. (A forward lookup zone contains information needed to resolve names within the DNS domain.) If the server is the first in the forest, it becomes the root DNS server. If the server is not the first, the wizard queries for the root servers and primes the root hints with the root DNS server names.

After the Active Directory Installation Wizard is finished, you are prompted to restart the computer. After the computer restarts, Netlogon attempts to add locator resource records to the DNS server by sending a dynamic update request to the authoritative DNS server.

**NOTE**

The Netlogon service starts after the DNS server service. The SRV resource records may not be registered in the zone for up to 15 minutes. You can force registration of these records by stopping and restarting the Netlogon service.

**NOTE**

You can also invoke the Active Directory Installation Wizard by executing an answer file that contains all the settings you need to configure. An answer file is a file that a wizard uses to provide answers to questions where a user would normally need to respond or be prompted to input information.

Follow these steps to install and configure DNS and Active Directory:

1. Log on with the appropriate administrative privileges. Depending on the type of DC promotion, the Eadmin account may be required.
2. Check the TCP/IP advanced settings of your computer to make sure that it is configured to use a DNS server. If your computer is the first DNS server on the network, you can configure your computer to use itself as a DNS server.
3. If the Windows 2000 Configure Your Server Wizard is not already open on your computer, click Start, Run, and then type **dcpromo**.
4. The Active Directory Installation Wizard then guides you through the installation and configuration of the DNS server component.
5. When you're directed to do so, restart your computer.

After you run the Active Directory Installation Wizard, you might need to add a delegation in the parent zone of the zone you created. If this server is a root DNS server, no parent zone exists; therefore, you do not need to add a delegation. However, if other DNS servers are running on the network, you should add a delegation if this zone will be managed outside of the root domain.

Follow these steps to add a delegation:

1. In the DNS console, locate the subdomain where you want to create a zone delegation.
2. From the Action menu, select New Delegation. Click Next.
3. On the Delegated Domain Name page, specify the domain you want to create (select the recently created domain you just installed in DNS), and click Next.

4. Specify the servers hosting the delegated zone, and click Next.
5. Review your entered information, and click Finish.

## Configuring Zones

The biggest part of configuring DNS involves configuring zones. After you have installed DNS, you will eventually be required to configure DNS zones. This next section addresses the Windows 2000 DNS console and how to configure various elements of zone creation.

### Adding and Deleting Zones

As mentioned earlier, you can configure zones as standard primary, standard secondary, or Active Directory–integrated.

To add a standard primary zone, perform the following steps:

1. Select Start, Programs, Administrative Tools, DNS.
2. In DNS, locate the server designated to be the primary server for the new zone.
3. Right-click the Forward Lookup Zone icon and select New Zone.
4. At the zone selection screen, select Standard Primary, and click Next.
5. Enter the domain name (this should correspond to your Active Directory namespace).
6. Click the Create a New File button if you are not importing or working with a current file. (Note that the default name is the zone name with an appended .dns extension.) If you are using an existing file, it must be located in the root\system32\dns folder.
7. Review your information, and select Finish.

To create a secondary forward lookup zone, follow steps 1 through 5, and then enter the IP address(es) of the DNS server(s) from which you want to copy the DNS zone information. Click the Add button, and prioritize the list of DNS servers. Then review your information, and click Finish.

### Adding a Reverse Lookup Zone

All zones (primary, secondary, and AD-integrated) can be either forward lookup or reverse lookup. A reverse lookup zone returns the host name when queried with the IP address.

To create a primary reverse lookup zone, perform the following steps:

1. Select Start, Programs, Administrative Tools, DNS.
2. In DNS, locate the server designated to be the primary server for the new zone.
3. Right-click the Reverse Lookup Zone icon and select New Zone.
4. At the zone selection screen, select Standard Primary, and click Next.

5. Enter the network ID for the zone (or enter the name, which is the reversed network ID followed by .in-addr.arpa) For example, if the network ID is 10.1.1, the reverse lookup zone name would be .10.1.1.in-addr.arpa. Click Next.
6. Click the Create a New File button if you are not importing or working with a current file. (Note that the default name is the zone name with an appended .dns extension.) If you are using an existing file, it must be located in the root\system32\dns folder.
7. Review your information, and select Finish.

To delete a zone, simply right-click the desired zone in the DNS console, and select Delete.

## Active Directory–Integrated Zones

Any zone you create is automatically replicated to all domain controllers in the zone. Therefore, do not create the same zone on more than one domain controller.

### NOTE

If you create a zone on one domain controller and then create the same zone on a second domain controller before Active Directory has replicated the zone, Active Directory deletes the zone on the first domain controller. As a result, you lose any changes that you made to the version of the zone that you created on the first domain controller.

To create an Active Directory–integrated zone, perform the following steps:

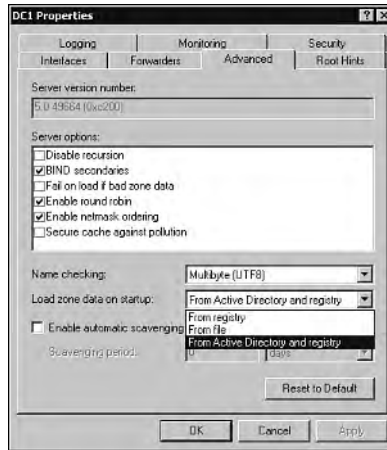
1. Select Start, Programs, Administrative Tools, DNS.
2. In DNS, locate the server designated to be the primary server for the new zone.
3. Right-click the Forward Lookup Zone icon, and select New Zone.
4. At the zone selection screen, select Standard Active Directory–Integrated Zone, and click Next.
5. Enter the domain name (this should correspond to your Active Directory namespace).
6. Review your information, and click Finish.

You can store many zones in Active Directory, which will act as primary zones. These zones can be modified by any DNS server running on a domain controller in the respective zone.

If you delete an Active Directory–integrated zone from a domain controller and Load Zone Data on Startup is set to Registry, the DNS console asks whether you also want to delete the zone from Active Directory. If you click Yes, the zone is completely deleted from Active



Directory and is no longer available to any domain controllers. If you click No, the zone is removed from the Registry but remains in Active Directory. The next time the DNS server polls the directory for changes, if Load Zone Data on Startup on the Advanced tab of the DNS server properties page in the DNS console is set to From Active Directory and Registry, the zone reappears (see Figure 3.6). If Load Zone Data on Startup is set to Registry, on the other hand, the zone does not reappear.



**FIGURE 3.6**

*Setting the load zone data preference.*

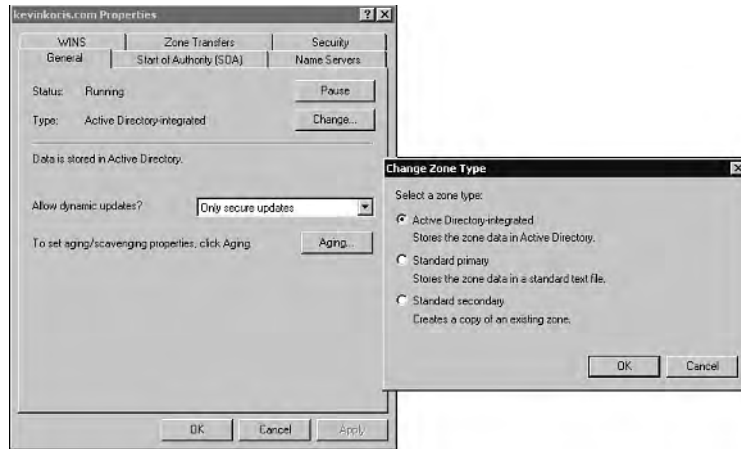
## Converting Standard Zones to AD-Integrated Zones

You can convert either a standard primary or secondary zone to an Active Directory–integrated zone. When you integrate a zone with Active Directory, consider the following issues:

- For a DNS server to use an Active Directory–integrated zone, that server must be running on a domain controller.
- You cannot load Active Directory–integrated zones from other domains. If you want your DNS server to be authoritative for an Active Directory–integrated zone from another domain, the server can only act as a secondary server for that zone.
- There is no such thing as an Active Directory–integrated secondary zone. All domain controllers can update the zone.
- You cannot have at the same time both an Active Directory–integrated zone and a standard primary copy of the same zone.

## Converting AD-Integrated Zones to Standard Zones

You can convert an Active Directory–integrated zone to either a standard primary or standard secondary zone (see Figure 3.7).



**FIGURE 3.7**

*Converting an AD-integrated zone to a standard primary zone. You can use this same window in the General tab to convert back to AD-integrated.*

If you convert an Active Directory–integrated zone to a standard secondary zone, the zone is copied to the name server on which you converted the zone. Although the server no longer loads the zone from Active Directory, it hosts its own secondary copy of the zone, and requests zone transfers from the primary server for the zone.

If you convert an Active Directory–integrated zone to a standard primary zone, the zone is copied to a standard file on that server and is deleted from Active Directory. The zone no longer appears on other Active Directory–integrated DNS servers.

## Preventing Problems When Converting or Deleting Zones

When you delete a zone or convert an Active Directory–integrated zone to a standard secondary zone, configuration errors can result. For example, if you delete a copy of the zone from a server and a secondary server is configured to pull zone transfers from that server, the secondary server is no longer able to pull zone transfers.

Also, if you convert an Active Directory–integrated zone to a standard primary zone, the DNS server loading the new primary zone becomes the single master of the zone. Because Active Directory removes the converted zone from Active Directory, the zone is deleted from all domain controllers.

To prevent this problem, be sure to update all secondary servers for the zone that you are converting from an Active Directory–integrated zone to a standard primary zone. This problem occurs only if you delete a zone from a server or you are converting an Active Directory–integrated zone to a standard primary zone, and a secondary server is pointing at a server from which the zone was deleted. The problem does not occur if you are converting an Active Directory–integrated zone to a standard secondary zone because converting this way does not cause the zone to be deleted from any server.

## Heterogeneous Environments

In the real world, DNS has been managed effectively on UNIX servers for years. Many UNIX administrators see no added value to implementing Microsoft DNS. This section addresses some of the issues from both perspectives.

This section discusses the issues that may arise when Microsoft DNS servers are used in a mixed environment with non-Microsoft DNS servers. Because the Microsoft DNS server is RFC-compliant, it is fully interoperable with all other RFC-compliant DNS servers. However, because the Microsoft DNS server provides a wider spectrum of features than specified in the RFC, you are advised to exercise caution when using these features. These features are limited to the use of Windows Internet Name Services (WINS), WINSR resource records, and UTF-8 character encoding.

### Using WINS and WINSR Records

Because currently only Microsoft DNS servers support the WINS and WINSR resource records, I recommend disabling replication of these records if all the following conditions are satisfied:

- The primary copy of the zone contains one of these records.
- At least one of the secondaries resides on a non-Microsoft DNS server.

At the same time, if the secondaries reside partially on Microsoft and non-Microsoft DNS servers, disabling WINS and WINSR resource records replication may require manual input of these records to the secondary zones residing on the Microsoft DNS servers.

### Using UTF-8 Characters Format

The Windows 2000 DNS server can be configured to allow or disallow the use of UTF-8 characters on a per-server or per-zone basis. A non-UTF-8–aware DNS server may accept a zone transfer of a zone containing UTF-8 names, but it might not be able to write back those names to a zone file or reload those names from a zone file. Administrators should exercise caution when transferring a zone containing UTF-8 names to a non-UTF-8–aware DNS server.

## Receiving Non-RFC-Compliant Data

If an Active Directory domain controller supports a secondary zone and receives unknown resource records, it drops such records and continues zone replication. The secondary server also drops circular CNAME resource records if it receives them.

## UNIX/BIND

Storing your Microsoft DNS information in Active Directory offers a significant advantage. In standard DNS, replication is single master, pushing updates to secondary servers. This leaves a single point of failure, so many companies implement primary and backup DNS servers. However, if you implement ADS storage of DNS, replication is multi-master because ADS replicates between the domain controllers running DNS on your network. With ADS storage of Microsoft DNS, you don't need to manage a separate replication structure, transfers are secure (managed by trusts in AD), and there is no single point of failure. You can also send standard zone transfers to other servers as necessary. With ADS storage, DNS data is converted to an object model in which a DNS name becomes the object and the resource record set is the attribute.

Performance and manageability advantages promote the integration of AD with DNS. There are a few caveats. For one, only primary zones can be AD-integrated, so the DNS zone must be running Windows 2000, not a third-party DNS such as BIND or NetWareDNS. Only domain controllers can host AD-integrated zones, although you can have read/write access from any client loaded with the DNS snap-in. Another is the manual process of importing current zone files into Windows 2000 DNS. The only current method for doing so is to move the pre-created zone file in the systemroot\system32\dns folder and then indicate to use that zone file when you set up the zone as primary. Then you can convert this zone to an AD-integrated zone.

However, despite all these new features and caveats, the challenge remains for corporations whether they will implement Microsoft DNS into their environments and how they will perform this integration.

The primary benefits for interoperability in these environments include

- Full interoperability with other DNS server implementations that implement RFC-compliant behavior for DNS name service
- Use of Windows DNS servers to provide DNS service on the Internet

For interoperability testing, the Windows 2000 development team has tested the Windows 2000 DNS Server service with the following versions of the Berkeley Internet Name Domain (BIND) DNS server implementation:

- BIND 4.9.7
- BIND 8.1.2
- BIND 8.2

Active Directory is completely dependent on DNS. However, great challenges and significant planning go into designing an effective directory service. Perhaps the greatest of these challenges in the enterprise for Microsoft AD implementation is interoperability. Because most enterprises currently host their DNS on UNIX servers running BIND, exactly how to integrate Active Directory into this environment will prove challenging. Because clients in a Windows 2000 environment look up SRV resource records in the DNS server to locate their network's AD and services, it is important that UNIX servers have recent BIND versions installed to perform these functions.

Some of the new DNS requirements of Active Directory are as follows:

- Support of SRV records (RFC 2782)
- Recommended support of dynamic updates (RFC 2136)
- Recommended support of incremental zone transfer (IXFR) (RFC 1995)

**NOTE**

BIND 8.2.2 or higher supports DNS extensions used by Active Directory.

Windows 2000 clients use DNS for name resolution and for locating domain controllers for logon. Down-level clients (Windows NT 4.0 and earlier, Windows 9x) rely on NetBIOS, which uses WINS, broadcast, or LMHOSTS files. WINS is used for domain controller location. Because Windows 2000 DNS is WINS-aware, a combination of DNS and WINS can be implemented in a mixed environment. Windows NT 4.0 clients can register in Windows 2000 WINS, and Windows 2000 clients can register in Windows NT 4.0 WINS.

The minimum DNS requirement for Active Directory integration is support of SRV resource records. BIND 4.9.6 and higher versions meet this requirement. However, upgrading to at least 8.x is strongly recommended to support dynamic updates. Note that BIND 8.2.2 supports integration with Active Directory, including dynamic updates, incremental zone transfers, and SRV record updates.

The Dynamic Update Protocol (RFC 2136) allows hosts to register domain names and IP addresses with the name service, which in turn allows for automatic namespace updates and alleviates manual administrative updates—which is important if you're using DHCP to assign IP addresses dynamically.

The Incremental Zone Transfer Protocol (RFC 1995) allows for incremental updates in the zone transfer process as opposed to transferring the entire zone file. This protocol alleviates bandwidth demands during zone transfers.

The Service Location Resource Record (RFC 2782) allows services to be published in DNS by specifying the location of the server(s) for a specific protocol and domain. The SRV record is used to locate AD services such as LDAP at port 389. It does not use round robin as an A record query would.

To determine whether your version of BIND supports dynamic record updates, use the `nsupdate` tool that ships with BIND. You can create a test domain and its zone file in your DNS server. Then you can turn on dynamic updating by using the `nsupdate` tool to perform manual dynamic updates.

**NOTE**

It is imperative that you coordinate and plan your Active Directory and UNIX DNS integration with your current DNS team.

Although implementing AD and Microsoft DNS may sound quite enticing to the Windows support team of a larger company, if you are operating in a heterogeneous environment, the debate over directory services may fall nothing short of a technological holy war.

Many large enterprises have been hosting their DNS domains on UNIX servers for a long time. From their perspective, why change something that isn't broken, especially to an unproven and proprietary Microsoft product? Windows DNS has raised the stakes by complying with Internet standards and providing a wider spectrum of features than specified in the current RFC documents. Because of its advanced features, you need to be cautious when planning integration, particularly AD-integrated zones.

Microsoft believes strongly the following features of Windows 2000 DNS make it a good choice for corporations looking to implement a reliable hierarchical distributed network environment:

- AD integration
- Incremental zone transfer
- Dynamic update and secure dynamic update
- Unicode character support
- Enhanced domain locator
- Enhanced caching resolver service

- Enhanced DNS Manager
- Record scavenging

Remember that some of the UNIX Internet DNS servers in your environment are currently stable and secure. Add to this the fact that many UNIX administrators feel that Microsoft tends to “alter” existing technologies and preface them with its name (that is, Microsoft TCP/IP, Microsoft DNS), and you understand their concern.

## DNS Integration Options

When you’re integrating AD into an existing DNS infrastructure, your discussions should focus on whether the AD namespace will join, overlap, or trump your existing DNS namespace. If you are in a larger corporation, chances are the AD service you are designing will need to be integrated into the existing DNS infrastructure. Let’s take a closer look at the three options for integrating Windows 2000 DNS into your current DNS.

If you are seriously considering installing Microsoft DNS as part of your Active Directory implementation, your three options are as follows:

- Implement Microsoft DNS in AD and replace current DNS services.
- Integrate your UNIX DNS structure into the DNS required for Windows 2000.
- Maintain your UNIX DNS structure with Windows 2000.

Your choice depends on a variety of variables, including your current DNS infrastructure and specifications, as well as the many pending political issues.

## Windows 2000 DNS as Primary DNS

Option 1, implementing proprietary Microsoft DNS with Active Directory, is Microsoft’s choice for obvious reasons. And if your company is committed to redesigning your DNS infrastructure around Windows 2000 Active Directory, this is your choice. If you have older UNIX machines running older versions of BIND (such as 4.x) and feel the upgrade process is not worthwhile based on the enterprise shift to Active Directory, consider this option. Migration from Windows NT 4.0 DNS is relatively easy.

## Migrating to Windows 2000 DNS

When migrating UNIX DNS servers to the Windows 2000 DNS, you should first introduce Windows 2000 DNS servers as secondary servers. Configure a zone transfer from a master to a secondary Windows 2000 DNS server, and make sure no errors occur in the zone transfer process. You may receive errors if the Windows 2000 DNS server cannot recognize records sent by the UNIX DNS server during the zone transfer. You should either repair or remove the records from the zone in order for the zone transfer to complete successfully. You can also FTP the forward and reverse zone files from your UNIX DNS server (db.xxx files located in etc/named.boot or etc/named.conf, depending on the BIND version) to the C:\winnt\system32\dns directory on your Windows 2000 DNS server.

When moving from a BIND DNS server to DNS service, however, you need to copy and rename any BIND-created zone or boot files that you intend to use with the DNS service. Also, if you continue to use a BIND boot file to provide the initial configuration settings used by the DNS service when it is started, you need to change the boot method used by the DNS service.

To migrate from BIND-based server zones to Windows 2000 DNS servers, perform the following steps:

1. At your Windows 2000 server computer, install a DNS server.
2. Using the DNS console, at the new server add secondary zones for all your existing zones hosted at the BIND-based DNS servers.  
Configure the BIND servers as the master servers for each of the secondary zones you need to create.
3. Initiate zone transfer at your Windows 2000 DNS servers to transfer the zones from the BIND servers.
4. After completing the zone transfers, convert any of the secondary zones to primary zones that were obtained from primary zones at the BIND servers.
5. For the other secondary zones that remain, update the master servers for those zones to use the new primary servers running Windows 2000 server.

If you continue to use your BIND DNS servers as secondary servers for zones for which your DNS server running Windows 2000 server is the primary server, you should review interoperability issues related to zone transfer for this configuration.

Keep in mind that any zone files created and stored on UNIX DNS servers that use BIND need to be manually copied from those servers to the `systemroot\system32\dns` folder on the computer running Windows 2000 server and appropriately renamed. BIND zone files have a different naming convention from that used by DNS servers running under the DNS service provided in Windows operating systems.

After you transfer the files, you can upgrade your secondary zones to AD-integrated zones. You should change the SOA resource record to one of the AD-integrated DNS servers. Then you can terminate your UNIX DNS servers (to avoid duplicate SOA records for the same zone) and remove them from the network.

One disadvantage comes in the form of integration. AD-integrated zones must be stored on DCs in the same domain. If you need it to cross domains, you must create secondary zones at other DNS servers outside the domain.

## **BIND as Primary DNS**

You can integrate your current DNS structure into the DNS required for Windows 2000. If your current DNS meets the recommended requirements for Windows 2000 and you have



tested dynamic updates, you can integrate it with Active Directory. This includes BIND 8.2.2 and higher, as well as Novell's NetWare 5.0. Remember that BIND 4.9.6 and 4.9.7 meet the minimum requirements. However, BIND 8.x supports dynamic updates and incremental zone transfers, and is strongly recommended for integrating with Active Directory.

The advantage to integrating your current DNS structure into Windows 2000 DNS is that less administrative effort is required to implement. Your company can maintain its current equipment and infrastructure. UNIX and NT administrators can cohabitate, and you can focus on your Windows 2000 implementation as opposed to fighting the DNS war.

Some disadvantages are that many UNIX DNS servers are running BIND 4.x, which may create a crossroads situation: upgrade or convert. The possible increase in future administrative overhead and manual data entry may be an issue. There will also be a single point of failure for dynamic registrations.

A final option is to supplement your current DNS structure with Windows 2000. If your company hasn't installed and maintained recent BIND versions on your root DNS servers, and issues have been minimal, you may decide that there's no reason to "fix something that's not broken." UNIX administrators may approach Microsoft's entry into the directory services arena very cautiously. With this option, you avoid the replacement of your current DNS, as well as additional effort.

You can delegate a new Windows 2000 DNS namespace from the existing DNS structure. When a DNS namespace is delegated off an existing DNS tree, the DNS server that owns the zone file for the newly delegated namespace becomes the primary master for that namespace. The DNS zone name should correspond to the ADS root domain. This is recommended if you want the benefits of the Windows 2000 DNS server. You can continue using the existing DNS server without delegating the Active Directory namespace as long as current DNS servers support the SRV records and dynamic updates.

One advantage of this option is that your initial integration efforts are minimized. You don't have to revamp your entire DNS infrastructure. Because your current DNS root is UNIX-based (north-rim.com), you can configure a subdomain (w2k.north-rim.com) and create a new zone strictly for your Windows 2000 clients. Another advantage is that you reduce Active Directory's dependence on your current DNS and avoid any potential incompatibility problems.

A disadvantage to this option is that it requires a separate namespace for Windows 2000 logons. This may increase administrative overhead in the long run, including managing dual DNS services. However, companies running DNS on BIND are familiar with distributed or "localized" DNS support, so hierarchical support of DNS as mentioned in this option is quite common already. As a result, many companies will likely opt for this integration solution.

If you are using the BIND boot file with the DNS service after migration, other limitations apply to the use of this file by the DNS service. For example, some BIND boot directives are not supported—in particular, xfrnets and other directives provided with versions of BIND, such as version 8.1.1 or higher.

If you are accustomed to manually editing DNS zone files, be aware that the DNS service uses RFC-compliant notation for its supported resource records (RRs). In most cases, the DNS service interprets and loads RRs from zone files originally created for BIND DNS servers without any need for file changes. If, however, you have used nonstandard record formatting, the DNS service will detect these edits and interpret them as bad or errored zone data.

## **DNS Considerations for Active Directory**

When you select the option to install and configure a DNS server using the Active Directory Installation Wizard, zones are created based on the DNS name you specified during the process of promoting the server to a domain controller. Other tasks might also be useful when the first server in the domain is promoted to a domain controller, such as changing the zone type from standard primary to Active Directory–integrated and changing the update policy for the zone to Allow Only Secure Updates.

If you are deploying DNS to support Active Directory, a simple method for redundancy and fault tolerance planning is to have a DNS server running on each domain controller. For each subnet, a good rule to follow is to have two Windows 2000 server computers configured as domain controllers so that they are also running as DNS servers that load and store only Active Directory–integrated zones.

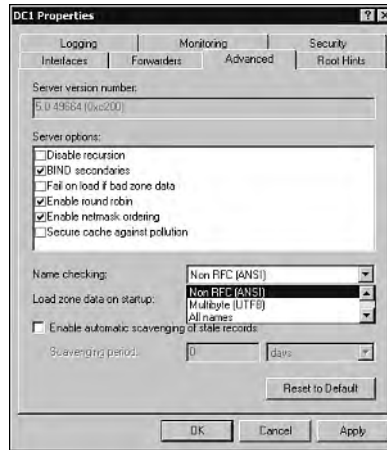
By observing these guidelines for simplified DNS and Active Directory configuration, you can enable your DNS servers to fully leverage the enhanced benefits of using Active Directory and Windows 2000 DNS servers, such as integrated storage, merged replication of Active Directory and DNS data, and secure authentication when allowing dynamic updates.

## **Zone Transfers Between AD and BIND**

When transferring a zone between two Windows 2000 DNS servers, the DNS Server service always uses a fast transfer method that uses compression. This method includes multiple resource records in each message sent to complete the transfer of the zone between servers. For Windows 2000 DNS servers, this is the default method used when initiating transfer with other DNS server implementations.

If necessary, the servers can be configured to transfer a zone using the slower uncompressed transfer format. This way, you can make successful zone transfers with DNS servers that do not support the faster transfer method, such as BIND servers prior to version 4.9.4.

When the BIND Secondaries option is checked on the Advanced tab of the server properties, no fast transfers are made (see Figure 3.8). By default, the check box is cleared to enable fast transfers.



**FIGURE 3.8**

*Enabling slow transfers to BIND servers.*

## Supporting Active Directory with Other DNS Server Implementations

In many large organizations, DNS is already implemented using other solutions, such as UNIX DNS servers that run legacy versions of BIND software. In some cases, these DNS servers are not equipped to support the DNS requirements for deploying Active Directory. This issue can be addressed in one of two ways:

- Upgrade any BIND DNS servers to version 8.1.2 or higher to meet the DNS requirements for Active Directory support.
- Use the DNS service provided with Windows 2000 server to migrate, if possible, any of your current DNS zones to Windows 2000 DNS servers.

Although the DNS service is recommended to support Active Directory, you can use other DNS server implementations for this purpose. These other implementations should support the (SRV) resource records (RFC 2782) and dynamic updates in DNS (RFC 2136).

Support for dynamic updates is recommended but not essential. Support for the SRV resource record is mandatory because it is required to provide basic DNS support to Active Directory. Windows NT Server 4.0 (updated to Service Pack 4 or higher) supports the DNS requirements of Active Directory including SRV RRs. It does not support IXFRs or dynamic updates.

Additional manual administration of SRV resource records is needed for DNS configuration support of Active Directory to function properly on a DNS server that does not support dynamic updates. For more information, see the “SRV Records” section earlier in this chapter.

If you decide to use Windows DNS service and manage it with a split DNS configuration in which one of the following is true:

- Existing DNS servers for root zones are not to be upgraded or migrated to other DNS solutions
- The DNS service and Windows 2000 server are to be deployed and provide management of any DNS domain names required to register, update, and support for use with Active Directory

Then you can modify your DNS namespace design plans in either of the following ways:

- Create a single new subdomain in your current DNS domain namespace to root your first Active Directory domain.

For example, if your organization has registered and is using a second-level domain name, such as north-rim.com, you can create a single subdomain such as ad.north-rim.com and use this domain to root the DNS domain namespace used by Active Directory. The DNS service is automatically configured to support Active Directory when you install the first domain controller.

Before you create a zone for the new subdomain at a computer running the DNS Server service, you can delegate these subdomains away at the primary zone for your second-level domain, such as north-rim.com. In some cases, you might only need to notify another DNS or UNIX system administrator in your organization to make the delegation for you.

- Create multiple subdomains based on your DNS second-level domain to support registration of Active Directory in DNS.

For example, if your organization has a registered second-level DNS domain name already in use (such as north-rim.com), you can create additional subdomains that are delegated to Windows DNS servers and used only for registering DNS names related to Active Directory.

This method is more complex to implement but causes less change to your currently deployed DNS infrastructure that is not Windows-based. With this namespace design, you create only those additional subdomains and appropriate zones needed to support your Active Directory deployment. For example, in this configuration, the domain name north-rim.com is both the root DNS and the root Active Directory domain name for your organization.

For this configuration, you first need to create zones for the following subdomains using the DNS snap-in tool at a computer running DNS service and Windows 2000 server:

```
_msdcs.north-rim.com  
_ldap._tcp.north-rim.com
```

Before these zones are created, you can delegate these subdomains away at the primary zone for your parent or second-level domain name or notify another DNS administrator who manages these zones for your organization to do so.

## DNS and WINS

In Windows NT Server 4.0 or later, the DNS service provides for the use of WINS lookup. This feature enables configured DNS zones to refer queries not answered from current zone information to a WINS server for further resolution. With this added search of the WINS namespace, both DNS and WINS are used to complete a full search of registered names for a matched response.

WINS lookup is supported for both forward and reverse lookup zones and can be enabled on a per-zone basis or configured for selected zones. This feature should also be configured to prevent replication or zone transfer of WINS resource records to servers with other DNS implementations that do not recognize the WINS resource records.

Down-level clients (Windows NT 3.5 and 3.51, Windows NT 4.0, Windows 95, and Windows 98), however, rely on NetBIOS, which can use an NBNS (WINS), broadcast, or flat LMHOSTS file. In particular, the NetBIOS name service is used for domain controller location.

### WINS Referral

WINS filled the role of domain and machine locator service for previous versions of Windows NT. Windows 2000 does not require WINS in a NetBIOS-less environment. However, WINS is always required in a mixed environment where Windows 2000-based machines interoperate with other systems such as Windows NT 4.0, Windows 9x, and Windows for Workgroups.

WINS Referral is the recommended way for Windows 2000 DNS clients to address down-level machines registered in WINS. Because Windows 2000 resolvers are optimized to use DNS, they would be much more efficient looking up down-level clients in a DNS database as opposed to a WINS database. To enable this kind of lookup, you can create a WINS referral zone in DNS that points to the WINS database. This zone does not perform any registrations or updates, as it simply refers DNS lookups to WINS.

Whenever Windows 2000-based clients send a query with an unqualified name, the default domain name suffix is tried first. Additional suffixes, however, can be supplied as part of the DHCP configuration. If the name of the WINS Referral zone is one of them, all WINS client names can be resolved.

## DHCP in Active Directory

The Dynamic Host Configuration Protocol (DHCP) allows a client to receive an IP address automatically from the DHCP server. This process avoids configuration errors caused by configuring each computer manually. DHCP helps prevent address conflicts that occur when an identical IP address is reused to configure a new computer on the network. In the case of users with portable computers who change locations frequently (and subsequently need updated client configurations), the DHCP lease renewal process helps ensure efficient and automatic updates.

Active Directory requires a DHCP server to be authorized before it can respond to client requests. If you happened to upgrade a Windows NT 4.0 DHCP server to a Windows 2000 domain controller, and the server's DHCP service isn't working, make sure the server is authorized.

To authorize the DHCP server for Active Directory, perform the following steps:

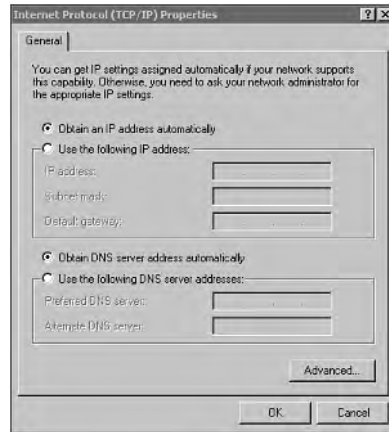
1. Select Start, Programs, Administrative Tools, DHCP.
2. In the console, right-click DHCP and select Manage Authorize Servers.
3. Click Authorize in the Manage Authorized Servers window.
4. Enter the name or IP address of the DHCP server to be authorized, and click OK.

To enable DHCP, a client must have the Obtain an IP Address Automatically radio button selected in the TCP/IP Properties property sheet (see Figure 3.9), which is accessible through the Local Area Connection icon in Windows 2000 clients, or the Network Neighborhood in Windows 9x and NT. This option is enabled by default when the client is initially installed, so if you are using DHCP, you do not need to manually set your IP configuration.

## Benefits

Here are some of the benefits of using DHCP:

- You don't need to manually change the IP settings for a mobile client that moves between different sites of your network because the client automatically receives a new IP address as long as a DHCP server is available on the new subnet.
- You don't need to manually configure settings for DNS or WINS. The DHCP server assigns these settings when you enable this option on the client by selecting the Obtain DNS Server Address Automatically option button.

**FIGURE 3.9**

*Setting a Windows 2000 client for DHCP.*

- You can avoid duplicate IP addresses conflicts and reduce network administration and manual entry errors by centrally defining global and subnet-specific TCP/IP configurations.
- Because most routers can forward DHCP configuration requests, you can eliminate the need to set up a DHCP server on every subnet.

## New Windows 2000 DHCP Features

The Windows 2000 DHCP service provides the following new features:

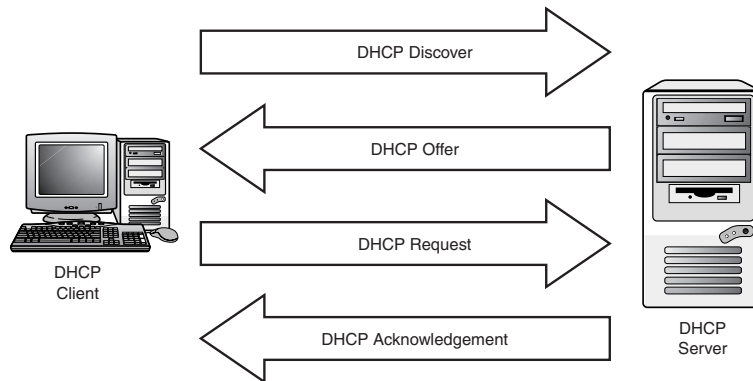
- Enhanced graphical server performance monitoring and reporting capabilities.
- DHCP with DNS integration, which allows a DHCP server to perform dynamic updates in DNS for DHCP clients supporting dynamic updates.
- Prevention of unauthorized DHCP servers from joining a DHCP network. Active Directory is queried when an unauthorized DHCP server is added to the network, and the server's IP address is compared to the list of authorized DHCP servers. If it is not on the list, its DHCP service is automatically shut down.
- Restricted access to the DHCP Manager console, which adds security to DHCP deployments by providing a special-purpose local group (the DHCP Users group) that can view, but not modify, information on the specified DHCP server. This user group is automatically added when the DHCP service is installed.

## DHCP Lease Process

The process consists of four basic steps:

- DHCP Discover
- DHCP Offer
- DHCP Request
- DHCP Acknowledge

In the Discover phase, the DHCP client asks for an IP address. In the Offer phase, the client is offered an address from the local DHCP server. In the Request phase, the client accepts the offer and requests the IP address. In the Acknowledge phase, the client is officially assigned the address. Figure 3.10 illustrates this process.



**FIGURE 3.10**

*The DHCP reservation process.*

After the client is configured, the DHCP server places a lease time on the address, which is based on the lease time setting in the DHCP options window (this value is set in seconds). Halfway through the lease period, the DHCP client requests a lease renewal, and the DHCP server extends the lease. This means that when a machine stops using its assigned IP address, the lease expires and the address is returned to the pool for reassignment. This occurs if a mobile computer leaves the network.

The four steps necessary for a DHCP client to acquire a lease from a DHCP server are initiated automatically when the computer is first booted. The following host systems can act as DHCP clients:

- Windows NT Workstation (3.5 through Windows 2000)
- Windows NT Server (3.5 through Windows 2000)
- Windows 9x computers



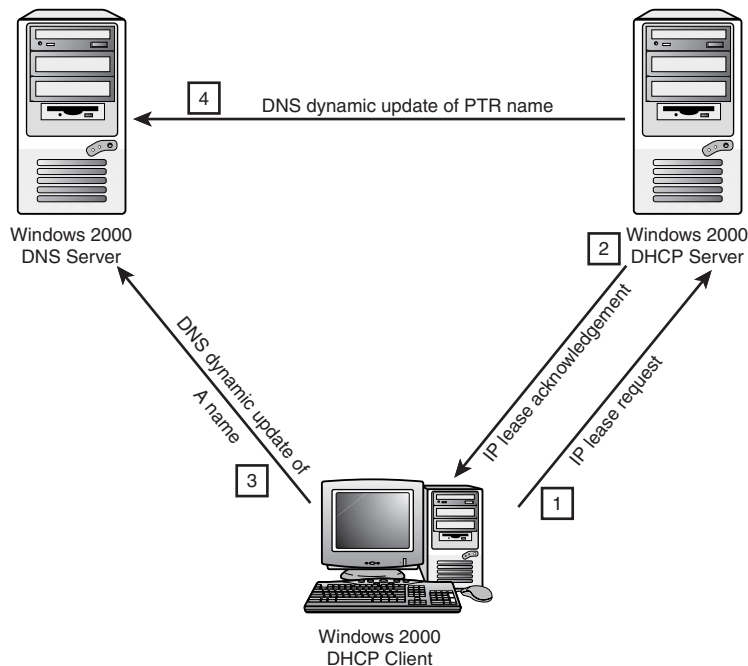
- Windows for Workgroups version 3.11 (with the Microsoft 32-bit TCP/IP VxD installed)
- Microsoft Network Client version 3.0 for the Microsoft MS-DOS operating system (with the real-mode TCP/IP driver installed)
- LAN Manager version 2.2c
- UNIX workstations
- Macintosh computers
- Network printers and print servers

## Integration of DHCP with Dynamic DNS

Windows 2000 DNS interacts with the DHCP service, allowing servers and DHCP clients to maintain synchronized name-to-IP mappings.

Windows 2000 DHCP clients and earlier versions of Windows DHCP clients interact with DNS in different ways. The DHCP server can be configured to register the DHCP client always for both the forward (A-type records) and reverse (PTR-type records) lookups with DNS. Windows 2000 DHCP clients update their own dynamic forward lookup names.

Figure 3.11 shows how Windows 2000 DHCP clients interact with dynamic updates.

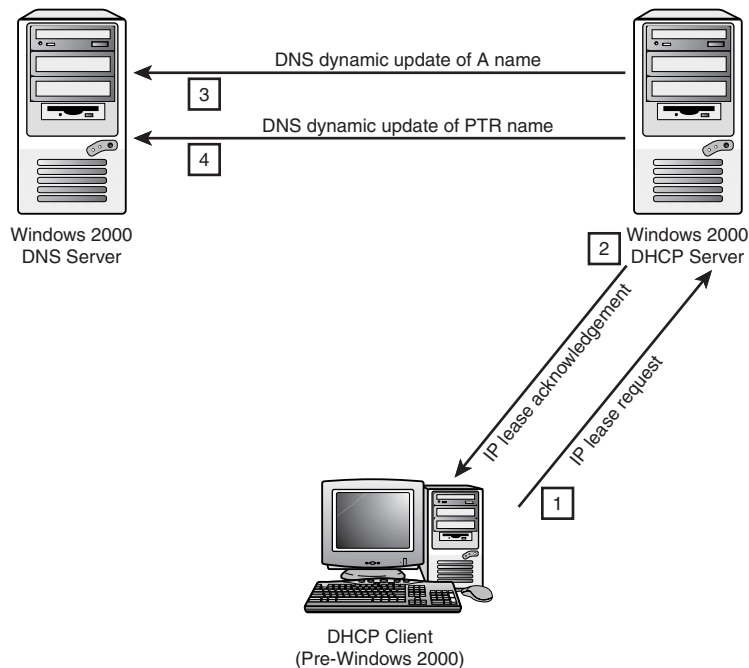


**FIGURE 3.11**

*Windows 2000 DHCP clients and dynamic update.*

1. The Windows 2000 DHCP client makes an IP lease request.
2. The DHCP server grants an IP lease.
3. The Windows 2000 DHCP client updates its forward (A) record with the DNS server.
4. The DHCP server updates the DNS reverse (PTR) record for the client using the dynamic update protocol.

Earlier versions of Windows DHCP clients do not interact directly with DNS servers that perform dynamic updates. Figure 3.12 shows how the forward and reverse lookup names are updated by a DHCP server if the service has been configured for this.



**FIGURE 3.12**

*Older DHCP clients and dynamic updates.*

1. The DHCP client makes an IP lease request.
2. The DHCP server grants an IP lease.
3. The DHCP server automatically generates the client's FQDN by appending the domain name defined for the scope to the client name obtained from the DHCPRequest message sent by the older client.

4. Using the dynamic update protocol, the DHCP server updates the DNS forward (A) record for the client.
5. Using the dynamic update protocol, the DHCP server updates the DNS reverse (PTR) record for the client.

## Configuring DHCP

You can define server- and scope-specific configuration settings to identify routers and set DHCP client configurations.

### DHCP Scopes

A DHCP scope identifies the possible IP addresses for DHCP clients on a specific subnet. Scopes define a range in which DHCP services are to be offered and allow the server to identify configuration parameters (such as DNS and WINS information if necessary) provided to DHCP clients. A scope must be defined before DHCP clients can acquire an IP address.

To configure a DHCP server scope, perform the following steps:

1. In the DHCP snap-in, right-click the server icon, select New Scope, and click Next.
2. In the scope window, input the scope name and detail information, and then click Next.
3. In the IP Address Range window, enter an address range and subnet mask for the scope, and click Next. A subnet mask, based on the address class, will be entered by default. You can modify this subnet mask, or click Next.

#### NOTE

You cannot modify the subnet mask after the scope has been created. Make sure that it is correct before continuing.

4. In the Add Exclusions window, input a range of addresses that are currently statically assigned, or are scheduled to be. Click Next.
5. In the Lease Duration window, set the lease time of the address. The default lease time is set to eight days (up from three days in NT4). Click Next.
6. In the Configure Your DHCP Options window, you have the option of entering settings including router, DNS, and WINS. You can elect to perform this configuration later by selecting the appropriate radio button. Click Next.
7. If you elect to configure the options, input the appropriate value into the subsequent windows for routers, domain name, and DNS servers and, finally, WINS servers.
8. In the Activate Scope window, click the radio button to activate the scope.
9. Click Next, and then click Finish.

## Address Pools

After a DHCP scope is defined and exclusion ranges are applied, the remaining addresses form an available address pool within the scope. The Address Pool folder in the Scope folder contains the various address pools.

## Exclusion Ranges

An exclusion range is a sequence of IP addresses within a scope that are excluded from assignment by the DHCP service.

## Reservations

Reservations allow permanent address lease assignment to a host (such as a printer or dedicated engineering PC). The DHCP server reserves the address in its pool, ensuring that the host will always use the same IP address. Reservations ensure that the DHCP service does not duplicate or reassign the IP address. Reservations can be useful for network devices such as UNIX workstations, print servers, printers, and so on.

Each reservation requires a Media Access Control (MAC) from the network interface card (NIC) for the DHCP client.

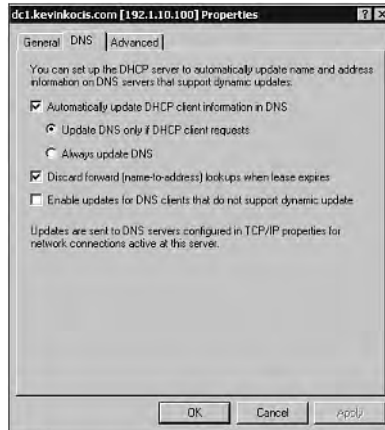
## Setting DDNS Update on a Scope

To configure DHCP to dynamically update DNS, you need to modify DHCP in the MMC DHCP snap-in.

Perform the following steps:

1. In the DHCP directory, right-click the specific scope folder and select Properties.
2. Select the DNS tab and check the Automatically Update DHCP Client Information in DNS box (see Figure 3.13).
3. If you want to make the update mandatory, click the Always Update DNS radio button. Otherwise, select the Update DNS Only If DHCP Client Requests button.

If you decide to let the client choose, the option is located in the Windows 2000 client's Advanced TCP/IP settings under the DNS tab (check the Register This Connection's Address in DNS box).

**FIGURE 3.13**

*Configuring the DHCP server for Dynamic DNS.*

## Summary

This chapter addressed the issues of managing more scalable domain structures and how to create new domains. Windows 2000 introduced two new domain models: mixed mode and native mode. The chapter also looked at the new trust relationship with Active Directory, which is composed of two-way transitive trusts to other domain controllers, while still having the option of setting exclusive one-way trusts. Lastly, it discussed DNS in Windows 2000 and other third-party versions. DNS will be a critical, and controversial, issue for many enterprises looking to implement Active Directory. Chapter 4, “Managing Users, Groups, and Computers,” looks into managing users, computers, and DHCP.



# Managing Users, Groups, and Computers

CHAPTER

# 4

## IN THIS CHAPTER

- Object Management Fundamentals 108
- Managing Users 110
- Managing User Profiles and Home Directories 117
- Managing Groups 120
- Managing Computer Accounts 133

A significant portion of Active Directory administration involves managing users, groups, and computers. Active Directory plays several major roles in providing security.

Active Directory confirms the identity of any user logging in to a domain through user authentication and allows users to access resources. A key feature of Windows 2000's user authentication process is its single sign-on capability, which provides access to multiple network resources through a single user logon.

## Object Management Fundamentals

Although we'll pursue object security in greater detail in Chapter 5, "Active Directory Security," it is important to briefly address some security features associated with users, groups, and computers.

Upon authentication, the user's access privileges are determined by the rights assigned to the user's account and by the access control permissions attached to the objects being accessed. User rights, which are assigned to users and groups, include both privileges and logon rights.

Access control permissions (such as Read, Write, Full Control, or No Access) are attached to all objects in Active Directory and can now be as granular as the property of each object. To understand the concepts of security a little better, let's take a brief look at some definitions:

- Access token. Each time a user logs on, Windows 2000 creates an access token. The access token is a representation of the user account and contains the following:
  - Security identifier (SID). A SID is a code that uniquely identifies a specific user, group, or computer to the Windows 2000 security system. A user's own SID is always attached to the user's access token. When the user is added to a group, a group SID is also attached to the user's access token.
    - Individual SID. Security identifier (SID) representing the logged-on user.
    - Group SIDs. SIDs representing the logged-on user's group memberships.
  - User Rights. Privileges (associated with each SID) granted to the user or to membership groups.

When the user attempts to access an object, Windows 2000 compares each SID in the user's access token to entries in an object's Discretionary Access Control List (DACL) to determine whether the user has permission to access the object and the type of access. Sometimes user rights in the user's token may override the permissions listed in the DACL, and access may be granted that way.

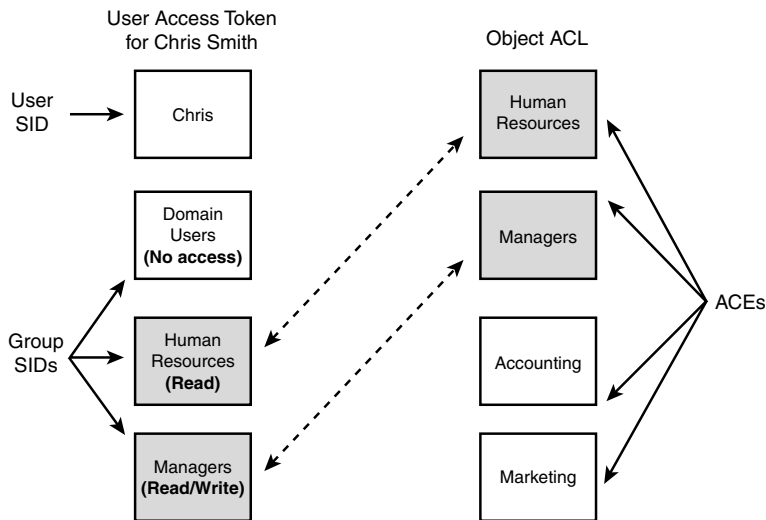


**NOTE**

Access tokens are not updated until the next logon, which means that if a user's group membership is changed, the user must log off and log on before the access token is updated.

- Access Control List (ACL). Each Active Directory object has two associated ACLs:
  - DACL. The Discretionary Access Control List (DACL) is a list of user accounts, groups, and computers that are allowed/denied object access.
  - SACL. The System Access Control List (SACL) defines which events are audited for a user or group.
- Access Control Entry (ACE). A DACL or SACL consists of a list of Access Control Entries (ACEs), where each ACE lists the permissions granted or denied to the users, groups, or computers listed in the DACL or SACL. An ACE contains a SID with a permission (such as Read or Write access).

Figure 4.1 illustrates the process of a user's access token allowing access to an object.

**FIGURE 4.1**

*User access token to an object in Active Directory.*

When the user, Chris, requests access to the human resources file object, Windows 2000 compares each SID in Chris's access token to each ACE in the DACL to verify whether access is explicitly denied to Chris or to any of his membership groups. Then it verifies whether the requested access is permitted. Windows repeats these steps until it encounters a No Access or until it has collected all the necessary permissions to grant the requested access. If the DACL does not specifically allow permission for each requested access, access is denied.

In Figure 4.1, the user authentication creates an access token for the user, which contains the user's primary SID, together with the SIDs of the user's group memberships. This user is authorized to access the human resources file.

## Managing Users

In Active Directory you can add, disable, reset, rename or delete user and computer accounts using the Active Directory Users and Computers tool. The following sections describe these processes further.

### User Accounts

A user requires a user account to log on to a computer or to a domain. The account identifies the user, and Active Directory uses this identity for user authentication. Subsequently, the user is given access to resources.

User accounts can also be used as service accounts for some applications, such as backup programs, Microsoft SQL, and Microsoft Exchange. A service can be configured to log on as a user account, and it is then granted access to resources via that user account.

### Predefined User Accounts

Active Directory provides the following two predefined user accounts:

- Administrator account
- Guest account

These two accounts allow you to log on locally to a Windows 2000 computer and access local resources, and are designed primarily for initial logon and configuration of a local computer. Obviously, the Administrator account has full control over the local machine. The Guest account is designed for temporary and restricted access where a regular account may not be necessary. Additional local accounts can be added, but are not predefined.

**NOTE**

The Guest account is disabled by default, and you must enable it to allow unrestricted access to the computer. This is not recommended, however.

## Adding and Deleting Users

You can manage users from an MMC console or Active Directory Users and Computers found in the Administrative Tools folder under the Start, Programs menu on the taskbar.

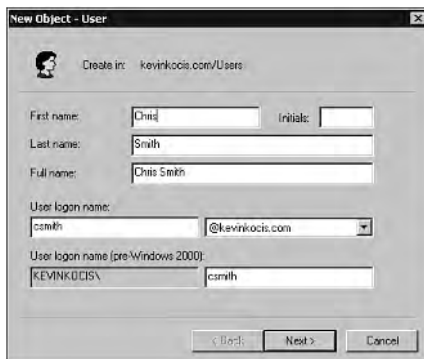
To add a user account, perform the following steps:

1. Open Active Directory Users and Computers.
2. In the console tree, expand the domain node if necessary.
3. In the details pane, right-click the container where you want to add the user, point to New, and then click User.
4. On the New Object-User window, enter the name information into the appropriate fields.
5. In the Account tab, in the User Logon Name field, type the name that the user will log on with and, from the drop-down list, click the User Principal Name (UPN) suffix that must be appended to the user logon name (following the @ symbol).

If the user will use a different name to log on from computers running Windows NT, Windows 98, or Windows 95, change the user logon name as it appears in User Logon Name (pre-Windows 2000) to the different name. You can also add multiple UPN suffixes. (See the section “Adding User Principal Name (UPN) Suffixes” later in this chapter.)

6. Click Next.
7. In the Password and Confirm Password boxes, type the user’s password.
8. Select the appropriate password options.
9. Click Next, and then Finish. Note an example of this in Figure 4.2.

You can also click the new user icon on the toolbar to add a new user. After creating the user account, you can edit the user account properties to enter additional user account information. Another method for creating users is to copy any previously created Windows 2000 user accounts. Using this procedure, you can create template accounts to allow for convenient creation of new user accounts.

**FIGURE 4.2**

*Creating a new user object in Active Directory.*

To copy a user account, perform the following steps:

1. Open Active Directory Users and Computers.
2. In the console tree, click Active Directory Users and Computers, then the domain, and then the folder containing the user account (typically Users).
3. In the details pane, right-click the user account that you want to copy, and then click Copy.
4. Enter the name information into the appropriate fields.
5. Select the UPN suffix in the User Logon Name field, and click Next.
6. In the Password and Confirm Password boxes, type the user's password.
7. Select the appropriate password parameters.
8. If the user account from which the new user account was copied was disabled, clear the Account Is Disabled checkbox to enable the new account.
9. Click Next, and then click Finish.

**NOTE**

The User Account Copy function is available only in Active Directory Users and Computers. Local user accounts created on non-domain controllers may not be copied.

## Deleting User Accounts

Deleting a user account is a dramatic step. This is a permanent process and should be avoided unless you are absolutely positive there is no chance the user account will be needed in the future (for example, a person is terminated for inappropriate actions). Many companies work with interns or have people leave the company only to return at a later time. Disabling the user accounts is a better option for these situations.

However, if you need to delete a user account, follow these steps:

1. Open Active Directory Users and Computers.
2. In the console tree, click Active Directory Users and Computers, then the domain, and then the folder containing the user account (typically Users).
3. Right-click the user account, and then click Delete.
4. Click yes in the Active Directory dialog box to confirm the deletion of the account.

### NOTE

After a user account (and corresponding SID) has been deleted, all permissions and memberships associated with that user account (SID) are deleted. The security descriptor for each account is unique. A new user account with the same name as a previously deleted user account cannot inherit the permissions and memberships of the previously associated with the deleted account (SID). All permissions and memberships must be manually re-created for the new account (SID) with the same name.

## Modifying Users

To modify user account properties, perform the following steps:

1. Open Active Directory Users and Computers.
2. In the console tree, click Active Directory Users and Computers, then the domain, and then the folder containing the user account (typically Users).
3. Right-click the user account, and then click Properties.
4. Edit the desired properties, and click OK.

## Rename

To rename a user account, perform the following steps:

1. Open Active Directory Users and Computers.
2. In the console tree, click Active Directory Users and Computers, then the domain, and then the folder containing the user account (typically Users).

3. In the details pane, right-click the user account and then click Rename.
4. Enter the new name, and click Enter.
5. In the Rename User window, enter the new name information.
6. In the User Logon Name box, type the name that the user will log on with and, from the drop-down list, click the UPN suffix for the new name.
7. Click OK.

As mentioned earlier, if the user will use a different name to log on from computers running down-level Windows versions, type the different name in pre-Windows 2000 logon name.

## Disabling and Enabling User Accounts

Enabling and disabling accounts is a security precaution depending on the situation. If an account has been disabled, you can easily enable it again. User accounts can be disabled rather than deleted as a security measure to temporarily prevent someone from logging on.

To disable a user account, follow these steps:

1. Open Active Directory Users and Computers.
2. In the console tree, click Active Directory Users and Computers, then the domain, and then the folder containing the user account (typically Users).
3. In the details pane, right-click the user.
4. Click Disable Account, and click OK.

You can create disabled user accounts with common group memberships. This way, disabled user accounts can serve as account templates for easier account creation.

To enable a disabled user account, perform the following steps:

1. Open Active Directory Users and Computers.
2. In the console tree, click Active Directory Users and Computers, then the domain, and then the folder containing the user account (typically Users).
3. In the details pane, right-click the user, click Enable Account, and then click OK.

## Resetting Passwords

To reset a user password, perform the following steps:

1. Open Active Directory Users and Computers.
2. In the console tree, click Active Directory Users and Computers, then the domain, and then the folder containing the user account (typically Users).
3. In the details pane, right-click the user whose password you want to reset, and then click Reset Password.

4. In the Reset Password box, type and confirm the password.
5. If you want to require the user to change this password at the next logon process, select the User Must Change Password at Next Logon check box.
6. Click OK twice in the subsequent windows.

**NOTE**

Any services that are authenticated with a user account must be reset if the password for the service's user account is changed.

## Changing a User's Primary Group

The user's primary group applies only to users who log on to the network through Services for Macintosh or to users who run POSIX-compliant applications. Unless you are using these services, there is no need to change the primary group from Domain Users, which is the default value.

Setting the user's primary group membership to a value other than Domain Users might adversely impact performance because all users in the domain are members of Domain Users. If the user's primary group is set to another group, it might cause the group membership to exceed the supported maximum number of members.

To change a user's primary group, perform the following steps:

1. Open Active Directory Users and Computers.
2. In the console tree, click Active Directory Users and Computers, then the domain, and then the folder containing the user account (typically Users).
3. In the details pane, right-click the user you want change and then click Properties.
4. On the Member Of tab, click the group that you want to set as the user's primary group; click Set Primary Group; and then click OK.

## Adding User Principal Name (UPN) Suffixes

To add UPN suffixes, perform the following steps:

1. Open Active Directory Domains and Trusts.
2. In the console tree, right-click Active Directory Domains and Trusts, and then click Properties.
3. On the UPN Suffixes tab, type an alternative UPN suffix for the domain, and then click Add.
4. Repeat step 3 to add additional alternative user UPN suffixes, and then click OK.

## Locating Users

To find a user account, perform the following steps:

1. Open Active Directory Users and Computers.
2. If you want to search the entire domain, in the console tree, right-click the domain node, and then click Find. If you know which organizational unit the user is in, right-click the organizational unit in the console tree and then click Find.
3. In the Name box, type the name of the user you want to find.
4. Click Find Now.

### NOTE

There are many search options in the In drop-down menu. If you are not certain of the resource's exact location, you can search the entire directory, which scans the Global Catalog for Active Directory.

You can also click the find object icon on the toolbar. Click the Advanced tab for more specific search options.

## Moving User Accounts

To move a user account, perform the following steps:

1. Open Active Directory Users and Computers.
2. In the console tree, click Active Directory Users and Computers, then the domain, and then Users.  
Or click the folder that contains the desired user account.
3. In the details pane, right-click the user you want to move, and then click Move.
4. In the Move dialog box, click the folder to which you want to move the user account, and then click OK.

### NOTE

Active Directory Users and Computers cannot move user accounts between domains. To move a user account between domains, use `movetree.exe`, one of the Active Directory support tools. See the glossary for more information on this tool.



## Managing User Profiles and Home Directories

Similar to Windows NT 4.0, each user has a respective profile that consists of folders and data storing the user's desktop environment, data, mapped network connections, and application settings. Home directories are used to store user data on a local file server. This section addresses the different profiles and how to create and manage them.

### Advantages of User Profiles

User profiles are created automatically when the user first logs on to the local computer. Profiles enable the user to customize desktop settings and files, such as screen savers, backgrounds, and shortcuts to various files and applications. Because each user profile is saved separately, multiple users can log in to the same computer and receive their customized profiles.

### Profile Types

Three types of user profiles exist in Windows 2000. They are

- Local
- Roaming
- Mandatory

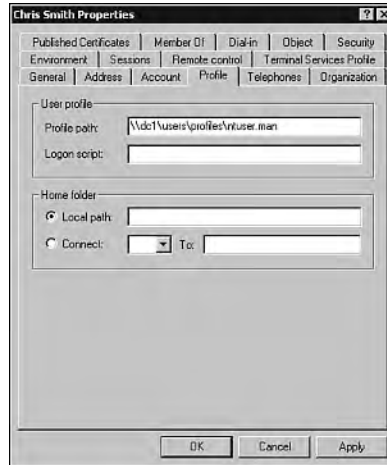
Local profiles are created on the local machine when the user first logs on. These profiles are stored locally on the computer's hard drive. For Windows 2000 clients, these are stored in the C:\Documents and Settings\<username> folder. Each time a user logs in to this particular machine, his respective profile is retrieved. Local profiles apply to that particular machine only and do not follow the user when he logs on to a different computer.

Roaming profiles are created by administrators and stored on network servers. This profile "roams" with the user, regardless of the computer being used. When the user first logs on to a different computer, the entire profile is copied locally. Any changes made to the roaming user profile are saved up to the server. Any updates are copied down to the different computers where the user's profile already exists when the user logs on.

#### NOTE

From an administrative standpoint, you should store roaming profiles on a member server as opposed to a domain controller. This will improve logon performance on your network.

To create a roaming user profile, you need to first configure a shared folder using the standard UNC \\server\share nomenclature (for example, \\dc1\profiles). Then, in the user's properties, on the profile tab, enter the profile path using the variable %username% as displayed in Figure 4.3.



**FIGURE 4.3**

*Configuring roaming user profiles.*

Mandatory profiles, as the name implies, are profiles that are enforced by the administrator. As with roaming profiles, mandatory profiles are downloaded from the network server. Even though they can be modified locally by the user, changes are not saved to either the server or local computer.

The profiles are made mandatory by changing one of the hidden files in the user's profile. By renaming the NTUSER.DAT file to NTUSER.MAN, the file is made read-only. You can create a mandatory profile template for use with multiple users (for example, \\dc1\profiles\users\ntuser.man).

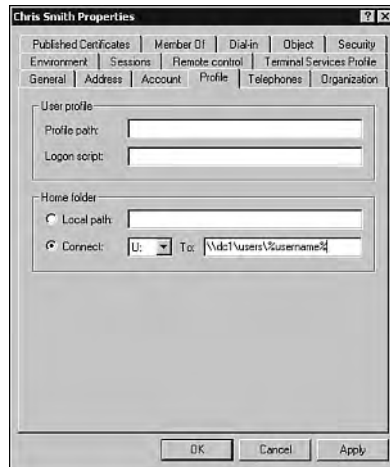
## NOTE

If user machines are Windows NT 4.0 or 2000, you do not need to include the file portion of the path (ntuser.man). If Windows NT 3.1 exists in your environment, you must include the filename in the profile path.

## Advantages of Home Directories

With home directories, you can standardize local mapped drives to a user's data repository on the network file server. By doing this, users can access their home directories from any computer on the network. Because the directories are stored on a file server (which should be backed up regularly), the backup is centralized and eases the impact of a client computer failure. Home directories can be configured for any computer running a Microsoft operating system (from MS-DOS to Windows 2000). Unfortunately for heterogeneous environments, home directories cannot be configured.

You can configure home directories in the same manner as profiles; that is, by creating a shared folder and path. This information is placed in the user's properties, under the profile tab, in the Home folder section as shown in Figure 4.4.



**FIGURE 4.4**

*Mapping a home directory folder path.*

### NOTE

Be sure to use a higher letter in the alphabet for your drive selection that will not conflict with locally configured physical or logical drives, such as removable media drives.

## Managing Groups

Groups are Active Directory (or local computer) objects that can contain users, contacts, computers, and other groups. Groups are created in domains using the Active Directory Users and Computers tool. You can create groups in any domain, organizational unit, or Container class object (such as the default Users container). Like user and computer accounts, groups are Windows 2000 security principals, which means that they have SIDs assigned to them when they are created.

Planning group strategies is an essential part of deploying Active Directory. You should understand how the number of domains and their type affect group strategy.

### NOTE

Both mixed-mode and native-mode domains can contain Windows NT 4.0 member servers and Windows NT and Windows 9.x clients.

## Group Types

Windows 2000 Active Directory has two kinds of groups:

- Distribution groups
- Security groups

Although this section is primarily about the role groups play in security, distribution groups are also briefly described to clarify the difference between the two group types. The next two sections describe the characteristics of distribution and security groups.

### Distribution Groups

Distribution groups serve only one purpose: to create email distribution lists. You use distribution groups with email applications, such as Microsoft Exchange, to send email to the members of the group.

Distribution groups have no security purpose. They cannot have permissions assigned to them.

Distribution groups can be used for bulk mailing and for universal groups, even in a mixed-mode domain.

### Security Groups

Active Directory security groups permit you to organize users and other domain objects into groups for easy administration of access permissions.

Security groups let you assign the identical permissions to large numbers of users at the same time, ensuring consistent permissions among all group members. You can add and remove users as necessary, and the Access Control Lists change infrequently. Changing a permission for the group affects all users in the group.

Active Directory provides several predefined security groups, and we'll talk about creating your personal security groups in the following section.

### Security Group Types

In Active Directory, security groups manage user and computer access to shared resources, as well as filter Group Policy settings.

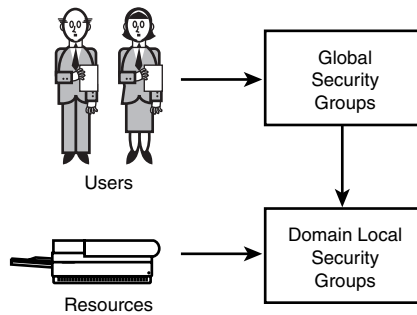
In Active Directory, you gather users, computers, and other groups into security groups and then assign appropriate permissions (such as printer or file access) to the security group. When you add users to an existing group, they automatically obtain the rights and permissions already assigned to that group.

As explained earlier, an access token is an object that contains security information for a logon session. An access token is created when a user logs on, and every process performed by the user has a copy of the token. The token identifies the user, his security group membership, and privileges.

When implementing an administration strategy for security groups, keep the following general guidelines in mind:

- Small organizations. Some small organizations with a Windows 2000 native-mode forest will choose to use security groups with universal scope to manage all their group needs. For organizations that expect to grow, two alternative strategies are available:
  - Use universal groups initially and then convert to the global/local pattern as suggested in the medium to large organization scenario discussed next.
  - Some growing small organizations will initially implement the global/local pattern used by larger organizations. Remember that universal security groups, and their group memberships, are listed in the global catalog database, and frequent changes might impact replication traffic. In this situation, follow the guidelines for medium to large organizations.
- Medium to large organizations. Experience shows that using the approach described next will help you achieve maximum flexibility, scalability, and ease of administration when managing security groups.
  - Put users with similar access needs into security groups with global scope.
  - Put users into security groups with domain local scope.
  - Put a global group into any domain local group in the forest (particularly for multi-domain environments), to grant access to the global group.

- Assign permissions for accessing resources to the domain local groups that contain them.
- Delegate administration of groups to the appropriate manager or group leader. Note the strategy in Figure 4.5.



**FIGURE 4.5**

*The administration strategy for security groups.*

Active Directory supports three types of security groups, differentiated by scope:

- Domain local
- Global
- Universal

Although not supported in Active Directory, local groups may exist on non-domain controller computers. We will look at these accounts in the local groups section later in the chapter.

Domain local groups are best used for granting access rights and permissions to resources located on any computer in the domain.

Global groups are used for combining users who share a common access profile based on job function or business role. Global groups are used when group membership is expected to change frequently.

Universal groups are used in larger, multi-domain organizations where similar access is required in multiple domains.

## NOTE

As previously mentioned, the Global Catalog stores the names of universal groups and the names of all the members of those universal groups. Because of this, you should use global groups as members of universal groups to reduce overall replication traffic from changes to universal group membership.

Universal groups are commonly used only in multiple domain trees. Active Directory domains must be in native mode to use universal security groups. A domain model that has only a single domain does not require universal groups.

Local groups are security groups that are created only on non-domain controllers and are not recognized elsewhere in the domain. They are created with the Local Users and Groups node of the Computer Management snap-in. This node is not available on domain controllers.

Group membership should be limited to 5,000 members, regardless of whether the groups are security groups or distribution groups. This number is not enforced but should be considered a hard limit. The design specification of Active Directory guarantees replication of up to 5,000 members but no more. In instances in which having more than 5,000 members in groups is required, use nested groups to bring down the number of objects in single group membership (as mentioned earlier in this section).

Let’s take a closer look at the different kinds of group scope.

### Domain Local Groups

Domain local groups, a new feature of the Windows 2000 Active Directory, have the features shown in Table 4.1.

**TABLE 4.1** Domain Local Group Features

<b>Mode</b>	Domain local groups are available in both native-mode and mixed-mode domains.
<b>Membership</b>	Like local groups, domain local groups can have members from anywhere in the forest, from trusted domains in other forests, and from trusted down-level domains.
<b>Permissions</b>	A domain local group has domain-wide scope; that is, it can be used to grant resource permissions on any Windows 2000 machine within the domain in which it exists (but not beyond its domain).

Domain local groups help you define and manage access to resources within a single domain.

To take advantage of domain local groups, perform the following steps:

1. Create a group with domain local scope, and assign it permission to access the resource.
2. Put user accounts into a group with global scope, and add (nest) this global group into a domain local group.

When you want to give new or other users access to the resource, you can add them to the global group that is a member of the domain local group that has permission to access the resource.

Using domain local groups in this way provides the following benefits:

- Membership of the domain local group is controlled by the administrator of the specified domain where the group is located, regardless of whether members of the group are from another domain.
- Because a domain local group is associated with an access token built when a member of that group authenticates to a resource in that domain, unnecessary network traffic is avoided. Less network bandwidth is required to validate one global group containing hundreds of users than to validate hundreds of individual users. If you were to assign a global group permission to access the resource, the global group could end up in a user's token anywhere in the forest, causing unnecessary network traffic.

One advantage of domain local groups is that membership is not published to the global catalog server (GC) and therefore does not affect GC replication. Another advantage is that Outlook clients can view full user membership if they are located in the same domain as the group (however, this does not apply to Outlook users in other domains). A final advantage of domain local groups is that they can consist of users from any domain, although domain local permissions cannot be assigned to resources in other domains. Also, their expansion must take place on a local domain controller.

### Global Groups

Global groups, effectively the same as Windows NT global groups, have the features shown in Table 4.2.

**TABLE 4.2** Global Group Features

<b>Mode</b>	Global groups exist in both mixed-mode and native-mode domains.
<b>Membership</b>	Global groups can have members from within their own domain only.
<b>Permissions</b>	Although a global group is limited to domain-wide scope as far as membership goes, it can be made a member of machine or domain local groups or granted permissions in any domain (including trusting domains in other forests and down-level domains with which a trust relationship exists). Groups with global scope can be put into other groups in any trusting domain.

Groups with global scope help you manage directory objects that require daily maintenance, such as user and computer accounts.

Use global groups to organize users or computers that are in the same domain and share the same responsibility, organizational role, or function. Managers and RAS Servers are examples



of possible global groups. Because members of global groups typically need to access the same resources, you should make these global groups members of local or domain local groups, which are listed on the DACL of needed resources.

Just like domain local groups, global groups membership is not published to the global catalog server (GC) and therefore does not affect GC replication. However, the global group definition (not the membership) is replicated to the global catalog. The impact of this function from a replication standpoint is minimal (about 180 bytes) space in the global catalog.

Also, like domain local groups, global groups can contain account objects only from the same domain. A similar advantage is that Outlook clients can view full user membership if they are located in the same domain as the group (however, this does not apply to Outlook users in other domains).

The big advantage of global groups is that permissions can be assigned to resources in any domain in the enterprise.

Universal Groups

Universal groups are the most flexible type of group in Active Directory. When your organization is in native mode, universal groups can be either distribution groups or security groups. If your organization is in mixed mode, universal groups can be distribution groups only.

Use universal groups when there is a requirement for both global membership and global usage. Universal groups and their membership are replicated to the global catalog. Universal group membership consists of global groups, which reduces GC impact. Universal group membership tends to contain many user accounts. Size and replication impact increases as a result.

Universal groups, a new feature of the Windows 2000 operating system, have the following features seen in Table 4.3:

TABLE 4.3 Universal Group Features

Mode	Universal groups are available only in native-mode domains.
Membership	Universal groups can have members from any Windows 2000 domain in the forest. (Universal groups can contain members from mixed-mode domains in the same forest, but this is not recommended. Members from such domains cannot have the universal group's SID added to their access token because universal groups are not available in mixed-mode domains. Therefore, troubleshooting access problems would be difficult.)
Permissions	Universal groups can be granted permissions in any domain, including in domains in other forests with which a trust relationship exists.

A small organization can use universal groups to implement a relatively simple group structure. If you choose to use groups with universal scope in a multi-domain environment, these groups can help you represent and consolidate groups that span domains.

Although few organizations will choose to implement this level of complexity, you can add user accounts to groups with global scope, nest these groups within groups having universal scope, and then make the universal group a member of a domain or machine local group having access to resources. This way, membership changes in the groups having global scope have no impact on the groups with universal scope.

One advantage of universal groups is that their membership can consist of any objects in the forest, and enterprise Outlook users can view full membership. They can also be used in mixed-mode domains when type is set to Distribution and not Security—because universal security groups can be created only when a domain is in native mode.

Unlike domain local and global groups, modifications to universal groups cause replication to the global catalog servers.

**Local Groups**

Local groups are sometimes referred to as machine local groups to contrast them with domain local groups. Local groups have the following features:

**TABLE 4.4** Local Group Features

<b>Mode</b>	Local groups are the only type of local group available in a Windows 2000 mixed-mode domain. In the case of Windows 2000 native-mode domains, only Built-in groups have local scope.
<b>Membership</b>	Local groups can have members from anywhere in the forest, from trusted domains in other forests, and from trusted down-level domains.
<b>Permissions</b>	A local group has only machine-wide scope; that is, it can be used to grant resource permissions only on the machine on which it exists.

**Group Scope and Replication Traffic**

Groups having universal scope and all of their members are listed in the global catalog. Whenever one member of a group with universal scope changes, the entire group membership must be replicated to all global catalogs in the domain tree or forest. For this reason, you should use groups with universal scope in situations where the membership of the group does not change frequently.

Groups having global or domain local scope are also listed in the global catalog, but their individual members are not listed. Using these groups thus reduces the size of the global catalog and reduces the replication traffic needed to keep the global catalog up-to-date. Therefore, use groups with global or domain local scope if the group membership changes frequently.

## How Domain Mode Affects Groups

As explained previously, a mixed-mode domain typically has one or more Windows NT Server 4.0 domain controllers in addition to Windows 2000 domain controllers, although it can have only Windows 2000 domain controllers. A native-mode domain can have only Windows 2000 Server domain controllers. Both mixed-mode and native-mode domains can include Windows NT 4.0 member servers and Windows NT and Windows 9.x clients.

In a native-mode domain, you can convert a security group to a distribution group and vice versa. You cannot convert either group to the other in a mixed-mode domain. A Windows NT domain controller cannot handle group type conversion because it sees only security-enabled groups.

Distribution groups are not affected by mode because distribution group membership is not enumerated at logon. If a process needs to know the composition of the group, it has to ask an Active Directory server, which, by definition, is a Windows 2000 domain controller.

When a user logs on to a domain, the user's security group membership is resolved on the domain controller that handles the logon. In mixed mode, if a Windows NT 4.0 domain controller handles the logon, then it must be able to enumerate the members of the security groups to which the user belongs. Thus, the behavior of security groups in a Windows 2000 domain running in mixed mode must match the behavior of security groups in Windows NT 4.0.

## Nesting Groups

You can also nest groups. Nesting groups is the process where you can add a group as a member of another group. Nesting groups makes it easier to manage users and can reduce network traffic caused by replication of group membership changes.

## Mode Governs Nesting Options

In Active Directory, group membership is limited to 5,000 users. Windows 2000 lets you get around this limitation by nesting groups to increase the effective number of members. Nesting also alleviates replication traffic caused by group membership changes.

Available nesting options depend on whether the domain is in native mode or mixed mode. The following list describes what can be contained in a group that exists in a native mode domain:

- Groups with universal scope can contain user accounts, computer accounts, other universal groups, and global groups from any trusted domain.
- Groups with global scope can contain user accounts from the same domain and other global groups from the same domain.

- Groups with domain local scope can contain user accounts, universal groups, and global groups from any trusted domain. They can also contain other domain local groups from within the same domain. (Typically, put user accounts into global groups, not into domain local groups, then put the global groups into domain local groups, and then assign access permissions to resources to the local groups.)

Security groups in a mixed-mode domain can contain only the following:

- Local groups can contain global groups and user accounts from trusted domains.

#### NOTE

You should not place users directly into local groups. Place them into global groups, put global groups into local groups, and then assign permissions to the local groups.

- Global groups can contain only user accounts.

## Modifying Groups

In addition to users and computers, membership in a particular group can include contacts and other groups.

## Creating and Deleting Groups

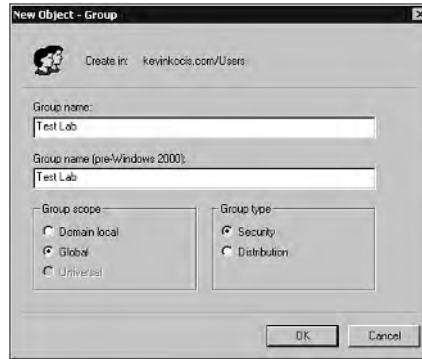
To add a group, perform the following steps:

1. Open Active Directory Users and Computers.
2. In the console tree, expand the domain node if necessary.
3. Right-click the folder in which you want to add the group, point to New, and then click Group.
4. Type the name of the new group. See Figure 4.6. By default, the name you type is also entered as the pre-Windows 2000 name of the new group.
5. Click the appropriate Group scope.
6. Click the appropriate Group type, and click OK.

You can also click the folder in which you want to add the group and then click the new group icon on the toolbar.

#### NOTE

If the domain in which you are creating the group is in mixed mode, you can only select security groups with domain local or global scopes.

**FIGURE 4.6**

*Creating a new group in Active Directory. Note that the Universal group type is not an option for this security group, indicating that this domain is in mixed mode.*

To delete a group, perform the following steps:

1. Open Active Directory Users and Computers.
2. In the console tree, expand the domain node if necessary.
3. Click the folder that contains the group.
4. In the details pane, right-click the group and then click Delete.
5. Click Yes to confirm the account deletion.

## Converting Group Type

Although changing the group type may not be a common process (with the contrasting attributes of security and distribution groups), you should know that this process is possible.

To convert a group to another group type, perform the following steps:

1. Open Active Directory Users and Computers.
2. In the console tree, expand the domain node if necessary.
3. Click the folder that contains the group.
4. In the details pane, right-click the group and then click Properties.
5. On the General tab, under Group type, click the group type.

## Changing Group Scope

When you create a new group, the default is a security group with global scope (regardless of the domain mode). Although you cannot change a group scope in a mixed-domain setting, in native mode you can change global and domain local groups to universal groups. This change usually applies to groups requiring access in multiple domains.

To change group scope, perform the following steps:

1. Open Active Directory Users and Computers.
2. In the console tree, expand the domain node if necessary.
3. Click the folder that contains the group.
4. In the details pane, right-click the group and then click Properties.
5. On the General tab, under Group scope, choose the new group scope, and click OK.

Local groups provided automatically by Windows 2000, such as Administrators and Account Operators, are located in the Built-in folder by default. Common global groups, such as Domain Admins and Domain Users, are located in the Users folder by default. New groups can be added or moved to any folder. It is recommended that they be located in an organizational unit folder.

## Locating Groups

To find a group, perform the following steps:

1. Open Active Directory Users and Computers.
2. In the console tree, right-click the domain node and then click Find.
3. On the Users, Contacts, and Groups tab, in the Name box, type the name of the group you want to find.
4. Click Find Now.

To find groups in which a user is a member, perform the following steps:

1. Open Active Directory Users and Computers.
2. In the console tree, expand the domain node if necessary.
3. Click the Users folder.
4. In the details pane, right-click a user account and then click Properties.
5. Click the Member Of tab.

## Editing Groups

To modify group properties, perform the following steps:

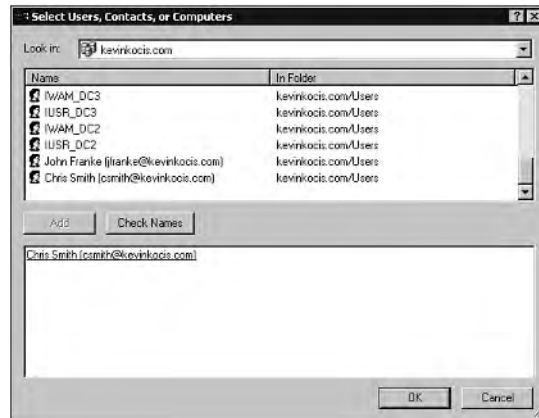
1. Open Active Directory Users and Computers.
2. In the console tree, expand the domain node if necessary.
3. Click the folder that contains the group.
4. In the details pane, right-click the group and then click Properties.

## Adding a Member to a Group

To add a member to a group, perform the following steps:

1. Open Active Directory Users and Computers.
2. In the console tree, expand the domain node if necessary.
3. Click the folder that contains the group to which you want to add a member.
4. In the details pane, right-click the group and then click Properties.
5. Click the Members tab, and then click Add.
6. Click Look In to display a list of domains from which users and computers can be added to the group, and then click the domain containing the users and computers you want to add.
7. Click the users and computers to be added, then click Add, and then click OK. See Figure 4.7.

You can also select the members to be added, click the add user icon on the toolbar, and then select the target group.



**FIGURE 4.7**

*Adding user object csmith to the Test Lab global security group.*

## Removing a Member from a Group

To remove a member from a group, perform the following steps:

1. Open Active Directory Users and Computers.
2. In the console tree, expand the domain node if necessary.
3. Click the folder that contains the group.

4. In the details pane, right-click the group, and then click Properties.
5. Click the Members tab.
6. Click the member(s) you want to delete, and then click Remove.
7. Click Yes to confirm the deletion of the account(s).

## Renaming a Group

To rename a group, perform the following steps:

1. Open Active Directory Users and Computers.
2. In the console tree, expand the domain node if necessary.
3. Click the folder in which the group is located.
4. In the details pane, right-click the group and then click Rename.
5. Type the new group name.
6. In the Rename Group Box, you may also change the pre-Windows 2000 group name.
7. Click Finish.

## Replication Conflicts

If administrators at two different domain controllers change group membership simultaneously, one of the changes might be lost. This situation can occur only if you are making group membership changes faster than the system can replicate them. When an administrator adds or removes members from a group, the entire group membership is replicated, not just the changed members. If two administrators change group membership on two different domain controllers and replication takes place on the second domain controller before the first domain controller completes replication, only one of the changes remains after the Active Directory resolves the replication conflict. The other change is lost. As a result, a user might unexpectedly retain or lose access to a resource.

One way to minimize this problem is to use nested groups. Create site-specific groups and make them members of a parent group that will be used to grant or deny access to a resource. Administrators in a site can then change the membership of a site-specific group and not lose changes as long as the membership of the site-specific group is not updated on multiple domain controllers faster than intrasite replication can complete. Also, if you delegate responsibility for group membership changes to one administrator per site, all changes will be made on a single domain controller, and no replication conflicts will occur.

In an Active Directory site, the amount of time it takes for a change to reach all the domain controllers increases as the number of domain controllers increases with a maximum latency of approximately three times the replicator notify pause interval. Generally, replication finishes



quickly within a single site. Replication between two or more Active Directory sites generally takes longer and is dependent on the replication schedule configured by the administrator as well as whether the administrator configures intersite replicator notifications.

To avoid this situation completely, make all group membership changes on a single domain controller. This prevents changes from being lost due to replication conflicts.

## Managing Computer Accounts

Because computer accounts are also security principals (as are user accounts), you can add, disable, reset, rename, and delete computer accounts in Active Directory in the same manner as user accounts; that is, with the Active Directory Users and Computers tool. The following sections further detail the computer account creation and modification processes.

### Creating Computer Accounts

To add a computer account, perform the following steps:

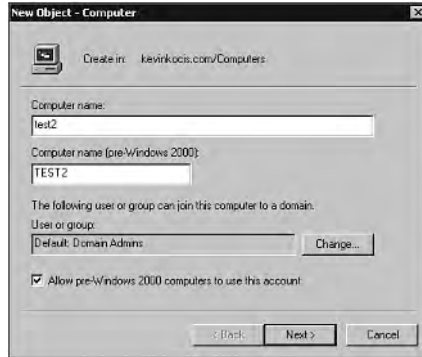
1. Open Active Directory Users and Computers.
2. In the console tree, expand the domain node if necessary.
3. Click the folder that contains the computer accounts.
4. Right-click Computers or the container in which you want to add the computer, point to New, and then click Computer.
5. Type the computer name, click Next, and then click OK (see Figure 4.8).

#### NOTE

Default settings enable only Domain Admins to add a computer account to a domain. Click Change to specify a different user or group that can add this computer to the domain.

To view or change the full computer name of a computer and the domain that a computer belongs to, right-click My Computer, click Properties, and then click the Network Identification tab.

There are two additional ways to give a user or group permission to add a computer to the domain: use a Group Policy object to assign the right Add computer user (discussed in Chapter 5, “Active Directory Security”), or allow them to create computer objects by assigning the user or group the permission Create computer objects in their organizational unit.

**FIGURE 4.8**

*Creating a new computer security object in Active Directory.*

To add a computer account to a group, perform the following steps:

1. Open Active Directory Users and Computers.
2. In the console tree, expand the domain node if necessary.
3. Click the folder that contains the computer accounts.
4. In the details pane, right-click the computer and then click Properties.
5. On the Member Of tab, click Add.
6. Click the group to which you want to add the computer click Add, and then click OK.

Or, to add the computer to more than one group, press the Ctrl key and click the groups to which you want to add the computer, and then click Add.

To add a computer to a group, you can also click the computer you want to add, click the add computer to group icon on the toolbar, and then select the target group.

Adding a computer to a group allows you to assign permissions to all the computer accounts in that group and to filter Group Policy settings on all accounts in that group.

To delete a computer account, perform the following steps:

1. Open Active Directory Users and Computers.
2. In the console tree, expand the domain node if necessary.
3. Click the folder that contains the computer accounts.
4. In the details pane, right-click the computer and then click Delete.
5. Click Yes to confirm the computer account deletion.

## Locating Computers

To find a computer account, perform the following steps:

1. Open Active Directory Users and Computers.
2. If you want to search the entire domain, in the console tree, right-click the domain node and then click Find.  
Or, if you know which organizational unit the computer is in, in the console tree, right-click the organizational unit and then click Find.
3. In the Find box, click Computers.
4. In the Name box, type the name of the computer you want to find.
5. To find only domain controllers, in Role, click Domain Controller.  
Or, to find only workstations and servers (not domain controllers), in Role, click Workstations and Servers.
6. Click Find Now.

The Advanced tab offers more powerful search options.

To find a computer, you can also click the search icon on the toolbar.

## Editing Computer Accounts

To manage a computer, perform the following steps:

1. Open Active Directory Users and Computers.
2. In the console tree, expand the domain node if necessary.
3. Click the folder that contains the computer accounts.
4. In the details pane, right-click the computer and then click Manage.

This starts the Computer Management snap-in, where you can administer local and remote computers.

## Modifying Computer Accounts

To modify computer account properties, perform the following steps:

1. Open Active Directory Users and Computers.
2. In the console tree, expand the domain node if necessary.
3. Click the folder that contains the computer accounts.
4. In the details pane, right-click the computer and then click Properties.

## Moving a Computer Account

To move a computer account, perform the following steps:

1. Open Active Directory Users and Computers.
2. In the console tree, expand the domain node if necessary.
3. Click the folder that contains the computer accounts.
4. In the details pane, right-click the computer and then click Move.
5. In the Move dialog box, click the domain node.
6. Click the folder to which you want to move the computer, and then click OK.

Active Directory Users and Computers cannot move computer accounts between domains. To move a computer account between domains, use `movetree.exe`, one of the Active Directory support tools.

## Resetting Computer Accounts

To reset a computer account, perform the following steps:

1. Open Active Directory Users and Computers.
2. In the console tree, expand the domain node if necessary.
3. Click the folder that contains the computer accounts.
4. In the details pane, right-click the computer and then click Reset Account.

Resetting a computer account breaks that computer's connection to the domain and requires it to rejoin the domain.

### NOTE

You cannot reset the computer account for a domain controller.

## Enabling and Disabling Computer Accounts

To disable a computer account, perform the following steps:

1. Open Active Directory Users and Computers.
2. In the console tree, expand the domain node if necessary.
3. Click the folder that contains the computer accounts.
4. In the details pane, right-click the desired computer and then click Disable Account.
5. Click Yes to confirm that no users will be able to log on from the disabled computer, and click OK.

To enable a computer account, perform the following steps:

1. Open Active Directory Users and Computers.
2. In the console tree, expand the domain node if necessary.
3. Click the folder that contains the computer accounts.
4. In the details pane, right-click the desired computer click Enable Account, and then click OK.

To allow a computer to use a different DNS name, perform the following steps:

1. Open Active Directory Users and Computers.
2. In the console tree, right-click Active Directory Users and Computers, and click Connect to Domain.
3. In Domain, type the domain name or click Browse to find the domain in which you want to enable computers to use different DNS names, and then click OK.
4. Right-click Active Directory Users and Computers, point to View, and then click Advanced Features.

#### NOTE

The Advanced Features enable security features that are hidden by default.

5. Right-click the name of the domain, and then click Properties.
6. On the Security tab, click Add, click the Self group, click Add again, and then click OK.
7. Click Advanced, click Self, and then click View/Edit.
8. On the Properties tab, in Apply Onto, click Computer Objects (see Figure 4.9).
9. Under Permissions, in the Properties tab, click Write dNSHostName, and then click the Allow check box.
10. Click OK three times to close all boxes and accept the changes.

#### NOTE

By modifying default security in this way, there is the possibility that a computer joined to the selected domain could be operated by a malicious user and might be able to advertise itself under a different name through the service principal name attribute.

This procedure also allows computers to have DNS host names longer than 15 bytes.



**FIGURE 4.9**  
*Allowing a computer to use a different DNS name.*

## Summary

In this chapter, we addressed managing users and groups, the different scopes (domain local, global, and universal) and types (distribution and security) associated with groups in Active Directory. We also addressed computer accounts as well as sharing folders and printers in Active Directory. In the next chapter, we'll take a closer look at Active Directory security.

# Active Directory Security

CHAPTER

# 5

## IN THIS CHAPTER

- The Active Directory Security Model 140
- Object-Oriented Security 155
- Active Directory Object Security 160
- Publishing Active Directory Resources 164

The Active Directory environment offers various new security features. Security in Active Directory involves the use of Access Control Lists (ACLs) and security identifiers (SIDs) to determine access to objects and the level of access. This chapter provides a closer look at the security features of Active Directory, such as Kerberos and IPsec, as well as access control security. This chapter also addresses publishing resources in Active Directory and the delegation and ownership of those processes.

## The Active Directory Security Model

The security model in Active Directory uses organizational units (OUs) to organize the namespace into subsets. The domain OUs contain objects, and each object can be granted or denied access to other objects in the Active Directory. This access can now be applied to the property level of an individual object. (This capability is new to Windows 2000.) Active Directory also introduces new authentication, network, and access control securities. These technologies include Kerberos, Public Key Infrastructure (PKI), and IPsec. Let's take a closer look at these security technologies.

### Authentication in Active Directory

The authentication process in Active Directory is perhaps the most significant security change from Windows NT 4.0 authentication. The process has been overhauled in an effort to become more standardized.

In Windows NT 4.0, Windows NT LAN Manager (NTLM) was the default authentication protocol for network authentication. While NTLM is still supported in Windows 2000, it is not the default. For Active Directory, Kerberos V5 (version 5) is the default protocol. Let's take a closer look at Kerberos.

### Kerberos Authentication

Kerberos is automatically installed when the Active Directory is installed on a Windows 2000 domain controller (DC). Kerberos is used for user logon authentication, as well as to support the transitive trusts in Windows 2000.

#### NOTE

NTLM is still supported in Windows 2000 for compatibility with down-level clients and servers, as well as with Windows 2000 standalone computer logons.



## Kerberos Components

As mentioned in the Note, the Kerberos name evolved from the three-part functionality of its protocol. The three parts are

- The client, which is the computer requesting a service
- The server, which is the computer providing the service
- The Key Distribution Center (KDC), which is the computer issuing the session key necessary for the client and server to communicate

The Kerberos authentication process is the result of an interaction among the client requesting a service, the server providing the service, and the KDC.

In Kerberos, a special key server—the KDC—distributes keys. For security purposes, the system actually requires the following three keys in order to provide secure interaction among client, server, and KDC:

- Session key—A temporary key generated by the KDC that encrypts messages between a client and a specific service running on a server computer.
- Client key—A key derived from the user's password that is used to distribute a session key to the client. Only the client and KDC know this key. The KDC uses the client key to encrypt a copy of the session ticket, which the client sends to the server to initiate the connection.
- Server key—A key known only by the server and the KDC. The KDC packages the information it wants the server to know (a copy of the session key and information on the client requesting the connection) and uses the server key to encrypt that package of information into the session ticket. It then encloses the session ticket into a message to the client, encrypted with the client key. The client extracts the session ticket and sends it to the server.

The following are some of the benefits of Kerberos over NTLM:

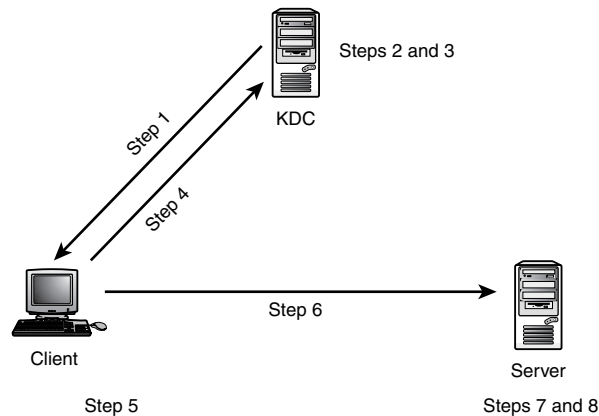
- Mutual authentication—With Kerberos, servers can identify clients, and vice versa. In NTLM, this was not the case—servers were capable of identifying clients, but not vice versa.
- Efficiency—In Kerberos, clients are authenticated only once and can reuse their credentials throughout the logon session.
- Delegated authentication—Kerberos allows a service to impersonate its client when connecting to other services. NTLM did not support this process.
- Interoperability—Although it is not the MIT-based Kerberos, Windows 2000 Kerberos can interoperate with other Kerberos servers and clients.

## A Little Kerberos History

Kerberos, which was developed at MIT, has been implemented widely on UNIX-based systems. Kerberos also provides a more network-based security environment for UNIX-like systems.

Kerberos is the Greek name for the legendary three-headed dog Cerberus, who guards the gate to the underworld. This name was chosen to correspond to the three parts of this protocol.

The Kerberos authentication process is quite complicated. To better understand Kerberos authentication, let's look at the process illustrated in Figure 5.1.



**FIGURE 5.1**

*The Kerberos authentication process.*

The Kerberos authentication process is as follows:

1. The client requests access to a server service.
2. The KDC creates the session key.
3. The KDC creates a ticket and encrypts it with the server's long-term key.
4. The KDC encloses the ticket in a response to the client that is encrypted with the client's long-term key.
5. The client decrypts the response with its long-term key and extracts the ticket.
6. The client sends an application request to the server (along with authenticator information encrypted in the session key) and a ticket encrypted with the server's long-term key.

7. The server decrypts the ticket with its long-term key and gets the session key.
8. The server uses the session key to decrypt the authenticator.

## Kerberos and Active Directory

In Active Directory, every domain controller is automatically configured as a KDC. You locate the Kerberos Key Distribution Center Service, which starts automatically, by choosing Start, Programs, Administrative Tools, Services.

### NOTE

Windows 2000 clients use the Kerberos logon, if possible. Pre-Windows 2000 clients do not support Kerberos and log on using NTLM authentication.

## Kerberos Preauthentication

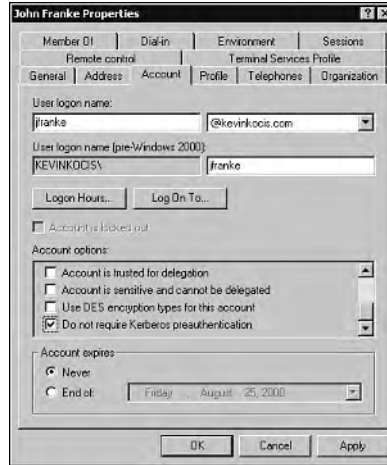
Kerberos is inherent to Windows 2000 computers, and no configuration is required to configure authentication. However, Microsoft Kerberos and other versions of Kerberos do differ. If your enterprise has already implemented Kerberos (for example, in your UNIX environment), you may need to disable a feature called Kerberos Preauthentication.

The Microsoft Kerberos protocol implements a feature known as *preauthentication*, in which the KDC performs a preliminary authentication before issuing a ticket granting ticket (TGT). From Microsoft's perspective, this process can assist with defending against password-guessing attacks.

However, not all Kerberos environments support Windows 2000 preauthentication. If your enterprise consists solely of Windows 2000 computers, there is no need to change the default setting. If your UNIX Kerberos clients (for example) are not able to successfully interoperate with Windows 2000 Kerberos, you need to disable preauthentication for those clients' accounts.

To disable Microsoft's Kerberos preauthentication, perform the following steps:

1. Open Active Directory Users and Computers.
2. Select the container or OU that holds the account for which you want to disable preauthentication (Users).
3. Right-click the Account icon in the display window and then select Properties.
4. On the Account tab of the account's properties, scroll through the Account Options list and select the Do Not Require Kerberos Preauthentication check box, as shown in Figure 5.2. Then click OK.

**FIGURE 5.2**

*Disabling Kerberos preauthentication for a non-Windows Kerberos client.*

## Setting Kerberos Policy

You can also set Kerberos policy in Active Directory. You can perform this task either through the Domain Security Policy tool or through Group Policy. Although group policy is addressed in the following chapter, the default security policy includes several Kerberos policy settings. You can configure Kerberos security policy through the Domain Security Policy tool or through Group Policy (covered in Chapter 6, “Administering Group Policy”).

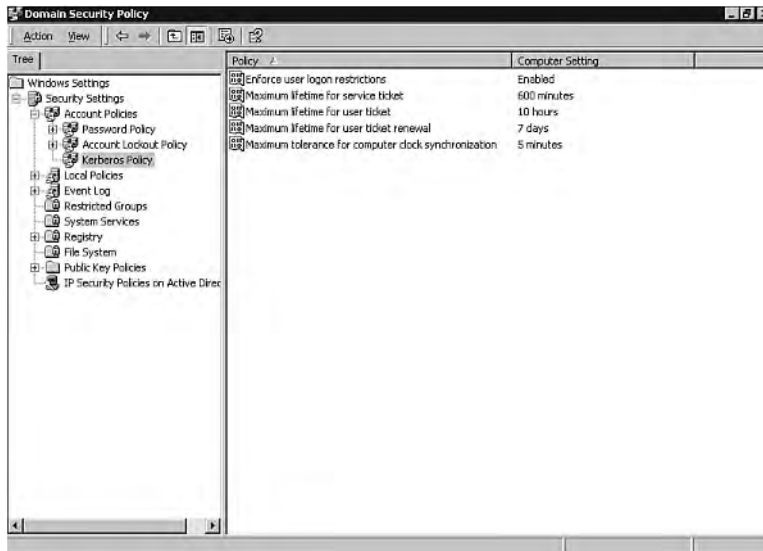
To configure the domain security policy, perform the following steps:

1. In Administrative Tools, select Domain Security Policy.
2. In the Security Settings container, open the Account Policies container.
3. Select the Kerberos Policy folder.
4. In the display pane, select the appropriate policy to configure.

Figure 5.3 shows where you should be when configuring your domain security policy.

The policies you can configure are as follows:

- Enforce User Logon Restrictions
- Maximum Lifetime for Service Ticket
- Maximum Lifetime for User Ticket
- Maximum Lifetime for User Ticket Renewal
- Maximum Tolerance for Computer Clock Synchronization

**FIGURE 5.3**

*Configuring your domain security policy.*

### NOTE

Kerberos Authentication Policy is implemented at the domain level and overrides local security policy.

## Windows 2000 Kerberos Interoperability

In addition to the preauthentication issue mentioned earlier, another interoperability concern is the `authorizationdata` field in Windows 2000 Kerberos, which contains security information for Windows only. Microsoft's use of this field is not in strict compliance with the Kerberos standard, and is somewhat controversial.

Microsoft's view of Windows 2000 Kerberos interoperability is that

- A Windows 2000 client can authenticate to a non-Windows KDC.
- A non-Windows Kerberos client can authenticate to a Windows KDC.
- A non-Windows KDC can allow access to resources in a Windows domain providing a two-way trust exists between the Windows domain and the KDC's Kerberos area.
- A Windows KDC can provide access to resources on a non-Windows Kerberos area through either a service account or a one-way trust.

## What Kerberos Doesn't Prevent

Although Kerberos can help with Active Directory security, Kerberos does not provide protection from some types of attacks:

- Denial of service attacks—Although not Kerberos-based, these attacks are designed to fail or prevent the authentication process.
- Password guessing—An person breaching security could intercept a KDE request and crack the long-term key offline.
- Key discovery—A person breaching security who gains access to a key can impersonate the key's owner or impersonate a server that the key's owner is attempting to contact.

## Public Key Infrastructure

Another component of the authentication process in Active Directory is a public key infrastructure (PKI). PKI is a set of services and components including digital certificates and certification authorities (CAs), which performs verification and authentication used in electronic transactions. These processes use public key cryptography, which is critical to the distributed security necessary for e-commerce and Internet/intranet applications.

The Windows 2000 public key infrastructure is based on the open standards that are recommended by the Public Key Infrastructure for x.509 Certificates (PKIX) working group of the Internet Engineering Task Force (IETF).

### NOTE

Plan your public key infrastructure (PKI) before deploying certification authorities (CAs). For more information on configuring PKI, see the Windows 2000 Help.

Many Windows 2000 distributed security systems use public key technology. You can deploy a wide variety of security solutions that take advantage of the benefits of this technology.

## Major Public Key Infrastructure Components

PKI is a security system that uses certificates as its basis. A certificate is a digital voucher containing the various names of the account (such as the Active Directory ID, the UPN, email account, and so on) and a public key. A certification authority (CA) signs a digital certificate to prove that the account's private key is truly possessed by the account and is associated with the public key.

The major components of the Windows 2000 public key infrastructure include the following:

- Windows 2000 Certificate Services
- Microsoft CryptoAPI and cryptographic service providers (CSPs)
- Certificate stores
- Certificates console
- Certification authority trust model
- Certificate enrollment and renewal methods
- Public key Group Policy
- Certificate revocation lists
- Preinstalled trusted root certificates
- SmartCard support
- Windows 2000 Certificate Services

You can deploy Windows 2000 Server and Certificate Services to issue and manage certificates for your organization. You can also obtain Certificate Services from third-party vendors, such as VeriSign.

### **Windows 2000 Certificate Services CA Support**

Windows 2000 Certificate Services support two types of certification authorities: enterprise and standalone CAs.

Enterprise CAs are integrated with Active Directory and use certificate templates to specify certificates types. Standalone CAs do not require Active Directory and do not use certificate templates.

The certificate issued by the CA includes the public and private encryption keys. Public keys are available to the public to use when encrypting messages to be sent, and private keys are used to decrypt the messages.

### **Public Key Security Benefits**

The Windows 2000 public key infrastructure enables you to deploy security solutions using digital certificates and public key technology. Security solutions can include the following:

- Secure mail and Web communications (including Web sites)
- Software code signing
- Smart Card logon process
- Internet Protocol security (IPSec) client authentication
- Encrypting File System (EFS)

## SmartCards

A SmartCard resembles a plastic credit card and contains the private key for an account. It may also contain personal information related to your corporate information. You must have a PKI implemented prior to implementing SmartCards. If you implement SmartCards, you must have a Windows 2000 enterprise CA installed. You will also need a SmartCard reader and at least one enrollment station. Note that Windows 2000 supports only PC/SC-compliant plug-and-play SmartCard readers.

## IPSec

IPSec (Internet Protocol security) offers secure TCP/IP networking by securing network packets among compliant hosts. IPSec provides another layer of security against network and Internet attacks by working below the socket layer, transparent to applications.

IPSec protects IP packets and defends network attacks using cryptography-based technology. IPSec can be implemented on wide area networks (WANs), local area networks (LANs), and remote access clients.

## IPSec Benefits and Disadvantages

Some of IPSec's benefits, according to Microsoft, are that it is based on an open industry TCP/IP standard and provides transparent authentication. IPSec also preserves confidentiality and data integrity.

Unfortunately, a disadvantage is that IPSec may impact network performance. For this reason, you may consider implementing IPSec inside your firewalls, even though IPSec can be implemented either internally or externally.

## IPSec Functionality

According to Microsoft, Windows 2000 IP security ensures the following:

- Anti-replay (cannot be reused)
- Authentication
- Confidentiality
- Integrity
- Non-repudiation (sender cannot deny)

Upon sending a packet, IPSec verifies whether any of the IP security policies apply to the packet. If so, the participating computers negotiate through the Internet Key Exchange (IKE) protocol, which results in a security agreement (SA) between the hosts.

IP security policies can be added to the default domain policy, to local computer policies, or to other group policy objects.



## Configuring IP Security Policy

Windows 2000 includes an IP Security Policy Management MMC snap-in. The snap-in wizard asks if you want to edit IP security for the local computer, for the current domain, or for another domain. You can also edit IP policy directly from the Domain Security Policy utility or from Group Policy Editor.

To implement IP security, you define security policies that specify authentication, IP filtering, and other settings. You can configure IP security using the IP Security policy settings.

IP security policies determine when and how network hosts will implement security policy through IP security.

An IP security policy is a collection of one or more IP security rules. An IP security rule consists of the following elements (which are guided by the wizard):

- Tunnel endpoint—Option to input IP endpoint.
- Network type—Scope of rule setting. Select all connections, LAN, or remote access.
- Authentication method—The trusted connection between the computers. The default is Kerberos. Options include certification authority (CA) or preshared key.
- IP filter lists—Select all ICMP traffic, IP traffic, or specific IP filters.
- Filter action—The resulting action if a packet satisfies the filter list. Options include Permit, Request Security, and Require Security.

IP security is based on an if/then statement. If the IP address satisfies the filter, then an action must be applied. The filter action defines the occurrence. The filter identifies a packet by protocol and by source address or destination address using any of the following identification criteria:

- My IP address
- Any IP address
- Specific DNS name
- Specific IP address
- Specific subnet

One or more filters can be combined into a filter list, and one or more filter lists can be associated with an IP security policy. Filter lists and actions can be applied to a policy or multiple policies.

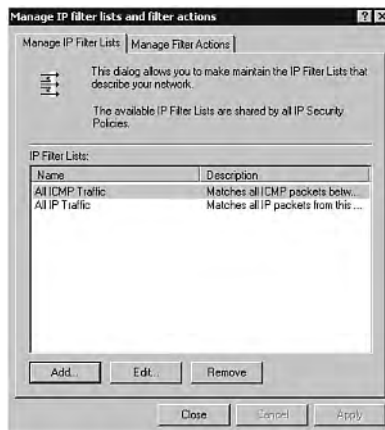
## Creating and Editing IP Filter Lists

An IP security policy must include at least one filter list. A filter list is a collection of source or destination DNS names, IP addresses, or IP subnets with optional information on the protocols to which the filter applies. Two filter lists are installed by default. They are

- All ICMP Traffic—Identifies all Internet Control Message Protocol (ICMP) traffic, regardless of source or destination address.
- All IP Traffic—Identifies all IP traffic, regardless of source or destination address.

To create an IP filter list for an Active Directory domain, perform the following steps:

1. In Administrative Tools, select Domain Security Policy and expand the Security Settings container if necessary.
2. Right-click the IP Security on Active Directory icon and select Manage IP Filter Lists and Filter Actions.
3. In the Manage IP Filter Lists tab, click the Add button (see Figure 5.4).
4. Enter a name and description for the new filter. Descriptions will be beneficial as your filter list grows. Any previously created filters will appear in the Filters box.
5. Click Add. If you uncheck the Use Add Wizard check box, the options are similar to those in steps 6 through 11.



**FIGURE 5.4**

*Creating a new IP filter list in IPsec.*

6. Click Next in the IP Filter Wizard.
7. Select the source address from the drop-down list to which the filter applies. It can be the computer's own address, any IP address, a specific DNS name, a specific IP address, or a specific IP subnet. For a specific DNS name, specific IP address, or specific IP subnet, you will be required to input the necessary addressing information. Then click Next.

8. Select the destination address to which the filter applies. This process is the same as step 6. Input the necessary information, and click Next.
9. Select the TCP/IP protocol to which the filter applies. Selecting Any filters all TCP/IP protocols. Or you can select Other, and input a port address. Click Next.
10. The next window informs you that you have finished creating the filter. You can edit the filter properties on completion of the wizard by checking the Edit Properties box if you like. Click Next.
11. When you complete the Add Filter Wizard, you return to the IP Filter List dialog box (step 4). To add another filter, click the Add button and repeat steps 5 through 9. To edit the properties of a filter, select the filter and click the Edit button. To finish creating the filter list, click Close.

Clicking the Edit button in the IP Filter List dialog box brings up the Filter Properties dialog box, which offers choices similar to creating a filter.

To edit a filter list, follow these steps:

1. In Administrative Tools, select Domain Security Policy. Expand the Security Settings container if necessary.
2. Right-click the IP Securities on Active Directory icon and select Manage IP Filter Lists and Filter Actions.
3. When the Manage IP Filter Lists and Filter Actions dialog box appears, select a filter list and choose Edit.
4. In the IP Filter List dialog box, edit the filter list name or description. Select the appropriate filter and choose Edit to open the Filter Properties dialog box. The tabs of the Filter Properties dialog box are as follows:
  - Addressing—Specifies source and destination address information, as well as mirroring information. *Mirroring* means that the filter also applies to packets with opposite source and destination addresses from the entries you specify. Filters are mirrored by default and can be disabled in this tab.
  - Protocol—Specifies the protocols and port addresses that will be filtered.
  - Description—Specifies a description or explanation for the filter.
5. Click Apply, and then OK.
6. In the IP Filter List dialog box, choose a new filter to edit and click the Edit button. Or click OK to close the IP Filter List dialog box.

## Creating and Editing Filter Actions

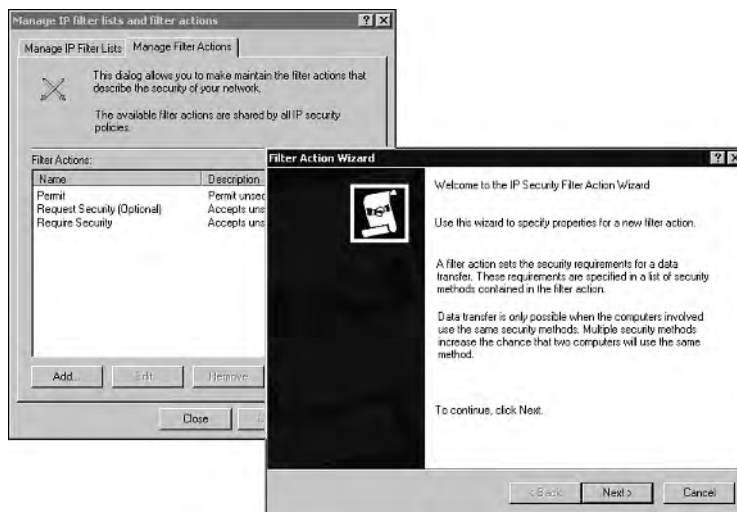
As mentioned earlier, the *filter action* is the resulting action that occurs when a packet meets the filter criteria. The following filter actions are installed by default:

- **Permit**—Permits a packet to pass even if it isn't secured.
- **Request Security(Optional)**—Attempts to establish a secure session but allows unsecured traffic to pass if a secure session can't be established.
- **Require Security**—Accepts unsecured requests initially but then requires the client to negotiate a secure session. This filter action refuses to communicate with unsecured clients.

You can create a rule that applies any of these filter actions to a default filter list or to a filter list that you create. You can also create your own filter actions.

To create a filter action, follow these steps:

1. In Administrative Tools, select Domain Security Policy. Expand the Security Settings container if necessary.
2. Right-click the IP Securities on Active Directory icon and select Manage IP Filter Lists and Filter Actions.
3. Select the Manage Filter Actions tab (see Figure 5.5).
4. Click Add. You can uncheck the Use Add Wizard check box to add a filter action without the Filter Action Wizard.



**FIGURE 5.5**

*Creating a new IP filter action in IPsec.*

5. Click Next in the Filter Action Wizard window.
6. Enter a name and description for the filter action, and click Next.
7. In the next window, select whether this filter action will permit, block, or negotiate security when filter criteria is met. If you select Permit or Block, you have already specified an action and there isn't much more to decide. Click Next and go to step 10. If you choose to negotiate security, you need to configure the security negotiation parameters. Click Next and go to step 8.
8. If you chose to negotiate security in step 7, the next window asks you whether this filter action will include communication with computers that do not support IPSec. The default is to not communicate with these. If you select Fall Back to Unsecured Communication, the computer accepts a lesser level of IP security (or no IP security) if the other host does not support IPSec. Click Next.
9. In the next window, choose one of the following security method options:
  - High (Encapsulated Secure Payload)—Provides encryption, authentication, and protection from modification
  - Medium (Authenticated Header)—Provides authentication and protection from modification
  - Custom—Allows you to set a custom integrity algorithm, encryption algorithm, and session key renewal settings
10. The next screen announces that you have finished creating a filter action. Check the Edit Properties box if you want to go immediately to the filter action's Properties dialog box. Then click Finish.

You can edit a filter action for a domain security policy by selecting Edit instead of Add in step 4.

## Creating an IP Security Policy

A security policy is a collection of rules. Each rule is a specific association of a filter list with a filter action, with accompanying settings for authentication, tunneling, and connection type. If you already created filter lists and filter actions, configuring security policy is relatively simple. If you haven't defined filter lists and filter actions, you can create them while you create the security policy, but wizards will overwhelm you in the process.

To create an IP security policy for a domain, perform the following steps:

1. In Administrative Tools, select Domain Security Policy. Expand the Security Settings container if necessary.
2. Right-click the IP Securities on Active Directory icon and select Create IP Security Policy.

3. Click Next.
4. Enter a name and description for the IP security policy, and then click Next.
5. The next window asks whether you want to include the default response rule in the security policy. Leave the box checked to include the default response rule. Or uncheck the box to leave the default response rule out of the policy. Then click Next.
6. If you elected to include the default response rule in step 5, the next window asks you to choose an authentication method to use with the default response rule. The default is Kerberos, but you can alternatively select a certification authority or enter a preshared key. Click Next after you make your choice.
7. The next window says you have completed the IP Security Wizard. Note that, so far, all you've done is created a policy and (optionally) placed the default response rule in that policy. To add other rules to the policy and configure the policy setting, you need to edit the policy. The last window asks whether you want to edit the policy immediately after completing the wizard. (The default is Yes.) If you leave this box checked, proceed to step 3 of the next procedure after this step. Otherwise, click Finish.

To add a rule to an IP security policy, follow these steps:

1. In Administrative Tools, select Domain Security Policy. Expand the Security Settings container if necessary.
2. Select IP Securities on Active Directory. The display pane lists the existing policies. Right-click one of the security policies and select Properties.
3. The policy's Properties dialog box appears. The Rules tab contains a list of the IP Security rules that are currently installed with the policy. The rule list might be empty, or it might contain the default response rule. To add additional rules, click the Add button.
4. When the Security Rule Wizard starts, click Next.
5. Select if this rule will be used in IP tunneling and, if so, select a tunnel endpoint IP address. Click Next.
6. Select the network type for this rule (all network connections, local area network, or remote access). Click Next.
7. Select the authentication method (Kerberos is the default), or select a CA or preshared key. Click Next.
8. Select an IP filter list for the rule. You can select one of the current filter lists or click Add to install a new filter list. Click Next.
9. Select a filter action for the rule. Select one of the current filter actions or click Add to install a new filter action. Click Next.
10. To edit rule properties before returning to the policy's Properties dialog box, leave the check box checked. Click Finish.

11. If you elected to edit the rule properties in step 10, the new Rule Properties dialog box appears. The tabs of the Rule Properties dialog box offer options similar to the Security Rule Wizard options described in steps 4 through 10.
12. After you add the new rule, you return to the policy's Properties dialog box. To add another rule, click the Add button and repeat steps 4 through 11. To edit the properties of a rule, select the rule and click the Edit button.
13. Select the General tab in the policy's Properties dialog box to edit the name and description of the policy. On the General tab, you also can configure policy update settings and key exchange settings.
14. Click Close to close the policy's Properties dialog box.

## Object-Oriented Security

Windows NT has always incorporated an object-oriented security approach in the form of NT file system (NTFS). Let's briefly review the fundamentals of object-oriented security.

When a user logs on, an access token is created. The access token contains security information for the user's account.

### NOTE

A security principal (or account) can be a user, group, computer or service. Security principals have accounts that can be assigned rights or privileges. Local accounts are managed on the local computers Security Accounts Manager (SAM). Active Directory manages domain accounts.

Accounts, or processes, use or manipulate resources known as objects, which can include files, shares, printers, computers, and so on. Security for objects includes access lists to determine what account or process can gain access. More on access control lists, and other object oriented security features is seen later in the chapter.

## Access Control Lists (ACLs)

As mentioned in Chapter 1, "Understanding Active Directory," Access Control Lists determine whether or not an account can access an object. An *ACL* is a list of Access Control Entries (ACEs) stored with the object. The Access Control List for an object is generally located in the Security tab of the property sheet. This tab shows the list of accounts that have access to this object, as well as the associated permissions. The Advanced button displays the account permissions in detail.

Windows 2000 stores the ACL as a binary value called a *security descriptor*. Each ACE contains a security identifier (SID), which identifies the security principal (user, for example) to which the ACE applies and the set of access rights that are allowed, denied, or audited for that security principal (user).

## Access Control Entries (ACEs)

ACLs on directory objects contain ACEs that apply to the object as a whole, and ACEs that apply to the individual attributes of the object. This allows an administrator to control not just which accounts can access the object, but what properties can be viewed by that account.

For example, all users might be granted read access to the email and telephone number attributes for all other users in the directory, but security properties of users might be denied to all but members of a special security administrators group. Individual users might be granted write access to personal attributes such as the telephone and mailing addresses on their own user objects.

To view the ACEs for a particular ACL, click Advanced at the bottom of the Security tab of the Properties dialog box. The Access Control Settings Editor then appears, as shown in Figure 5.6. When you're adding an ACE, rights can be granted and denied. As usual, denying a permission takes precedence over granting one, so if an account is granted write access to an object via one group and denied access through another group, the account has no access to the object.

### NOTE

Because there is a good chance that groups will require similar access to objects, applying ACEs to groups can reduce administrative effort in the future.

## Security Identifier (SID)

A security identifier (SID) is a unique value of variable length used to identify a security principal or security group. The SID identifies a user, group, service, or computer account within an enterprise. Every account is issued a SID when it is created. Access control mechanisms in Windows 2000 identify security principals by SID rather than by name. Windows 2000 uses SIDs in the following access control components:

- Access tokens—One SID in an access token identifies the account represented by the token. Additional SIDs identify the security groups to which the account belongs.
- Security descriptors—One SID in an object's security descriptor identifies the object's owner. Another SID identifies the owner's primary group.
- Access Control Entries (ACEs)—Each ACE contains a SID that identifies the account for which access is allowed, denied, or audited.



The Local Security Authority (LSA) creates the SID when the local account is created. The domain security authority (DSA) generates the SID when a domain account is created, and the SID is then stored as an attribute of the object in Active Directory.

## Rights and Permissions

A *right* is authorization to perform an operation. In Windows 2000, only the right to allow or deny access to resources that you own is inherent. All other rights must be granted. From an administrator's perspective, there are two types of rights: permissions and user rights.

A *permission* is authorization to perform an operation on a specific object, such as a file. Owners grant permissions. If you own an object, you can grant any user or security group permission to do whatever you are authorized to do with it, including granting permission to take ownership.

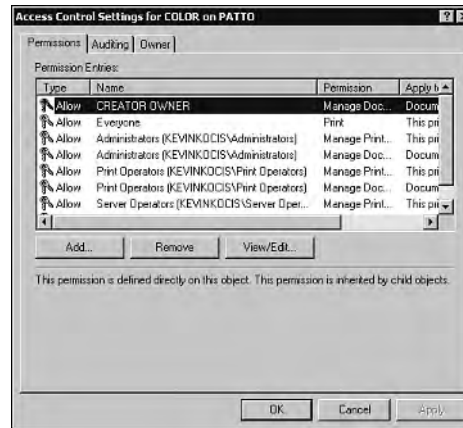
### NOTE

Although you can give permissions to individual users, giving them to a security group is more efficient. That way, you can grant permission once to the group rather than several times to each individual. Every user added to a security group receives the permissions defined for that group.

When permission to perform an operation is not explicitly granted, it is implicitly denied. For example, if Jennifer allows the Sales group, and only the Sales group, permission to read her file, users who are not members of the Sales group are implicitly denied access. The operating system does not allow users who are not members of the Sales group to read the file.

Permissions can also be explicitly denied. For example, Jennifer might not want Chris to be able to read her file, even though he is a member of the Sales group. She can exclude Chris by explicitly denying him permission to read the file. Explicit denials are best used when excluding a subset (such as Chris) from a larger group (such as Sales) that has been given permission to do something.

Each permission that an object's owner grants to a particular user or group is stored as an ACE in a DACL that is part of the object's security descriptor. In the user interface, ACEs appear as Permission Entries in the Access Control Settings dialog box (see Figure 5.6).



**FIGURE 5.6**

*Access Control Entries for a local shared printer ACL.*

## Security Descriptor

An object's security descriptor contains access control information, and identifies the object's owner by SID. If permissions are configured for an object, its security descriptor contains a Discretionary Access Control List (DACL) with SIDs for the account allowed or denied access. If auditing is configured for the object, its security descriptor also contains a System Access Control List (SACL) that controls how the security subsystem audits attempts to access the object.

For example, if a user attempts to modify the driver on a network printer, the operating system examines the user's security descriptor to determine whether he is allowed to perform the modification.

Generally, security descriptors can include information about the following:

- **Owner**—The only security principal with an inherent right to allow or deny permission to access an object. Ownership can be transferred. By default, the built-in Administrators group on a computer is assigned a user right that allows this group to take ownership.
- **Permission**—Authority to perform an operation or a set of operations on an object, which is granted or denied by an object's owner. Because access to an object is given at the owner's discretion, the type of access control used in Windows 2000 is called *discretionary access control*.

- **User right**—Authority to perform an operation that affects an entire computer rather than a particular object. User rights (also known as *privileges*) are assigned by administrators to individual users or groups as part of the security settings for the computer. Although user rights can be managed centrally through Group Policy, they are applied locally.
- **Access right**—A permission from a subject's point of view. When a user allows or denies permission through the Access Control Settings dialog box, the result is recorded as an ACE in the object's DACL. Although a permission is represented by a word or phrase in the user interface, in an ACE, it is represented by a set of bit flags in an access mask. Each bit flag corresponds to an access right.

## Security Descriptor Components

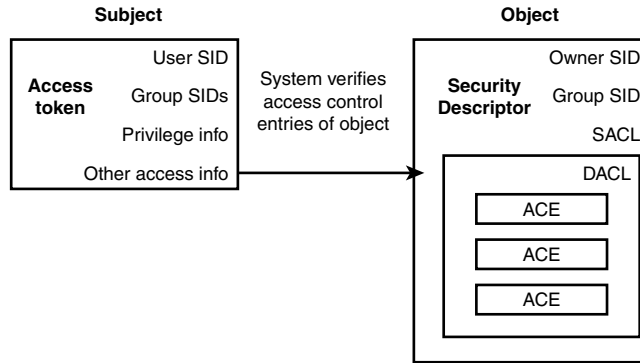
A *security descriptor* is a binary data structure of variable length, which contains the following components:

- **Header**—The header contains a revision number and a set of control flags that describe characteristics of the security descriptor.
- **Owner**—The Owner field contains the SID for the object's owner.
- **Primary Group**—The Primary Group field contains the SID for the owner's primary group, which is only used for POSIX compliance.
- **Discretionary Access Control List (DACL)**—The DACL is a list of ACEs, which is controlled by the object's owner.
- **System Access Control List (SACL)**—The SACL resembles the DACL but is used to control object auditing. When an audited event occurs, it is recorded in the security log. Each ACE in an SACL has a header that indicates whether auditing is triggered by success or failure or both, a SID that specifies an account to monitor, and an access mask that lists the operations to audit.

An access check compares information in the thread's access token with information in the object's security descriptor:

- The access token contains a SID that identifies the user associated with the thread and SIDs that identify the groups whose members include the user.
- The security descriptor contains a DACL with ACEs that specify the access rights allowed or denied to specific accounts.

The security subsystem checks the object's DACL, looking for ACEs that apply to the user and group SIDs from the thread's access token. It steps through each ACE until it finds one that either allows or denies access to the user or one of the user's groups, or until there are no more ACEs to check. If it comes to the end of the DACL and the thread's desired access is still not explicitly allowed or denied, the security subsystem denies access to the object. Figure 5.7 illustrates this process.

**FIGURE 5.7**

*Validating a request for access.*

The order in which ACEs are listed in a DACL is important. For example, an object's DACL might contain one ACE that allows access to a group and another ACE that denies access to a user who is a member of the group. If the allowing ACE precedes the denying ACE, then the user is allowed to access the object, which is not a desirable situation.

## Active Directory Object Security

Active Directory stores objects and their respective information, and makes them available on the network. When Active Directory is queried, it provides information back to the respective user or program. Although the security systems in Windows 2000 resemble those in Windows NT. They include

- User-based authorization
- Discretionary access to securable objects
- Permission inheritance
- Administrative privileges
- Auditing of system events

## Active Directory Objects

The common object types in Active Directory are

- User
- Group
- Computer
- Contact

- Printer
- Shared folder

The objects in the Active Directory are protected by ACLs. If a user is not permitted to view an object, he will not be able to see that object in Active Directory.

In Active Directory, an object can have standard permissions or special permissions assigned to it. While most objects will have both, standard permissions are more common and will suit most access requirements.

The standard object permissions are

- Full Control, which allows change permission and the ability to take ownership.
- Read, which allows the ability to view objects and their attributes, the owner, and Active Directory permissions.
- Write, which allows the ability to modify the object attributes.
- Create All Child Objects, which allows the ability to create any type of child object in the OU.
- Delete All Child Objects, which allows the ability to delete any type of object in the OU.

## Assigning Active Directory Permissions

You can set object permissions for an Active Directory object just as you would for an object on an NT File System (NTFS). Before you can administer object security, you must make sure the Advanced Features option is selected. To do this, open Active Directory Users and Computers, and under the View menu, select Advanced Features. This makes the security options available to administer.

To set object permissions, perform the following steps:

1. Right-click the desired object and click Properties.
2. In the Properties dialog box, click the Security tab.
3. Select the account you wish to modify and check the appropriate permission (Allow or Deny).

### NOTE

You can also add accounts using the Add button, and then assign the appropriate permissions to the object.

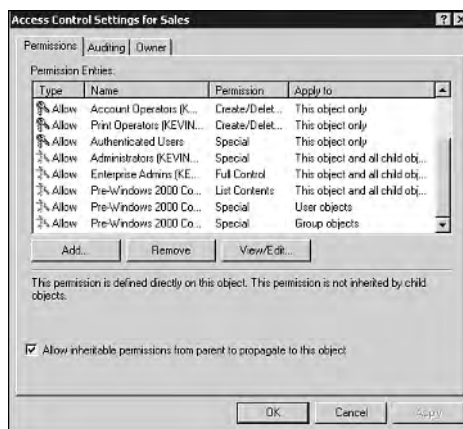
While standard permissions will suffice for most administrative needs, special permissions can be accessed by clicking the Advanced button in the Security tab.

**CAUTION**

Avoid assigning permissions for object properties, as this can significantly impact functionality. By assigning a wrong or misinterpreted permission, you can hide the object or make it inaccessible in Active Directory.

## Object Permission Propagation

Permission propagation works differently in Active Directory than it does on NTFS volumes. With Active Directory permissions, you can assign permissions to the object, as well as child objects and which type of child objects. Let's take the example of the Sales container. In Figure 5.8, the access control settings window shows the varying permissions for the Sales container. Some permissions apply to the container (this object only), some apply to the container and all child objects, and others apply only to user and group objects in the container.



**FIGURE 5.8**

*The Access Control Settings for the Sales Container.*

If you own the Sales container object, you can allow access to certain types of objects without allowing access to other types of child objects. For example, you can add a permission that allows the Sales group write access, and then apply the permission to a particular type of child object contained by the OU. This demonstrates how permissions propagated from container objects in Active Directory can be object-specific in direct contrast to NTFS volumes, where this form of permission propagation cannot occur.

Active Directory permissions for objects can be managed at two levels: the object level and the property level. Permissions allowed or denied at the object level apply to the entire object. For example, you assign an object-level permission on the Sales container that allows the group Sales Managers to create child objects in the Sales container.

Permissions allowed or denied at the property level apply only to specific properties. For example, you can set a property-level permission on the Domain Users object that allows a human resources representative to change various properties such the address or work number of an employee.

To assign per-property permissions, perform the following steps:

1. Right-click the desired object and click Properties.
2. In the Properties dialog box, click the Security tab.
3. Click the Advanced button.
4. Click View/Edit to modify an existing account permission, or click Add to create new one. (If you click Add, you must select the account, such as the Human Resources group, that will be assigned the permission.)
5. In the Permission Entry window, click the Properties tab to view the per-property permissions list.

Not all object properties are listed in the Properties tab. Only properties commonly used for controlling access are listed. The schema contains and defines all the object types and properties (see Chapter 7, “Managing and Modifying Active Directory Schema,” for more information). The user interface for access control filters out object types and properties to make the list easier to manage.

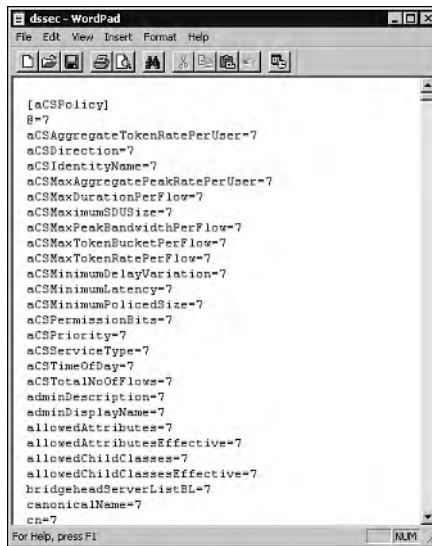
The list of filtered object types and properties is kept in the file `dssec.dat` located in the `%systemroot%\System32` folder on every domain controller. You can modify the behavior of the filter by adding or removing items from the list. `dssec.dat` is a text file in the following format:

```
[objectType]
@ = 7
attributeName = 7
```

Figure 5.9 shows this file in Notepad.

Object types are in brackets. If an @ sign is on the next line below the bracketed object type and is set to 7, the object type is filtered. To stop filtering for that object type, change the setting to 0. Filtered attribute names are listed below the object type.

When you change `dssec.dat`, your changes are not reflected on the Properties tab until you close the current tool or snap-in and restart it. Filter data is read when the tool is initialized.

**FIGURE 5.9**

*The dssec.dat file viewed in Notepad.*

## Publishing Active Directory Resources

In Active Directory, various resources are made available to users through publication. These resources include user and group information, computer, shared folders, printers, and other services.

Published resources are stored in the directory for quicker and more convenient access, and are integrated with Windows 2000 security to control access.

Users and computers are published in the directory through the Active Directory Users and Computers console. Non-security information regarding these accounts is then made available to the enterprise. Security information such as group membership and personnel information is restricted to approved groups and administrators.

## Publishing Shared Folders

Publishing shared folders in Active Directory parallels the sharing process in NTFS. The difference is that folders shared in Active Directory are available enterprise-wide, based on permissions.



To publish a shared folder, perform the following steps:

1. Open Active Directory Users and Computers.
2. In the console tree, expand the domain node if necessary.
3. Right-click the container where you want to add the shared folder, select New, and click Shared Folder.
4. In the New Object-Shared Folder dialog box, enter the folder name in the Name field.
5. In the Network Path box, type the UNC name (`\\server\share`) that you want to publish in the directory, and then click OK.

As mentioned earlier, shared resources integrate NTFS permissions. You can modify folder permissions by performing the following steps:

1. Right-click on the desired shared folder, click Properties, and then click the Security tab.
2. To create permissions for a new group or user, click Add. Type the name of the group or user you want to set permissions for using the *domainname\name* format and then click OK.
3. To change or remove permissions from an existing group or user, click the name of the group or user.
4. In the Permission list, click Allow or Deny for each permission, if necessary. Select Remove if you wish to remove the group or user from the Permission list, then click OK.

#### NOTE

To change permissions, you must be the owner or have been granted permission to do so by the owner.

Groups or users granted Full Control for a folder can delete files and subfolders within that folder, regardless of the permissions protecting the files and subfolders. If the check boxes under Permissions are shaded or if the Remove button is unavailable, the file or folder has inherited permissions from the parent folder.

To set, view, or remove permissions for a shared folder or drive, perform the following steps:

1. Right-click the shared folder or drive on which you want to modify permissions, and click Sharing.
2. On the Sharing tab, click Permissions.

3. To set shared folder permissions, click Add. Type the name of the group or user you want to set permissions for and then click OK to close the dialog box.

To remove permissions, select the group or user in the Name list and then click Remove.

4. In the Permission list, click Allow or Deny for each permission, if necessary.

To share folders and drives, you must be logged on as a member of the Administrators, Server Operators, or Power Users.

Shared folder permissions apply to all files and subfolders in the shared folder and are effective only when the folders or files are accessed over a network. Shared folder permissions do not protect folders or files when opened locally. NTFS permissions are required to protect files and folders on your local computer. NTFS permissions are applied in addition to shared folder permissions.

You can use the Shared Folders snap-in to create and manage shared folders, and to view connected users to shared folders and files. You can also change permissions for shared folders on remote computers.

#### NOTE

For administrators interested in managing shares with NT 4.0 tools, you can still utilize the Server Manager tool by entering **usrmgr.exe** in the Start, Run window or at a command prompt on your domain controller.

## Publishing Printers

Printers are another resource that can be published in Active Directory. To publish a Windows NT printer, perform the following steps:

1. Open Active Directory Users and Computers.
2. In the console tree, expand the domain node if necessary.
3. Right-click the container where you want to publish the printer, select New, and then click Printer.
4. In the New Object-Printer box, enter the UNC name that you want to publish in the directory, and then click OK.

**NOTE**

All Windows 2000 printers are published in Active Directory by default. You can disable this process by selecting **Do not share this printer** in the wizard's **Printer Sharing** screen. Windows NT printers require installation before being published in Active Directory.

## Publishing Guidelines

When considering publishing resources in Active Directory, consider whether or not the resource is useful to a wide audience of users. By publishing an uncommon resource, such as an engineering plotter, you may be wasting network resources since the plotter may not be useful to anyone but the local engineering group.

You should also ensure that the object would remain relatively static, and not change very often. Attribute changes are replicated throughout Active Directory, so the more frequent the modification, the more impact on Active Directory replication.

## Ownership and Delegation

The ownership permission parallels the Window NT 4.0 version. Administrators or delegated authorities can take ownership of a file or folder. Permission to take ownership can also be delegated. Delegating permissions to administrators serves no purpose; this permission must be delegated to approved users who are not members of the Administrators group.

To take ownership of a file or folder, perform the following steps:

1. Right-click the file or folder on which you want to take ownership, click **Properties**, and then click the **Security** tab.
2. Click **Advanced** and then click the **Owner** tab.
3. Click the new owner and then click **OK**.

**NOTE**

To change the owner of all subcontainers and objects within a tree, select the **Replace Owner on Subcontainers and Objects** check box.

When an administrator takes ownership of a file or folder, the Administrators group then owns the file. Any member of this group can grant access to the file or folder on the computer. The administrator can then grant access to herself or to another user. The owner or administrator

cannot transfer ownership to others. She can only grant the Take Ownership attribute. This restriction maintains accountability on behalf of the administrator.

As I mentioned earlier, delegation applies to approved users or groups that do not have administrative privileges. To delegate control to a child domain or organizational unit, perform the following steps:

1. Open Active Directory Users and Computers.
2. In the console tree, expand the domain object to display the child domains or organizational units.
3. Right-click the child domain or organizational unit for delegated administration, and click Delegate Control.
4. Follow the instructions in the Delegation of Control Wizard.

## Permission Inheritance

Permission inheritance allows the administrator or owner of an Active Directory object to minimize the granular assignment of permissions for that object. With permission inheritance, you can allow a permission to propagate to child objects that are created under the parent object.

For example, you can assign Full Control permission to the Sales Managers group for the Sales Reports folders, and assign permission inheritance to the folder. Any subfolders and files will inherit the permissions, and any member of the Sales Managers group will have full control.

To assign permission inheritance, in the Permissions window, under the Object tab, select the desired inheritance in the Apply onto pull-down menu.

To prevent permission inheritance on a child object, clear the Allow Inheritable Permissions From Parent to Propagate To This Object check box.

## Conflicts Between Privileges and Permissions

For the most part, conflicts between privileges and permissions occur only in situations in which the rights required to administer a system overlap the rights of resource ownership. When rights conflict, a privilege overrides a permission.

For example, if the Backup Operators group has the right to back up all files and folders, they can do so regardless of their permissions to those files and folders.

## Best Practices for Access Control

Distributed security greatly simplifies the work of assigning rights and permissions across a container tree—hierarchy of containers, groups, users, computers, and other resource objects. To take best advantage of this feature, apply the following general principles:

- Assign user rights on a group basis.
- Rely on inheritance from group assignments. Because maintaining user accounts directly is inefficient, assigning rights and permissions on a user basis should be the exception.
- Assign rights as high in the container tree as possible. By doing so, you gain the maximum effect with little effort. The rights and permissions you establish should be adequate for the majority of the security principals.
- Apply inheritance to propagate rights and permissions through the container. Just as applying access control from a higher level of the tree provides a wider scope, inheritance provides the depth. You can quickly and effectively apply access control settings to child objects.
- Delegate the administration of containers to the approved users who manage the resources where those containers reside. By delegating authority to administer the rights for a container, you can decentralize administrative operations and issues.

## Summary

This chapter addressed the new security features in Active Directory, including Kerberos authentication and IPSec network security. Although NTFS permissions still apply to Windows 2000 objects, you can delegate authority to approved users who are not members of an administrative group. This eases administrator responsibility and allows for local “control” of resources. Publishing resources is another security benefit of Active Directory, allowing users to more conveniently browse the OU, domain, or even entire directory for resources in the enterprise.



# Administering Group Policy

CHAPTER

# 6

## IN THIS CHAPTER

- **Group Policy Fundamentals 172**
- **Trust Relationships with Previous Versions of Windows 202**

The System Policy Editor that was introduced in Windows NT 4.0 evolved into Group Policy in Windows 2000. The term “Group Policy” is derived from grouping multiple policies together in one policy. Group Policy MMC snap-in is new to Active Directory. The snap-in is a flexible tool that extends the functionality of the System Policy Editor and allows you to configure multiple user and computer settings. Group Policy can be enforced or blocked at the site, domain, and organizational unit level, as you’ll learn later in this chapter.

The following sections provide a closer look at group policy.

## Group Policy Fundamentals

Group Policy is similar to NT 4.0’s System Policy Editor in that it manages computers and users in Active Directory, including software deployment and configuration management. Group Policy dictates configurations for users and computers in Active Directory.

Group Policy forces users to maintain customized and consistent environments as deemed by the group policy administrator. Group Policy, by default, affects all computers and users in a site, domain, or organizational unit (depending on where it is linked), but no other objects such as printers or shares.

### NOTE

In particular, Group Policy does not affect security groups. The location of a security group in Active Directory is irrelevant to Group Policy.

Security groups are used to filter Group Policy. Filtering is changing the scope. You do so by changing the Apply Group Policy and the Read permissions on the Group Policy object for the relevant security groups. You’ll learn about Group Policy objects in the Group Policy Objects section, and filtering in the Configuring Group Policy section.

## Windows NT 4.0 and Windows 2000 Policy Comparison

In Windows NT 4.0, you could use the System Policy Editor (Poedit.exe) to specify user and computer configurations stored in the Registry. With the System Policy Editor, you could control the user work environment and enforce system configuration settings for all domain computers running Windows NT 4.0.

In Windows 2000, you can use the Group Policy snap-in to manage desktop configurations for computers and users.

System Policy Editor could only apply to domains, whereas Group Policy can also apply to sites and OUs. NT 4.0 policies were also not secure.



**NOTE**

Default policy settings in Group Policy do not remain in the registry permanently as they did in NT 4.0. Microsoft referred to this as registry “tattooing.” Windows 2000 Group Policy settings are removed when they no longer apply.

Group Policy affects all Windows 2000 computers and users located in the site, domain, or OU that are linked to the Group Policy objects. Group Policy can be filtered, based on an accounts security group membership. Filtering is covered in the “Filtering and Delegating Group Policy with Security Groups” section later in the chapter.

Windows 9x and NT machines cannot interpret a Windows 2000 container. They can only recognize their existence in a domain.

**CAUTION**

Although Windows 9x and NT 4.0 clients and computers can be supported by using Windows NT 4.0 templates and the System Policy Editor, it is not recommended to support this in addition to Windows 2000 Group Policy. Also, due to possible registry tattooing mentioned earlier, problems may result when upgrading from earlier clients exposed to NT 4.0 policies.

## Group Policy Administrative Requirements

To set Group Policy for a selected Active Directory site, domain, or organizational unit, you must have access to a Windows 2000 domain controller for that Active Directory, and you must have Read/Write permissions to access the system volume of domain controllers (that is, the Sysvol folder). You also must have Modify Rights to the selected directory site, domain, or OU.

**NOTE**

Only Domain Administrators, Enterprise Administrators, Group Policy Creator Owners, and the operating system can create new Group Policy Objects by default.

A non-administrative user or group can be added to the Group Policy Creator Owners security group. When a member of that group creates a Group Policy Object (GPO), he or she becomes the Creator Owner of the GPO, and can edit the GPO.

As a member of the Group Policy Creator Owners group, you have full control of only GPOs that you created or that were delegated to you.

## Group Policy Objects (GPOs)

Group Policy settings are stored in Group Policy Objects (GPOs). GPOs can control settings for users and computers in sites, domains, and OUs, and are linked with Active Directory containers.

Group Policy settings are hierarchical, and can be applied at the following levels:

- **Site**—These GPOs apply to all domains and servers within a site.
- **Domain**—These GPOs are the second level where GPOs can be assigned, and apply to all the containers (OUs) within the domain. They are usually assigned by local domain administrators.
- **Organizational Units (OUs)**—This GPO setting is the most granular; here, policies can be set for business units inside the domain.

To understand the inheritance and granularity of GPOs, refer to the user John in Figure 6.1. John will receive the Group Policy settings from all the policies from his current OU up to the site group policy.

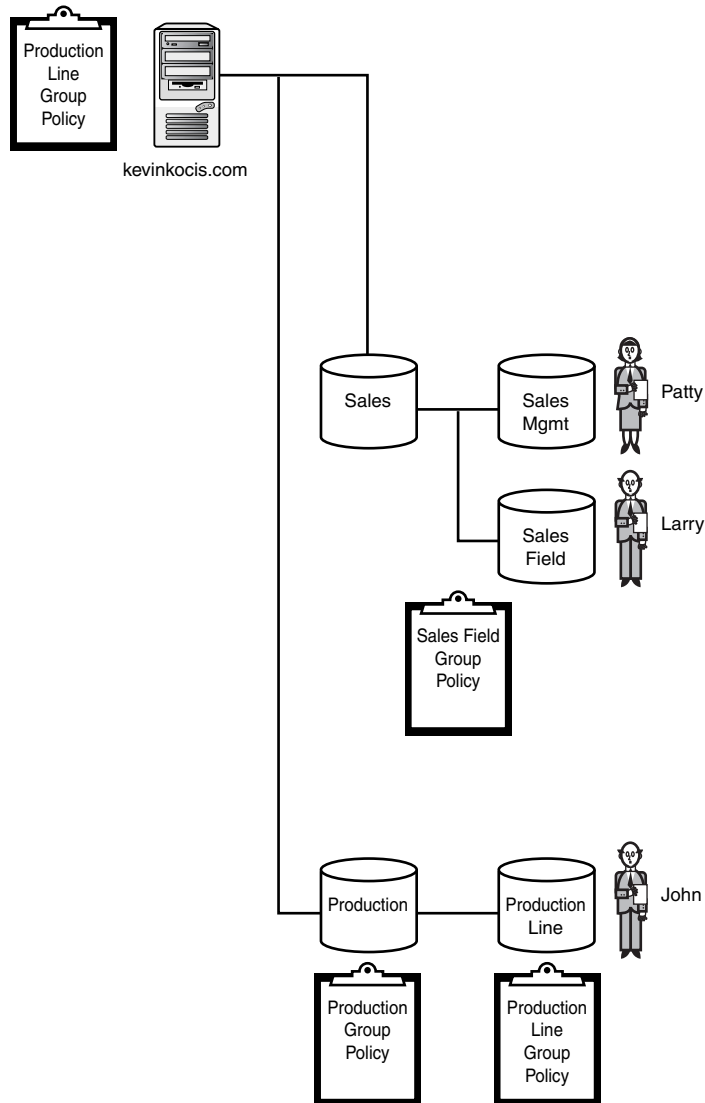
In this example, John would inherit the group policies for the domain, the production OU, and the production line OU. Also, because Patty is a member of the sales management OU, she is subject only to the domain group policy (because the sales OU does not have a group policy associated with it). Larry would inherit both the domain and sales field group policies.

In most cases, group policies are cumulative. You can also associate multiple GPOs with the same container. Although GPOs cannot be copied between containers, you can link a GPO to more than one container.

### NOTE

It is important to note that without GPOs, a user's location in the OU hierarchy does not impact logon times. The number of GPOs that must be read and applied to the user or computer creates logon delays.

In Figure 6.1, users logging in to the root domain (kevinkocis.com\users container) would have comparable logon times to those in the sales management OU. Users such as Larry in the production line OU would experience longer logon times because of the number of group policies (three) that must be applied. So, it is not the “depth” of an account, but the number of policies that are applied that affects logon speed.

**FIGURE 6.1**

*Inheritance of group policies.*

Group Policy settings from more than one GPO can be applied to a particular site, domain, or organizational unit. A separate policy for each domain could determine specific security settings for the domain's computers. The ability to automatically configure and secure computers throughout your organization by using selectively applied GPOs can be helpful as an administrative tool.

**NOTE**

You cannot open Group Policy Objects in read-only mode. Also, there is no exclusive save mode for changes to Group Policy. Saving occurs during the actual editing process.

Computers and users in Windows 2000 domains can belong to security groups, which can be used to filter how Group Policy settings are applied to collections of users and computers belonging to a particular site, domain, or organizational unit. Filtering and delegation of Group Policy are discussed near the end of the chapter in the “Filtering and Delegating Group Policy with Security Groups” section.

**NOTE**

Group Policy Objects fit into two categories: local and non-local. Only one Local Group Policy Object is stored on each Windows 2000-based computer. Local policy is least influential if the computer resides in an Active Directory environment. Non-Local Group Policy Objects are Active Directory-based.

## Creating Group Policy Objects (GPOs)

To create a group policy using the MMC snap-in, perform the following steps:

1. Click Start, Run. Then type **mmc** and press Enter.
2. On the Console menu, click Add/Remove Snap-in.
3. Click the Add button, select Group Policy, and then confirm by clicking Add again.
4. In the Group Policy Object window, click Browse to choose a policy other than the local computer policy. You should see the Default Domain Policy (as well as OUs, domains, and sites that may contain additional policies).
5. Right-click in the window and select New, or click the New Policy icon to the right of the drop-down list to create a new domain Group Policy Object (see Figure 6.2).
6. Name the new object and click OK to open it.
7. Check the Allow the Focus of the Group Policy Snap-In to Be Changed When Launching from the Command Line option. This way, you can change the context of the snap-in when you launch the MMC item.
8. Click Finish and then OK to add the new snap-in.

**FIGURE 6.2**

*Creating a new domain Group Policy Object.*

These new objects are linked to the domain or organizational unit by default. To create a Group Policy Object with this method, you must have permission to create the Group Policy Object, as well as permission to link it to the domain or organizational unit. Otherwise, the New button on the Properties page for the domain or organizational unit is grayed out.

## Configuring Group Policy

By default, Domain Administrators, Enterprise Administrators, the operating system, and the Group Policy Object Creator Owner have full control of Group Policy Objects without the Apply Group Policy attribute. GPO permissions can be delegated as well (see the section, “Delegation of GPO Administration,” for more information).

The group policy itself is contained within a Group Policy Object in Active Directory. The object is created in the Group Policy Editor, which can be launched in three different ways:

- From the Active Directory Users and Computers console, from the Group Policy tab on container objects
- From the Active Directory Sites and Services console, from the Group Policy tab on container objects
- As a separate management console using MMC and opening gpedit.msc

Group policies can be applied to four containers:

- **Local Group Policy**—The Local Group Policy Object exists on each Windows 2000 computer. It contains, by default, only policies regarding security. The policy is located in the %systemroot%\system32\GroupPolicy directory. The Local Group Policy is the only non-Active Directory policy, and is least influential in AD.

- **Site Group Policy**—Site Group Policy Objects are linked to site objects and can affect any object across the entire forest because sites can span domains. Although linked to a site, a Site Group Policy physically exists on a domain controller within a single domain.
- **Domain Group Policy**—Domain Group Policy Objects are linked to a single domain and affect all user and computer objects within the domain and its subcontainers.
- **Organizational Unit (OU) Group Policy**—OU Group Policy Objects are linked to a specific OU. The OU Group Policy affects all objects within the OU and within any OUs nested below it in the hierarchy.

## Order of Policy Implementation

When a computer boots into Active Directory, it inherits the settings in the Computer Configuration portion of its associated GPOs and applies them. When a user logs on, he or she inherits the settings in the User Configuration portion of the group policy, which are then applied to the user's account. Only users and computers receive group policies. Computer configuration and user configuration are discussed in further detail in the next section.

### NOTE

Group policies override any local user profile settings, so if a default screen saver is set in Computer Configuration, any users who customize their screen saver will inherit the default set in the GPO.

The order of policy application begins with legacy NT4 system policies, if they exist (and they are not recommended to coexist with Windows 2000 Group Policy!). Otherwise, the order is as follows:

- Local Group Policy Object
- Site Group Policy Object
- Domain Group Policy Objects
- OU Group Policy Objects from the parent OUs down to the user's or computer's OU location

The order of application detailed in the preceding list is significant to the architecture of Active Directory because, by default, a policy applied later overwrites a policy applied earlier for each setting where the later applied policy is either Enabled or Disabled.

You also can force or prevent GPOs from affecting groups of users or computers. The most powerful settings for avoiding the default behavior are the No Override and Block Policy Inheritance settings. It is best to minimize the use of these settings. These concepts will be addressed later in the chapter.

## Local GPOs

A Local Group Policy Object exists on every computer, and by default only nodes under Security Settings are configured. Settings in other parts of the Local Group Policy Object's namespace are not enabled or disabled. The Local Group Policy Object is stored in %systemroot%\System32\GroupPolicy, and it has the following permissions set through Discretionary Access Control Lists (DACLS):

- Administrators: full control
- Operating system: full control
- User: read

### NOTE

If Read permission is withdrawn from the Local Administrator group, Group Policy does not apply. Withdrawing this permission is a convenient way to exempt Local Administrators from a GPO even though they have the Apply Group Policy permissions set to Allow.

## Site GPOs

Site GPOs are processed after Local GPOs. Any GPOs that are linked to the site are processed synchronously, per the administrator's set order (in the case of multiple linked GPOs). The site GPOs apply to any user or computer in the site, regardless of the domain. Site GPOs override any Local GPO.

## Domain GPOs

Domain GPOs are processed after Site GPOs and override any Local or Site GPOs. Domain GPOs are also processed synchronously, per the administrator's prescribed order. These GPOs apply to the domain, including all the subcontainers.

## Organizational Unit GPOs

OU GPOs are processed after the previously mentioned GPOs (Local, Site, Domain), and have the greatest influence on group policy for the user or computer. GPOs linked to the highest OU in the Active Directory hierarchy are processed first, followed by the child OUs, and so on down the hierarchy. OU GPOs also follow the synchronous order as set by the administrator. The last OU GPO to be applied overrides any previous OU GPOs in the list.

## MMC Snap-in Extension Model

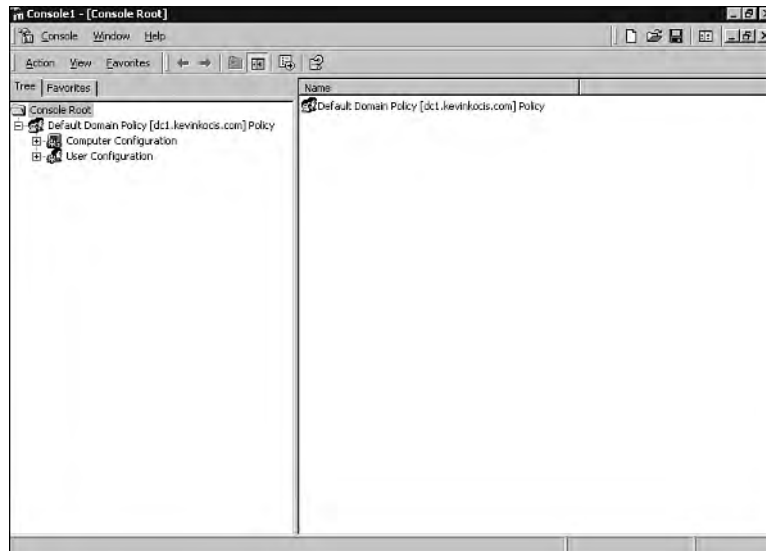
The main nodes of the Group Policy snap-in are MMC snap-in extensions that load when you start the Group Policy snap-in. These extensions include Administrative Templates, Scripts, Security Settings, Software Installation, Remote Installation Services, Internet Explorer Maintenance, and Folder Redirection.

The root node of the Group Policy snap-in appears as the name of the Group Policy Object and the domain in which it is stored, in the following format:

*<Group Policy Object name> [<server name>] Policy*

Here's an example (see Figure 6.3):

Default Domain Policy [dc1.kevinkocis.com] Policy



**FIGURE 6.3**

*Domain-level group policy.*

The next level of the namespace has two nodes: Computer Configuration and User Configuration. They are the parent folders used to configure and enforce Group Policy on computers and users.

## Computer Policy

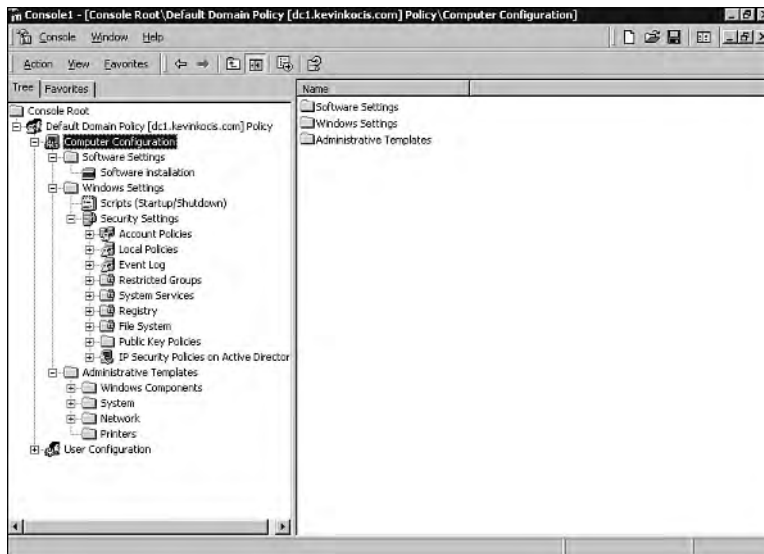
The Computer Configuration policies can change Registry settings within HKEY\_LOCAL\_MACHINE. The settings in the Computer Configuration policies are applied to a computer



regardless of the logged on user. Aside from assigning interface preferences, group policies can apply logon, logoff, startup, and shutdown scripts; distribute software; change security settings; and redirect system folder locations such as My Documents.

The Computer Configuration settings specify operating system behavior, which include desktop and security settings, as well as startup and shutdown scripts. Because the Computer Configuration settings are applied to a computer, this policy is best applied to computers that require being locked down to protect local data or prevent misuse of applications.

The Computer Configuration portion of group policies includes a abundance of security settings, as shown in Figure 6.4, that are applied to individual computers.



**FIGURE 6.4**

*Computer Configuration group policy in the default domain policy.*

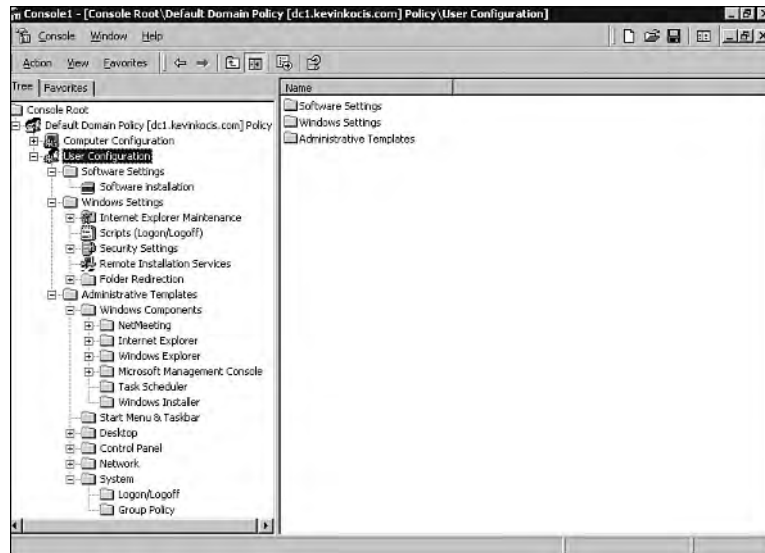
## NOTE

Computer policy takes precedence over conflicting user policy.

## User Policy

The User Configuration settings are similar to the Computer Configuration settings. User Configuration policies can change Registry settings within HKEY\_CURRENT\_USER. The User Configuration policies are applied to any computer that a user logs on to and will follow

a user around the enterprise. Many of these settings are similar in content to the Computer Configuration set, but this interface has many more settings in the User Configuration set, as shown in Figure 6.5.



**FIGURE 6.5**

*User Configuration group policy in the default domain policy.*

User Configuration settings enable the same interface to appear wherever a user logs on, which is preferred for roving users (such as in a lab setting).

Scripts (which have different settings for users than those for computers) show why a setting is placed under the Computer Configuration as opposed to the User Configuration. Computer settings include startup and shutdown scripts, which run automatically for a computer regardless of any logged on users. User settings include logon and logoff scripts that occur only when a user logs on the network and Group Policy is applied.

## Settings and Templates

The Computer Configuration and User Configuration parent nodes have several child nodes, including the following:

- **Software Settings**—Provides a location for independent software vendors (ISVs) to add further extensions. If no nodes have been added by ISVs, the Software Settings node contains just the Software Installation extension included with Windows 2000.

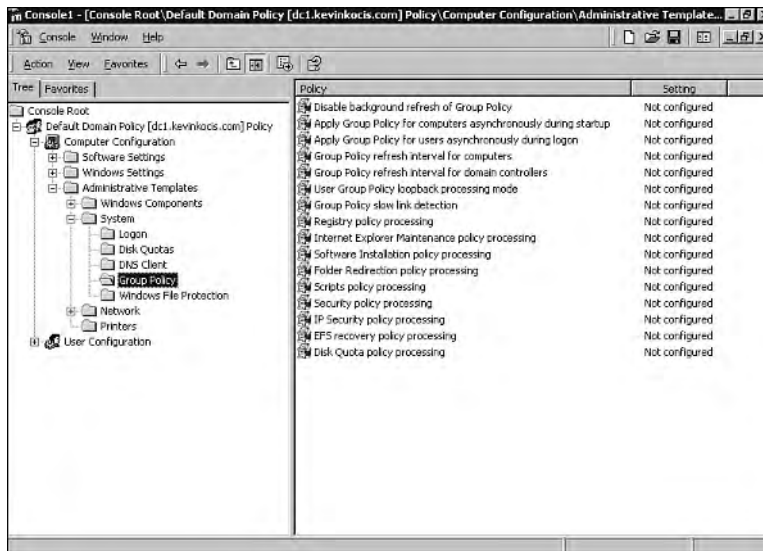
- Windows Settings—Holds Microsoft extensions.
- Administrative Templates—Shows namespaces for Registry-based policy settings. You create the Administrative Templates namespace by adding .adm files. You do so by right-clicking either of the Administrative Templates nodes and then clicking Add/Remove Templates.

## Administrative Templates

Administrative templates include Registry-based Group Policy, which you use to enforce Registry settings that administer the behavior and appearance of the client desktop, including the operating system components and applications. More than 450 of these settings are available for configuration, and you can add more by using .adm files. In Windows 2000, the Administrative Templates node of the Group Policy snap-in uses an administrative template (.adm) file to specify the Registry settings you can modify through the Group Policy snap-in user interface.

To avoid Registry tattooing discussed earlier, you should place any additional Registry settings in `\Software\Policies` or `\Software\Microsoft\Windows\CurrentVersion\Policies`.

Figure 6.6 shows some Administrative Template Group Policy settings. The Policy pane lists some policy settings that make up the User Configuration part of the Default Domain Policy of the Group Policy Object.



**FIGURE 6.6**

Administrative Template Group Policy settings.

The Administrative Templates nodes of the Group Policy snap-in present Registry-based Group Policy settings that are written to the HKEY\_CURRENT\_USER or HKEY\_LOCAL\_MACHINE portion of the Registry database, as appropriate.

The .adm file in Windows 2000 is a Unicode text file that specifies a category hierarchy that defines how the options are displayed through the Group Policy interface. Unicode support for .adm files is new in Windows 2000. The .adm file also specifies the Registry locations where you need to make changes if a particular selection is made.

The Administrative Templates nodes of the Group Policy snap-in can be extended by using custom .adm files.

The following are some of the common templates included in Windows 2000:

- System.adm—Includes common configuration options and settings for Windows 2000 clients
- Common.adm—Includes settings common to Windows 9x and NT4 computers
- Inetres.adm—Contains policy options for configuring Internet Explorer for Windows 2000 clients
- Windows.adm—Contains settings for Windows 9x computers
- Winnt.adm—Contains settings for Windows NT clients

## Security Settings

You use the Security Settings extension to set security options for computers and users within the scope of a GPO. You can define local computer, domain, and network security settings.

The Security Settings extension of the Group Policy snap-in complements existing system security tools such as the Security tab on the Properties page (of an object), and Local Users and Groups in Computer Management.

You can configure these security areas for computers in the Security Settings extension:

- Account Policies—computer security settings for password policy, lockout policy, and Kerberos policy in Windows 2000 domains. Set only at the domain level.
- Local Policies—security settings for audit policy, user rights assignment, and security options.
- Event Log—controls settings for viewing logs in Event Viewer.
- Restricted Groups—manages control and membership of security-sensitive groups, such as Human Resources.
- System Services—controls startup mode and access permissions for system services.
- Registry—configures security settings for Registry keys, including access control, audit, and ownership.

- **Files System**—configures security settings for file system objects, including access control, audit, and ownership.
- **Public Key Policies**—configures PKI security settings, including settings for certificate services and trusts.
- **IPSec Security Policies on Active Directory**—configures security settings for Active Directory's IPSec.

To view or edit the security settings for a GPO, perform the following steps:

1. In the Group Policy console, double-click the appropriate GPO.
2. Expand the Computer Configuration, Windows Settings, then Security Settings containers if necessary.
3. Double-click a security policy node (for example, Account Policies), and select a security area (for example, Password Policy).
4. Double-click the security attribute you wish to view or edit, make the appropriate change, and click OK.

## Software Installation

You use the Software Installation snap-in to centrally manage software in your organization. You can assign and publish software to users, and assign (but not publish) software to computers. Only Windows 2000 clients with the client-side extension for Software Installation (appmgmts.dll) can take advantage of software installation.

### NOTE

You cannot assign software to a domain controller in Active Directory.

You can only deploy software using Software Installation if the file is one of the following:

- **Windows Installer package (.msi)**—a native installation package that optimally utilizes Windows Installer.
- **Modified Windows Installer package (.mst)**—a modified or customized version of an application. Modifications are also considered to be transforms, thus the .mst name.
- **An application setup file (.zap)**—a basic setup file which uses the setup.exe program to install software.

**NOTE**

You should consider setting up a software distribution point on a domain controller. This will alleviate excessive browsing for software installation files, and provide centralization. You can use the \\server\share path.

**Assigning Applications**

When an application is assigned to a user, an advertisement is given when he or she logs on to the computer. When the user selects the application from the Start menu, or launches a document associated with the application, the application is then installed.

To assign an application, perform the following steps:

1. In the Group Policy console, double-click the appropriate GPO.
2. Expand the Computer or User Configuration (depending on which is applicable), Software Settings, then Software Installation.
3. Right-click in the details pane, click New, and then click Package.
4. In the Open dialog box, click the appropriate Windows Installer package (or Browse if necessary), and click Open.
5. In the Deploy Software dialog box, click Assigned, then click OK.

When you assign software to a computer, the installation process typically occurs when the computer starts up to avoid any competing processes.

**Publishing Applications**

Published applications are not as obvious as assigned applications. Instead, attributes such as the application's name and file associations are stored in Active Directory. The application can then be installed by using the Add/Remove Programs utility in Control Panel, or by launching a file associated with the software (such as a .doc file for Microsoft Word).

To publish an application, perform the following steps:

1. In the Group Policy console, double-click the appropriate GPO.
2. Expand the Computer or User Configuration (depending on which is applicable), Software Settings, then Software Installation.
3. Right-click in the details pane, click New, and then click Package.
4. In the Open dialog box, click the appropriate Windows Installer package (or Browse if necessary), and click Open.
5. In the Deploy Software dialog box, click Published, then click OK.

Published applications are available for installation either by using Add/Remove Programs in Control Panel, or by opening a file with a file name extension that you have associated with the application.

**NOTE**

Packages can only be published to users. They cannot be published to computers.

**Modifying Software Installation**

You can also modify how software is installed, including upgrading, removing and setting permissions for software.

To modify Software Installation, perform the following steps:

1. In the Group Policy console, double-click the appropriate GPO.
2. Expand the Computer or User Configuration (depending on which is applicable), Software Settings, then Software Installation.
3. Right-click the Software Installation node, and select Properties.
4. In the General tab, modify the necessary attributes for package location, deployment and uninstallation options.
5. In the File Extensions tab, you can automate software installation based on file name extension. Select the appropriate file extension and prioritize the respective application in the Application Precedence list box.
6. In the Categories tab, you can create or change the category list under which programs would appear in the Add/Remove Programs in Control Panel.

To upgrade applications, perform the following steps:

1. In the Group Policy console, double-click the appropriate GPO.
2. Expand the Computer or User Configuration (depending on which is applicable), Software Settings, then Software Installation.
3. Right-click the Software Installation node, and select Properties.
4. Click the Upgrade tab, then click Add to create or add to the list of packages that are to be upgraded.
5. In the Add Upgrade Package dialog box, choose either Current Group Policy object or a specific GPO, then select the package to upgrade.

6. Select whether you want to uninstall the existing package first, or if you want to upgrade over the existing package, and click the proper button.
7. If you want to make the upgrade mandatory, enable the Required upgrade for existing packages check box, then click OK.

To remove a installed application, perform the following steps:

1. In the Group Policy console, double-click the appropriate GPO.
2. Expand the Computer or User Configuration (depending on which is applicable), Software Settings, then Software Installation.
3. Right-click on the application you wish to remove in the details pane, click All Tasks, and then click Remove.
4. In the Remove Software box, select to either immediately uninstall or to leave the application installed but disallow new installations, and then click OK.

To set permissions for software installation, perform the following steps:

1. Right-click the desired GPO.
2. Click Properties, and then select the Security tab.
3. Click the security group on which you want to set permissions.
4. If the security group has administrative powers, set Full Control to Allow. If the group represents users, set both Apply Group Policy and Read to Allow.
5. In the GPO property window, click OK.

Removing and modifying software installation in Group Policy allows you to conveniently recover from an erroneous install by allowing you to configure optional and mandatory upgrades or removals.

## Scripts

You can use scripts to automate computer startup and shutdown and user logon and logoff sessions. You can use any Windows Script Host-supported language, including VBScript, JavaScript, Perl, and DOS batch files (.bat and .cmd).

Windows 2000 includes Windows Script Host (WSH), a language-independent scripting host for 32-bit Windows platforms. WSH has low memory requirements and serves as a controller of ActiveX scripting engines. With WSH, you can run scripts directly in Windows 2000 by double-clicking a script file or by typing the name of a script file at the command prompt.



In Windows 2000, the following five script types are supported:

- Legacy Logon scripts
- Group Policy Logon scripts
- Group Policy Logoff scripts
- Group Policy Startup scripts
- Group Policy Shutdown scripts

To set up scripts on the domain controller, copy the script and any dependent files to the Netlogon share (or share of your choice) of the domain controller from which you want the script to run.

To assign computer startup or shutdown scripts, perform the following steps:

1. Open the Group Policy snap-in.
2. Select the GPO and open Computer Configuration, Windows Settings, Scripts (Startup/Shutdown).
3. In the details pane, double-click the Startup or Shutdown icon.
4. In the Startup or Shutdown properties page, click Add.
5. In the Add a Script dialog box, select the options you want to use, and then click OK.
6. In the Startup or Shutdown properties page, select any options you want to use, and then click OK.

**NOTE**

Startup and Shutdown scripts are run as Local System.

To assign user logon or logoff scripts, perform the following steps:

1. Open the Group Policy snap-in.
2. Select the GPO and open Computer Configuration, Windows Settings, Scripts (Logon/Logoff).
3. In the details pane, double-click the Logon or Logoff icon.
4. In the Logon or Logoff properties page, click Add.
5. In the Add a Script dialog box, select any options you want to use, and then click OK.
6. In the Logon or Logoff properties page, select any options you want to use, and then click OK.

**NOTE**

Logon scripts are run as User, not Administrator.

## Remote Installation Services

You use Remote Installation Services (RIS) to control the behavior of the Remote Operating System Installation feature as displayed to client computers.

RIS is an optional component of the Windows 2000 Server operating system that you can use to set up new client computers without attending physically to each client computer.

To install RIS, perform the following steps:

1. Open Administrative Tools, and select Configure Your Server.
2. In the Configure Your Server dialog box, click Finish setup.
3. In the Configure Remote Installation Services dialog box, click Configure to start the Remote Installation Setup wizard.
4. In the Remote Installation Setup wizard dialog box, click Next.
5. In the Remote Installation Services Setup wizard, you are prompted for the following information:
  - RIS drive and directory
  - Windows 2000 Professional Source Path
  - Friendly Description and Help Text

You can also launch the Remote Installation Services Setup wizard by clicking Start, Run, and typing Rlsetup.

In order to install RIS, you must have DNS and DHCP services available on your network. RIS must be installed on an NTFS-formatted drive other than the one containing the Windows 2000 Server operating system.

**NOTE**

RIS currently does not support the Encrypting File System (EFS) or the distributed file system (Dfs).

To configure Remote Installation Services, perform the following steps:

1. Open Active Directory Users and Computers.
2. In the console tree, open the desired domain or container and right-click the applicable remote installation server.
3. Click Properties, and then in the Properties dialog box, click the Remote Install tab.
4. In the Remote Install dialog box, select one of the following options:
  - Respond to client computers requesting service
  - Do not respond to unknown client computers

To configure Remote Installation Services advanced settings, perform the following steps:

1. Open Active Directory Users and Computers.
2. In the console tree, open the desired domain or container and right-click the applicable remote installation server.
3. Click Properties, select the Remote Install tab, and then click Advanced Settings.
4. In the Advanced Settings dialog box, select the New Clients tab.
5. Click the client computer naming format or click Customize to create a client computer naming format.
6. Click one of the following options to determine where the client computer account is created:
  - Default directory service location
  - Same location as that of the user setting up the client computer
  - The following directory service location
7. If you chose the last option, click Browse and specify where to create the computer accounts.

**NOTE**

If there are multiple remote installation servers on your network, each one must be configured to identically respond to client computers.

To manage client installation images, perform the following steps:

1. Open Active Directory Users and Computers.
2. In the console tree, open the desired domain or container and right-click the applicable remote installation server.

3. Click Properties, select the Remote Install tab, and then click Advanced Settings.
4. Select the Images tab, and then click the installation image or unattended setup answer file.
5. Select one of the following options:
  - Add
  - Remove
  - Properties
  - Refresh

## Internet Explorer Maintenance

You use Internet Explorer Maintenance to administer and customize Internet Explorer on Windows 2000-based computers after their deployment. For example, you may want to restrict users from altering connection settings for Internet Explorer. You could remove these settings from the user interface. You could also alter the settings for offline pages and ActiveX controls. These settings can be applied on either a per-computer or per-user basis.

To modify Internet Explorer maintenance features, perform the following steps:

1. Start Group Policy.
2. Select the GPO and open User or Computer Configuration, Windows Settings, Internet Explorer Maintenance.
4. Click the icon for the desired feature area you want to customize.
5. Double-click the feature title you want to administer.
6. Modify the desired settings, and click OK.

### NOTE

For most Internet Explorer templates, clicking **Enabled** sets a restriction, and clicking **Disabled** prevents the restriction from applying to that group of users or computers. If you select **Not Configured**, the restriction is not applied.

## Folder Redirection

Folder redirection allows you to redirect the path of a local computer folder to a server location. This strategy allows users to work with individual or shared documents on a secure server as if the folders were on the local drive. Windows 2000 special folders include My Documents (which includes My Pictures), Application Data, Desktop, and Start Menu. These folders are a common location for users to store data, and are located in the Documents and Settings user profile folder on the local computer.

**NOTE**

You should not use folder redirection if the portable computer is not connected to the network very often, or is usually remotely connected (such as for sales people or field engineers). Folder redirection is best suited for non-portable computers or portable computers that move between locations on a high-speed network.

**6**

To redirect special folders to one location for everyone in the site, domain, or organizational unit, perform the following steps:

1. Open the GPO linked to the site, domain, or organizational unit containing the users whose special folders you want configured for folder redirection.
2. In the console tree, open User Configuration, Folder Redirection.
3. Right-click the special folder for redirection (such as My Documents), and click Properties.
4. In the Target tab, click on the Setting drop-down menu, select Basic (for the same location), enter the network path (UNC path) or click Browse, and browse to the desired location. Click OK.

If you want each user to maintain a subfolder at this location, use `%username%` into the UNC path, such as `\\Dc2\My Computer files\%username%`.

5. In the Settings tab, you can set various options, although the defaults are recommended. Click OK.

To redirect special folders to different locations according to security group membership, perform the following steps:

1. Open the GPO linked to the site, domain, or organizational unit containing the users whose special folders you want configured for folder redirection.
2. In the console tree, open User Configuration, Folder Redirection.
3. Right-click the special folder for redirection (such as My Documents), and click Properties.
4. In the Target tab, click on the Setting drop-down menu, select Advanced, then click Add.
5. Enter or browse for the desired security group and target folder location (use the same naming standards mentioned in the previous redirection example), then click OK. (You can repeat these steps as necessary to add all the appropriate security groups.)
6. In the Settings tab, you can set various options, although the defaults are recommended. Click OK.

## Special Policies (Account Policies)

Special policies are highly configurable computer security settings for password policy, lockout policy, and Kerberos policy in Windows 2000 domains. Account policies can be configured by opening the GPO, Computer Configuration, Windows Settings, Security Settings, Account Policies.

### NOTE

Windows 2000 allows only one domain account policy that is applied to the root domain of the domain tree. However, you can apply account policies to OUs that contain computers. If an OU only contains users, it can obtain account policies from the domain account policy.

These policies apply to user accounts:

- Password policy—applies to domain or local user accounts. It determines settings for passwords such as enforcement, and lifetimes.
- Account lockout policy—applies to domain or local user accounts. It determines the parameters for an account to be locked out of the system.
- Kerberos policy—applies only to domain user accounts. It determines Kerberos-related settings, such as ticket lifetimes and enforcement.

## Linking GPOs

As you learned earlier, a GPO can be applied at the site, domain, and OU levels. This is accomplished by linking the GPO to a container or to multiple container objects.

To link a GPO with an organizational unit, perform the following steps:

1. Open the Active Directory Users and Computers tool.
2. Locate the OU for linking, right-click it, and select Properties.
3. Select the Group Policy tab and click Add.
4. In the drop-down list, select the name of the local domain.
5. Select the desired group policy (in Figure 6.7, the Sales Field group policy is selected) and click OK.
6. To save the GPO link, click OK.

**FIGURE 6.7**

*Linking the Sales Field policy to the Sales Field OU.*

If you want to see what a Group Policy Object is linked to, open it in the Group Policy console, right-click the root node, click Properties, and then click the Links tab. Click Find Now after setting the domain on the drop-down list.

## Inheritance

By default, Group Policy Object settings flow from parent to child containers, and include all settings for computer and user objects in each container. Inheritance can be combined with delegation to grant administrative rights to directory subtrees.

### Blocking Policy Inheritance

A group policy inherited from a higher level can be blocked so that it does not pass further down the hierarchy. Blocking Inheritance is usually set at the OU level to prevent certain GPO settings from “trickling down” from a parent container GPO.

To block a policy, perform the following steps:

1. Right-click the container linked to the GPO, and select Properties.
2. Click the Group Policy tab.
3. Check the Block Policy Inheritance box at the bottom of the dialog, as shown in Figure 6.8.

**FIGURE 6.8**

*Blocking policy inheritance in the Group Policy tab.*

## NOTE

The Block Policy Inheritance policy setting is not available for sites.

## Enforcing Group Policy

If conflicts occur between a policy setting in two different GPOs, the last GPO applied overrides the setting applied previously. An administrator can stop a policy from being blocked (overridden).

To enforce a policy, perform the following steps:

1. Right-click the site, domain, or OU, and select Properties.
2. Click the Group Policy tab.
3. Select Options.
4. Check the No Override box (see Figure 6.9).

**FIGURE 6.9**

*Forcing a policy.*



## Comparing No Override and Blocking

You can set No Override on a specific GPO link so that GPOs linked at a child container cannot override that specific policy. In this situation, GPOs linked at the same level, but not as No Override, are also prevented from overriding. If you have several links set to No Override at the same level of Active Directory, you need to prioritize them. Links higher in the list have priority on all Configured (Enabled or Disabled) settings.

If GPO is linked to a domain and is set to No Override, the configured Group Policy settings will apply to all containers in that domain.

### NOTE

GPOs linked to organizational units and set with the No Override option cannot override a domain-linked GPO.

The following are some comparisons and facts regarding No Override and Block Policy:

- No Override is set on a link, not on a site, domain, organizational unit, or GPO.
- Block Policy Inheritance is set on a domain or organizational unit and applies to all GPOs linked at that level or higher in Active Directory which can be overridden.
- No Override takes precedence over Block Policy Inheritance if conflicts occur.

For performance reasons, both No Override and Block Policy should be used sparingly.

## Delegation of GPO Administration

Group Policy is one of the administrative tasks that can be delegated in Windows 2000. The following three Group Policy tasks can be independently delegated:

- Managing Group Policy links for a site, domain, or organizational unit
- Creating Group Policy Objects
- Editing Group Policy Objects

One example of delegation is granting a non-administrative user permission to create a new GPO. This permission is often useful in combination with the right to create links. To allow for creation of new GPOs, you need to add the user to the Group Policy Creator Owners administrators group as follows:

1. In Active Directory Users and Computers, open the Users container in the domain root.
2. Right-click Group Policy Creator Owners, and select Properties.

3. On the Properties page, select the Members tab.
4. Click Add and then add the approved user to the security group.

As you learned earlier, a user in this group can create new GPOs and becomes the Creator Owner of that GPO.

#### NOTE

When linking a GPO, anyone assigned the task of creating a GPO must have permission not only to create it but also to link it to the domain or organizational unit, otherwise they will be restricted from creating a new GPO.

To delegate administrative right to create GPO links, perform the following steps:

1. Open Active Directory Users and Computers.
2. Expand the local domain in the console pane and right-click the domain.
3. Select Delegate Control.
4. Click Next to start the Delegation of Control Wizard.
5. On the Users and Groups page, click Add.
6. Select the users, group, or computer account you want to delegate administration to.
7. On the Tasks to Delegate window (see Figure 6.10), select the Manage Group Policy links check box, and click Next.
8. Click Finish to complete the delegation process.



**FIGURE 6.10**

*Delegation of group policies.*

## Mixed Mode Group Policy

Running Group Policy in mixed mode has various effects. Let's take a look at some examples.

It is best not to use System Policy on Windows 2000 clients. You can uncheck Show Policies Only so that true Group Policy settings appear in blue, and System Policy settings appear in red. The next time you run the Group Policy snap-in, non-Group Policy settings will be hidden again.

## Upgrading the Computer Accounts

Windows NT 4.0 Registry settings (the tattooing mentioned earlier) can be an issue when you're upgrading computer accounts from Windows NT 4.0 to Windows 2000. While the client computer was subject to the System Policy, it may have received Registry settings outside the approved Group Policy trees, and they are not removed from the client when the computer is upgraded.

You should look for unwanted residual effects of the System Policy and take corrective steps, such as using Regini.exe, which you can find in %systemroot%/System32/, to remove the old settings.

### NOTE

The best option is to perform a clean installation of Windows 2000 as opposed to upgrading to avoid any legacy or tattooed Registry settings.

## Group Policy Permissions

The default permissions on Group Policy Objects are as shown in Table 6.1.

**TABLE 6.1** Default Permissions for GPOs

<i>Security Group</i>	<i>Default Settings</i>
Authenticated users	Read, Apply Group Policy (AGP)
Local system	Full Control (includes AGP)
Domain administrators	Read, Write, Create Child, Delete Child
Administrators	Read, Write, Create Child

By default, the Default Domain Policy Group Policy Object cannot be deleted by any administrator, which prevents the accidental deletion of this critical GPO.

A user or administrator who does not have Write access (but does have Read access) to a GPO cannot use the Group Policy snap-in to view its settings. Every extension to Group Policy assumes that it has Write access to where the GPO is located.

## Filtering and Delegating Group Policy with Security Groups

You use security groups in Group Policy for two purposes:

- To filter the scope of a Group Policy Object
- To delegate control of Group Policy

The following sections explain how these items work.

### Filtering the Scope of a Group Policy Object

You can filter which groups of computers and users a particular GPO influences by using security groups. To do so, use the Security tab on the Properties page of the Group Policy Object.

Filtering affects the GPO as a whole—you cannot restrict access to only some of the settings in a GPO. Two exceptions exist however: Folder Redirection and Software Installation, which have further Access Control Lists (ACLs) set at the GPO level.

### Active Directory-Based Storage

Non-Local Group Policy Objects store Group Policy information in two locations: a Group Policy container and a Group Policy template. They are named with a globally unique identifier (GUID) to maintain synchronicity.

### Group Policy Container

The Group Policy container stores both computer and user Group Policy information and has the following properties:

- Version information, which ensures that the information is synchronized with the Group Policy template information
- Status information, which indicates whether it is enabled or disabled
- List of components (extensions) that have settings in the Group Policy Object
- Policy settings as defined by the extension snap-ins

### Group Policy Template

GPOs also store information in the domain controller System Volume folder (SYSVOL). The template file resides in the Policies subfolder and contains various policy settings and script files.

**NOTE**

Group Policy is backed up with Active Directory.

**6**

## Group Policy Template Subfolders

The Group Policy template folder can contain a number of subfolders but always contains at least a machine and user folder.

The machine folder includes a Registry.pol file that contains the Registry settings that are applied to computers. The Registry.pol file is downloaded and applied to the HKEY\_LOCAL\_MACHINE Registry hive when a computer is initialized.

The user folder includes a Registry.pol file that contains the Registry settings that are applied to users. This file is downloaded and applied to the HKEY\_CURRENT\_USER Registry hive when a user logs on to a computer.

The Group Policy template folder also includes a Gpt.ini file. For Local Group Policy Objects, the Gpt.ini file stores information indicating

- Which client-side extensions of the Group Policy snap-in contain User or Computer data in the Group Policy Object
- Whether the User or Computer portion is disabled
- The Group Policy snap-in extension version number

## Multiple Group Policy Objects

Each non-Local GPO is stored in a specific domain, called the storage domain, which should not be confused with a domain link to the GPO.

You can link multiple GPOs to a single site, domain, or organizational unit. Also, multiple sites, domains, and organizational units can obtain policy from one GPO by linking to it, regardless of which domain hosts the GPO.

**NOTE**

For performance reasons, you should avoid linking to a GPO in a different domain.

## Trust Relationships with Previous Versions of Windows

You need to avoid a subtle migration issue related to how trusts are handled in Windows NT 4.0 and how this relates to Windows 2000 upgrades.

Suppose you have a Windows 2000 domain controller (DC1) with a previous version trust relationship to a Windows NT 4.0 domain controller (DC2). You upgrade DC2 to Windows 2000 and then link an organizational unit managed by A to a Group Policy Object stored in DC2's domain. A user in the organizational unit logs on to DC1 expecting to receive a policy from the Group Policy Object stored in DC2's domain, but it doesn't work because the upgrade of the domain controller does not automatically upgrade the trust relationship, and the user doesn't have access to the SYSVOL share on DC2.

To solve this problem, you need to break the trust after upgrading DC2 to Windows 2000. Then you can create a new Windows 2000-trust, and the user then receives Group Policy.

## Summary

This chapter addressed how system policies introduced in Windows NT 4.0 evolved into group policies in Windows 2000. Using the new Group Policy MMC snap-in, you can configure Group Policy Objects and delegate to local administrators. This chapter also looked at how policies could be enforced or blocked and how mixed environments could affect policy assignment.

# Managing and Modifying Active Directory Schema

CHAPTER

7

## IN THIS CHAPTER

- Active Directory Schema Fundamentals 204
- Schema Structure: Exploring the Directory Information Tree 206
- Schema Modification 212
- Deactivating Schema Objects 223

Managing the Active Directory schema is a topic that could fill an entire book. I will not attempt to write that book in this chapter but instead will provide some strong administration fundamentals (combined with some tools found in the glossary) and some light scripting details. Again, one could write a book on scripting Active Directory. Let's take a look at how to modify, manage, and understand the impacts of changing the Active Directory schema.

## Active Directory Schema Fundamentals

The Active Directory schema contains definitions and rules for all the objects stored in the directory. The schema maintains object creation consistency, and is composed of classes, attributes, and syntaxes.

### NOTE

Active Directory contains a default set of classes and attributes that you cannot modify. However, additional classes and attributes can be added. This is called extending the schema.

The schema is composed of object classes, attributes and syntaxes.

A class is a category of objects that share a set of common characteristics. A directory object such as a computer account is an instance of a schema class (a computer class, to be exact). Each object in the directory is an instance of one or more classes in the schema. Also, each object class contains mandatory and optional attributes.

An attribute describes the characteristics of some aspect of an object, such as model information (Dell, Compaq, IBM) for our computer class. Attributes define the types of information contained by the object.

A syntax is the data type or format of a particular attribute. It determines what data type an attribute can contain. Active Directory uses a set of predefined, standard syntaxes, which do not appear in the directory.

### NOTE

You cannot add new syntaxes to the Active Directory schema.

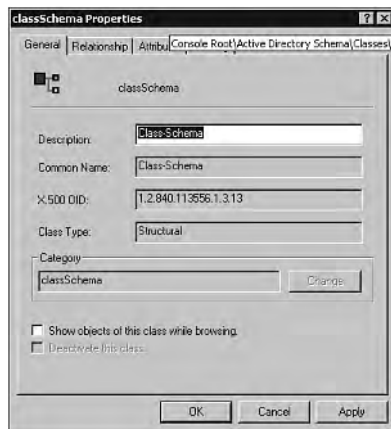


For example, a vehicle object can belong to the class of airplanes, the class of motorcycles, or the class of cars, and so on. A motorcycle can be described by its make, model, and color. These would be the attributes of the motorcycles. The possible values for the color of the car might be black or blue, and the syntax for black might be represented by a value such as COM0Y0K100 (if you're familiar with printing colors) that indicates a specific color combination.

The schema specifies the relationships between classes of objects. For example, let's say the schema contains a class called User, and the user accounts of John and Chris are objects in the directory that are instances of this class. The object Chris might contain an optional attribute defined for this class called homePhone. This attribute for the object Chris of the class User might have the value 555-5500.

The attribute homePhone can be defined to take values of the syntax String(numeric), which means that the value can contain only the numbers 0 through 9.

The schema itself is represented in Active Directory by a set of objects known as schema objects. For each schema class, there is a schema object that defines the class, which is called a classSchema object. For each schema attribute, there is also a schema object that defines the attribute, which is called an attributeSchema object. Therefore, every class is actually an instance of the classSchema class, and every attribute is an instance of the attributeSchema class. Figure 7.1 shows the classSchema Properties window in Active Directory.



**FIGURE 7.1**

*The classSchema Properties window in Active Directory Schema.*

Administrators and applications can extend the schema by adding new attributes and classes or by modifying existing ones. Schema definitions are required by applications that need to create or modify Active Directory objects. Such applications are considered to be “directory-enabled,” meaning they recognize the attributes and syntaxes required to interact with the directory.

## Schema Structure: Exploring the Directory Information Tree

The objects stored in Active Directory are arranged in a logical hierarchy called the Directory Information Tree (DIT). The base DIT is the preconfigured database installed during a fresh install of a Windows 2000 domain controller. One section of the base DIT is the base schema.

The Directory Information Tree is divided into directory partitions, which are units of replication in Active Directory.

Active Directory is composed of three partitions:

- Domain directory partition
- Configuration partition
- Schema partition

The Schema partition hosts all the schema objects, and is not a container in terms of an Active Directory object that contains other objects. The Schema container (cn=schema,cn=configuration,dc=< forest root domainName>) is the topmost object of the schema directory partition, and contains all the class and attribute definitions required to locate objects in Active Directory and to create new objects.

Every Active Directory object can be referenced by a unique and unambiguous name known as the distinguished name (DN). The distinguished name identifies the object and its complete path up through the Active Directory container hierarchy. The distinguished name of the Schema container can be expressed as follows:

```
cn=schema,cn=configuration,dc=< forest root domainname>
```

For more information about the distinguished name and naming in general, see Chapter 3, “Managing Domains, Trusts, and DNS.”

You can view the contents of the Schema container by using the Active Directory Schema MMC console. You also can bind to the schema directory partition and view schema objects by using the Active Directory Service Interfaces (ADSI) Edit MMC console or the Ldp tool.

**NOTE**

The ADSI Edit snap-in is not one of the default MMC snap-ins provided with Windows 2000 Server. To use ADSI Edit, you must install the Support Tools located in the Support\Tools folder on the Windows 2000 Server CD.

## Starting the Active Directory Schema Snap-In

The Active Directory Schema snap-in has no saved Schema console or Administrative Tool on the Administrative Tools menu because schema management is not frequently performed. You must load the Schema Manager manually into MMC.

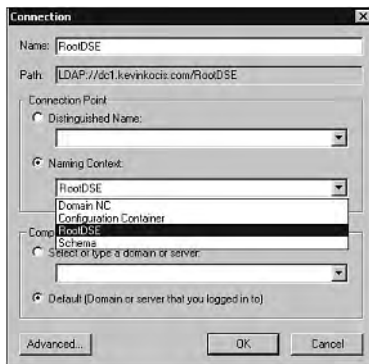
To start the Active Directory Schema snap-in, perform the following steps:

1. Click Start, Run, and type **regsvr32.schmmgmt.dll** in the Open box. Click OK.
2. Click OK at successful operation window.
3. Click Start, Run, and type **MMC** in the Open box.
4. On the Console menu, click Add/Remove Snap-in, click Add, and then click Active Directory Schema. Click Add, click Close, and then click OK.
5. You can save the MMC console containing the Schema snap-in. On the Console menu, click Save As, and type a name for the saved console (for example, Schema.msc). Click Save.

Some installation scripts and applications may require access to the schema. This raises the issue of locating the schema. These scripts and applications may not be aware of which domain they are to be used in but can still gain access to the schema. The applications bind to a special entry at the top of the logical namespace called rootDSE, which provides the schema location. The rootDSE logically represents the top of the namespace and the LDAP search tree. The attributes of rootDSE identify the directory partitions, and allow applications to locate and read the schema.

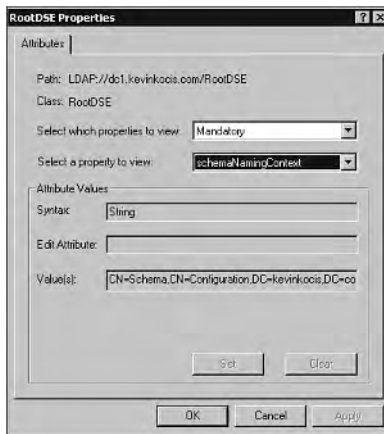
To identify the Schema directory partition by using ADSI Edit, perform the following steps:

1. Add the ADSI Edit snap-in console to MMC.
2. Right-click ADSI Edit and then click Connect to.
3. In the Connection Point check box, make sure that Naming Context is selected.
4. Select RootDSE from the drop-down list and then click OK (see Figure 7.2).

**FIGURE 7.2**

*Selecting rootDSE in the ADSI Connection window.*

5. In the Console Tree, expand the RootDSE container.
6. Right-click the RootDSE folder and then click Properties.
7. In the Select a Property to View box, select schemaNamingContext from the list of properties (or “attributes”), as shown in Figure 7.3.
8. In Attribute Values area, view the Value(s) box to see the distinguished name of the schema directory partition.

**FIGURE 7.3**

*Setting the schemaNamingContext in the rootDSE Properties window.*

**NOTE**

In order to view the schema from a non-Windows 2000 server, you must install the admin tools package, called adminpak.msi from the Windows 2000 Server CD.

Every Active Directory domain controller holds a copy of the schema as a file named Ntds.dit. The location of the Ntds.dit file is set during the promotion process (dcpromo.exe) or during the first Windows 2000 domain controller installation. The default location is %SystemRoot%\Ntds.

Another file, the Schema.ini initialization file contains data required to create the default directory objects, the default security for the DIT, and the Active Directory display specifiers. The Schema.ini file is located in the base version of Ntds.dit.

## Active Directory Schema Objects

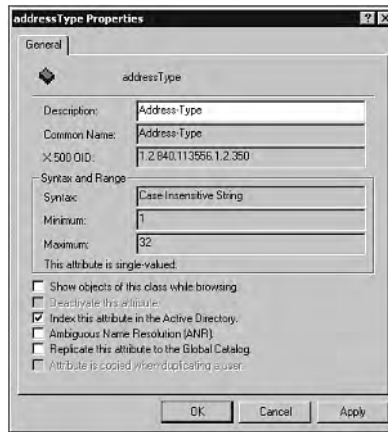
As mentioned earlier, Active Directory attributes and classes are stored in the Schema container as directory objects called schema objects. They are called attributeSchema and classSchema respectively. Let's take a closer look at these two schema objects.

### Attributes

Attributes are attributeSchema objects, and contain information in relation to the attribute, including:

- The LDAP display name
- The object identifier (OID)
- The globally unique identifier (GUID)
- The syntax
- The attribute range, which is the minimum and maximum value or length
- Whether the attribute is single- or multi-value
- Indexing features, which allow the attribute to be searched or referenced more efficiently

To index an attribute in the Schema snap-in, right-click the attribute and select Properties. Then click the check box Index this Attribute in Active Directory, as shown in Figure 7.4.

**FIGURE 7.4**

*Configuring the addressType attribute for indexing in Active Directory.*

## Mandatory Attributes

Mandatory attributes are object attributes requiring values. If you do not specify a value for a mandatory attribute, one of the following happens:

- The attribute inherits a default value.
- The object is not created (at least until you specify a value).

The object class determines which of the object's attributes are mandatory. Some of these mandatory attributes are inherited from parent classes.

## Attribute Syntax

The syntax for an attribute defines the way it is stored and the rules for attribute comparison. Syntax determines whether the attribute value must be a string, number, or unit of time. Every attribute of every object is associated with exactly one syntax. The allowable syntaxes in Active Directory are predefined, and no new syntaxes can be added.

## Classes

The classSchema object defines the various class attributes including a list of mandatory (mustContain) and optional (mayContain) attributes, as well as the hierarchical rules that determine DIT parent classes.

An object can have only attributes that belong to either the mustContain or the mayContain list for the class. After an object has been created, the object's class can never be changed.

The classSchema object dictates the rules for creating objects in an Active Directory class. When a new object is created in a class, the classSchema object ensures it has the same attributes as all other objects in the class.

The classSchema object contains the following information in regards to a class:

- The LDAP display name
- The object identifier (OID)
- The globally unique identifier (GUID)
- Mandatory attributes (mustContain)
- Optional attributes (mayContain)
- Hierarchy rules for parent classes
- The type of object class (Abstract, Structural, or Auxiliary)

### Object Class Categories

The X.500 1993 specification requires that object classes be assigned to one of four categories or types:

- Structural—are the only classes allowed instances in the directory, and can be used in defining the directory structure. Structural classes can include multiple auxiliary classes, and are specified by a value of 1 in the objectClassCategory attribute.
- Abstract—are used to derive new structural classes. Abstract classes are not allowed to have multiple instances, and can be derived from any existing abstract class. The only function of abstract classes is to provide attributes for subordinate classes. They are specified by a value of 2 in the objectClassCategory attribute.
- Auxiliary—contain attribute lists that are appended to structural and abstract classes, and are not allowed to have multiple instances in the directory. They can be derived from existing auxiliary classes, and are specified by a value of 3 in the objectClassCategory attribute.
- 88—are classes that were defined prior to the 1993 standards, and do not fall into the three previous categories. This class is specified by a value of 0 in the objectClassCategory attribute.

For example, the schema object named contact is a structural object type. By default, it has a mandatory attribute (cn) and optional attribute (notes).

#### NOTE

When you define new schema classes, do not define new 88 classes. You need to use one of the X.500 1993 categories (structural, abstract, or auxiliary).

## Schema Modification

The Active Directory schema can be modified three ways. First, you can use the Schema snap-in to modify classes and attributes. Second, you can design scripts to automate schema modification. And third, you can install applications that add classes or attributes to the schema.

Modifying the schema is intimidating, since modifications cannot be deleted. They can, however, be deactivated. You can also modify the schema using Active Directory Services Interfaces (ADSI), or the LDAP Data Interchange Format (LDFIDFE) tool.

You should consider modifying the schema when at least one of the following conditions is met:

- No existing class meets your needs.
- A class requires more specific attributes.
- You require a set of unique attributes to apply to multiple classes.
- Existing classes or attributes are no longer needed.

If all the conditions are in place for schema modification, you can install the Active Directory Schema MMC snap-in to manage the classSchema and attributeSchema objects. The schema can be modified through the addition, deactivation, or modification to any objects or attributes within it. Active Directory then runs a validation process to ensure that integrity is retained throughout the database following the modification.

There are four basic steps you should take prior to modifying the schema:

1. Obtain an approved Object identifier for each new class or attribute you intend to create (see the section Obtaining Valid Object Identifiers later in this chapter).
2. Verify membership in the Schema Admins group.
3. Install Schema Manager.
4. Configure Registry settings to allow modifications.

## Planning to Extend the Schema

When the existing class and attribute definitions in the schema do not meet the needs of your organization, adding or modifying schema objects can extend the schema. Like the rest of Active Directory, schema objects are also protected by ACLs, so only authorized users can modify the schema (see Chapter 5, “Active Directory Security,” for more information).

The schema modification process is similar to adding or modifying any object in Active Directory, except that additional checks are performed to ensure that changes do not cause future inconsistencies or problems.



Modifying the schema is a major change, with implications throughout the directory. Because many schema modifications cannot be reversed, you should modify the schema only when it is absolutely necessary. Make sure changes are well planned before they are implemented. Inconsistencies in the schema can result in significant problems that impair or disable Active Directory, which may not be apparent immediately. The following are the modifications that can be made to the schema:

- Creating classes
- Modifying existing classes
- Creating attributes
- Modifying existing attributes
- Deactivating classes and attributes

There are three ways to effectively add a new class:

- Extending an existing class by adding attributes or additional possible parents
- Deriving a new subclass from an existing class
- Creating an entirely new class with any attributes that you want to assign

You should derive a subclass from an existing class when the following conditions apply:

- The existing class meets your needs but requires additional attributes.
- You want to identify the extended class as distinct from the original class.
- You want to use the Active Directory Users and Computers console to manage the extended attributes of the objects.

## Extending the Schema

Windows 2000 has some safety features, or interlocks, that control modification of the schema. First, schema modification is disabled by default on all domain controllers. You need to use the Active Directory Schema console to permit write access to the schema on that local domain controller. The second safety feature is that the schema object is protected by ACLs. An administrator must be given explicit permissions or be a member of the Schema Administrators group (Schema Admins) to modify the schema. The third interlock is that only one domain controller in the enterprise, the one holding the Schema Master Role (the Schema FSMO), is allowed to write to the schema. For more information on FSMOs, see Chapter 9, “Managing Updates with Flexible Single-Master Operations.”

## Methods for Extending the Schema

There are several ways you can extend the Active Directory schema. You can import and export objects in a batch mode by using each of these administrative tools: LDIF Directory Exchange (LDIFDE), CSV Directory Exchange (CSVDE), and ADSI scripts. These tools

enable you to administer many objects (such as users, contacts, groups, servers, and printers) in one operation. You can also export Active Directory data to other applications and services, as well as import information from other sources into Active Directory using any of these tools. You can also perform schema extension programmatically by using ADSI Edit and the Active Directory Schema console. See the glossary for more explicit information regarding these tools.

To allow a domain controller to modify the schema, use the Active Directory Schema console in MMC on the selected server.

To enable schema modification, perform the following steps:

1. Open the Active Directory Schema console in MMC.
2. Right-click Active Directory Schema (Manager) and select Operations Master.
3. Select The Schema May Be Modified on This Domain Controller check box and then click OK (see Figure 7.5).

The value of this check box is stored in the registry in the Schema Update Allowed entry (in HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS\Parameters). Active Directory adds this entry to the registry when you use the Active Directory Schema console to change the default value.



**FIGURE 7.5**

*Enabling schema modification at the local domain controller.*

## NOTE

Do not use a registry editor to edit the registry for schema extension, but only to verify this modification.

## Obtaining Valid Object Identifiers

When extending the schema, it is important to remember to obtain valid object identifiers (OIDs) from a governing agency. This will ensure that OIDs do not conflict with one another when different directories, such as Active Directory and Novell Directory Services, are brought together in a global directory namespace.

OIDs are typically assigned by a National Registration Authority, or NRA, which may vary from country to country. In the United States, the American National Standards Institute (ANSI) provides NRA services. For a modest fee, ANSI can supply your organization with a root OID. Any objects created by your organization will have this root OID as the prefix, ensuring uniqueness among your OIDs.

A list of NRAs can be found at the International Standards Organization's Web site, at [www.iso.ch](http://www.iso.ch).

OIDs are found in Open Systems Interconnection (OSI) applications, X.500 directories, Simple Network Management Protocol (SNMP), and other applications where uniqueness is important.

Object identifiers in the Active Directory base schema include some issued by the International Standards Organization (ISO) for X.500 classes and attributes and some issued by Microsoft. Object identifier notation is a dotted string of non-negative numbers (for example, 1.2.345.678901.2.3.4.).

## Verifying Schema Admins Membership

To modify the schema, you must use an account that is a member of the Schema Admins group. By default, the only member in that security group is the Administrator account in the root domain of the enterprise. If you want to add other accounts, you have to add them explicitly.

### NOTE

Membership in the Schema Admins group must be highly restricted to prevent unauthorized access to the schema because modifying the schema improperly can have serious consequences.

One way to verify that an account is a member of the Schema Admins group is to use the Active Directory Users and Computers console in MMC.

To verify that an account is a member of Schema Admins, perform the following steps:

1. Open Active Directory Users and Computers.
2. Expand the domain container, then the Users container.
3. Double-click the Schema Admins security group and then click the Members tab.
4. If the account is not listed under Members, click Add.
5. Select an account from the displayed list or type the name of the account.
6. Click Add and then click OK.

### **LDAP Data Interchange Format (LDIF)**

The LDAP Data Interchange Format (LDIF) (file) format has a command-line utility called LDIFDE.EXE that allows you to create, modify, and delete directory objects. It can be run on a Windows 2000–based server or copied to a Windows 2000–based workstation. For example, LDIFDE.EXE can be used to extend the schema, export Active Directory user and group information to other applications or services, and populate Active Directory with data from other directory services.

LDIF is an Internet standard for a file format to perform batch import and batch export operations for directories that conform to LDAP standards. An LDIF file consists of a series of records divided by line separators. A record describes either a single directory entry or a set of modifications to a single directory entry and consists of one or more lines in the file.

### **Order of Processing When Extending the Schema**

If you decide to extend the schema either programmatically or by using scripts, apply updates in the following order:

1. Plan your update on the Schema FSMO Role Owner. Don't change the schema master role between domain controllers for the sake of convenience. To determine the Schema operations master, right-click on the Schema container and select Operations Master. Make changes from this domain controller only.
2. The Schema may be modified on this Domain Controller check box. This enters a Registry value that unlocks the schema for updates. You will reverse this step later in the process to apply security again.
3. Make sure that you are a member of the Schema Admins group. (The Administrator account is automatically a member of the Schema Admins group.)
4. Add any new attributes, and reload the schema.
5. Add any new classes, and reload the schema.
6. Add attributes to classes.
7. Reapply the safety interlock by clearing the check box in step 2 above.

## Schema FSMO Role

Active Directory performs schema updates in a single-master fashion to prevent conflicts. Simultaneous schema updates on two different computers might conflict with each other. The one domain controller in the enterprise that is allowed to perform schema updates at any specific time is referred to as the schema master. Only one domain controller in the entire enterprise, the domain controller holding the schema master role, accepts updates to schema objects. By default, the Schema snap-in is targeted to the schema FSMO role.

For more information on the managing and transferring the schema FSMO role, see Chapter 9.

## Adding an Attribute

It is recommended that you try to use existing attributes wherever possible. If you need to create a new attribute, follow Microsoft's recommended guidelines:

- Use cn as the name (relative distinguished name) attribute, which is the default for most classes. Because cn is an indexed attribute, it allows efficient object name searches.
- Avoid using large multi-value attributes since they are costly to store and retrieve.
- Remember that attributes are “flat,” which means they have no substructure. All attributes in a specific class must relate directly to instances of that class.
- Do not include spaces when entering the attribute and class names. An LDAP display name with embedded spaces can cause problems.
- Object identifiers (OIDs) are issued by International Standards Authorities to prevent duplication. If your organization expects to create new classes and attributes, you might want to first request OIDs from the relevant standards body in your country.

To add a new attribute to the schema, you must create a new attribute object. First, follow the preliminary steps described in “Order of Processing When Extending the Schema” earlier in this chapter. Then do the following:

1. Choose a name for the attribute.
2. Obtain a valid object identifier from an issuing authority.
3. Determine the attribute syntax.
4. Decide whether the attribute needs to be a single-value or multi-value attribute.
5. Decide whether the attribute needs to be indexed or replicated to the Global Catalog.

For every attribute that you define, some attributes are mandatory, and some are optional; these attributes are listed in Table 7.1 and Table 7.2.

**TABLE 7.1** Mandatory Attributes for New Attribute-Definition Objects

<i>Mandatory Attributes</i>	<i>Default Status</i>
cn	No default. Administrator must specify a name.
objectClass	No default. Administrator must specify as attributeSchema.
attributeID	No default. Administrator must specify as an object identifier string.
attributeSyntax	No default. Administrator must specify one of the syntaxes recognized by Active Directory.
oMSyntax	No default. Administrator must specify an oMSyntax that matches the corresponding attribute syntax.
schemaIDGUID	It is defaulted to a value generated by uuidgen if not specified.
nTSecurityDescriptor	Defaults if the administrator does not specify. The default value depends on the defaultSecurityDescriptor attribute of the attributeSchema class.
isSingleValued	Defaults to FALSE if not specified by the administrator.
IDAPDisplayName	Defaults from the common name if not specified by the administrator.

**TABLE 7.2** Optional Attributes for New Attribute-Definition Objects

<i>Optional Attributes</i>	<i>Default Status</i>
rangeLower	No default. The administrator must specify a value.
rangeUpper	No default. The administrator must specify a value.
isMemberOfPartialReplicaSet	Defaults to FALSE if not specified by the administrator.
searchFlags	No default. The four currently defined bits for this attribute are as follows: 1 = Index over attribute only; 2 = Index over container and attribute; 4 = Add this attribute to the Ambiguous Name Resolution (ANR) set (needs to be used in conjunction with 1); 8 = Preserve this attribute on logical deletion (that is, make this attribute available on tombstones).

### Modifying an Attribute

To modify an attribute, modify the existing attribute-definition object that represents the class. Some attributes are designated as system-only, and cannot be modified (even for new classes that you created). System-only attributes have the `systemOnly` attribute set to TRUE.

The following attributes of an attribute-definition object are `systemOnly`, and cannot be modified:

- `attributeID`
- `schemaIDGUID`
- `attributeSyntax`
- `oMSyntax`
- `isSingleValued`
- `extendedCharsAllowed`
- `systemOnly`
- `objectClass`
- `instanceType`

## Adding a Class

To add a new class, you add a new schema-definition object with all the desired attributes. After you remove the Active Directory safety interlocks, make sure that you have done the following before you add a class:

1. Choose a name for the class.
2. Obtain a valid object identifier from an issuing authority.
3. Determine the object class category.
4. Determine the class from which this new class inherits information.

For every class, some attributes are mandatory, and some are optional, as shown in Table 7.3 and Table 7.4. If you do not define values for some of these attributes, they are given default values.

**TABLE 7.3** Mandatory Attributes for New Class-Definition Objects

<i>Attribute</i>	<i>Default Status</i>
<code>cn</code>	No default. Administrator must specify a name.
<code>objectClassCategory</code>	Defaults to 88 class because it is assumed to be a class with no category. The desired options are Structural, Abstract, or Auxiliary.
<code>governsID</code>	No default. Administrator must specify an object identifier string.
<code>possSuperiors</code>	No default. Administrator must specify the structural class or classes that are legal parents of instances of this class.
<code>subClassOf</code>	No default. Administrator must specify a value.

**TABLE 7.3** Continued

<i>Attribute</i>	<i>Default Status</i>
schemaIDGUID	If not specified, a default value is automatically generated by the system.
nTSecurityDescriptor	If not specified, a default value depends on the default SecurityDescriptor of the classSchema class.
IDAPDisplayName	If not specified, defaults from the common name.

**TABLE 7.4** Optional Attributes for New Class-Definition Objects

<i>Optional</i>	<i>Default Status</i>
defaultSecurityDescriptor	If no default security descriptor is specified, the default security descriptor of the immediate superclass is used.
auxiliaryClass	The list of additional (auxiliary) classes from which this class is derived.

For a new class, you must define `cn`, `objectClass`, and `governsID`. However, if you want to make the new class actually useful, you should define some attributes in `mustContain`, `mayContain`, and `possSuperiors`. Any attributes you specify when you add a new class must already exist. You must add the new attributes to the schema before adding a new class with new attributes.

When you add a new class, the object identifier specified in `governsID` must be unique, not only in your enterprise but also globally.

## Modifying a Class

To modify a class, modify the existing class-definition object that represents the class. Just as with attribute modification, some class attributes are designated as system-only, and cannot be modified (systemOnly attribute set to TRUE).

The following attributes of a class-definition object are system-only attributes, and cannot be modified:

- `governsID`
- `schemaIDGUID`
- `rDNAttID`
- `subClassOf`
- `systemMustContain`

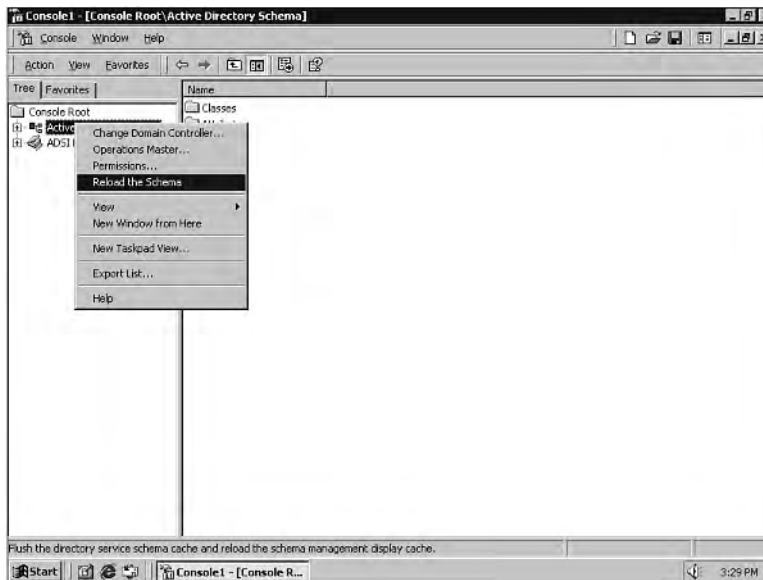


- systemMayContain
- systemPossSuperiors
- systemAuxiliaryClass
- objectClassCategory
- systemOnly
- objectClass
- instanceType

## Verifying Schema Modifications

All changes made to Active Directory are validated first against the version of the schema held in memory. This version is known as the schema cache. Updates to the schema cache are performed automatically after the on-disk version has been updated.

There is also a mechanism for updating the schema cache on demand. You can use this when you modify the schema. You can add the `schemaUpdateNow` attribute to the `rootDSE` with a value of 1. The value is not used, but acts as an operational attribute. Writing this attribute reloads the cache. You can also reload immediately by right-clicking the schema manager root node and selecting **Reload the Schema**, as shown in Figure 7.6.



**FIGURE 7.6**

*Reloading the schema manually.*

The rootDSE is a DSA-specific entry that holds the attributes that pertain to the local domain controller, such as directory partitions, server name, and supported LDAP version numbers. The schemaUpdateNow attribute is defined as an operational attribute, and does not require any storage. Generally, when you set an operational attribute, you trigger some action on the server.

#### NOTE

You should only force an immediate schema cache update once and only after all required schema updates are finished because cache loads have high memory impact.

## Issues with Extending Schema

When you modify the schema, you must be aware of the implications and potential problems that can arise. Three main issues are involved with modifying the schema: replication, concurrency control, and handling invalid object instances. The easiest way to ensure you don't encounter problems with extending the schema is to not make multiple modifications at the same server (such as installing multiple AD-ready applications or scripts). Also, planning and communication are critical in ensuring you do not encounter situations like invalid object instances.

## System Checks for Schema Modifications

Active Directory performs some checks when you try to add or modify a class or attribute, to make sure that the changes do not cause inconsistencies or other problems in the schema. The checks can be divided into two classes: consistency checks and safety checks.

Consistency checks maintain the consistency of the schema. Safety checks reduce the possibility of a schema update by one application breaking another application.

### Consistency Checks

For both class and attribute changes, the system makes sure that the values of LDAPDisplayName and schemaIDGUID are unique and also that LDAPDisplayName is valid.

The class-schema object addition and modification extensions are successful only if the new class definition passes all the necessary tests as well as the normal extension checks.

### Safety Checks

Safety checks reduce the possibility of schema updates by one user or application breaking another application when they share a schema definition.

Schema modifications are subject to certain restrictions enforced by Active Directory.

## Deactivating Schema Objects

You cannot deactivate schema objects that are part of the default schema that ships with Active Directory. You can only deactivate schema objects that have been added to the default schema.

### NOTE

However, you can only deactivate an attribute if it is not associated with an active class. If the class is active, it must be deactivated in order to deactivate the attribute.

You might want to delete schema classes or attributes that are not needed in your organization, but Active Directory does not support the actual deletion of schema objects, only deactivation. When you deactivate a schema object, you make it unusable for most purposes, and you get most of the benefits of deletion.

A deactivated or defunct schema object can be made active again—in the Active Directory Schema console.

To reactivate a class or attribute by using the Active Directory Schema console, perform the following steps:

1. Open the Active Directory Schema console.
2. Double-click the Classes folder or Attributes folder to display the schema classes or attributes.
3. Right-click the class or attribute that you want and then click Properties.
4. Click the Deactivate this Class (Attribute) check box to clear it and then click OK.

To reactivate a class or attribute by using the ADSI Edit console, perform the following steps:

1. Open ADSI Edit.
2. Right-click ADSI Edit and then click Connect To.
3. In the Connection Point box, make sure that Naming Context is selected.
4. In the Naming Context box, select Schema and then click OK.
5. In the console tree, double-click My Connection.
6. Double-click the Schema folder to display a list of attributes and classes in the navigation pane. This might take a few moments.
7. Right-click the class or attribute that you want and then click Properties.
8. In the Select which properties to view box, select Optional and then select isDefunct in the Select a Property to View box.

9. In the Test Attribute Properties dialog box (shown previously in Figure 4.4), type **FALSE**.
10. Click Set and then click OK.

## Deactivating Existing Classes and Attributes

Deactivating schema classes and attributes is subject to the following restrictions:

- You cannot deactivate a category 1 class or attribute.
- You cannot deactivate an attribute that is a member of a class that is not also disabled (as mentioned earlier).

To deactivate an attribute or class, expand the attribute or class container in the Schema console and locate the item to be deactivated. Double-click or right-click and select Properties. In the General tab, select the check the Deactivate this attribute (or object), and click OK.

## Summary

The Active Directory schema is the underlying layout for the Active Directory database and is comprised of objects and attributes. It also controls the structure and content that users can view when browsing Active Directory. The only group that can access the schema is the Schema Admins group. Changing or extending the schema is a significant alteration and should be managed with a change management policy to avoid integrity issues and conflicts.

# Managing Sites, Replication, and Network Traffic

CHAPTER

8

## IN THIS CHAPTER

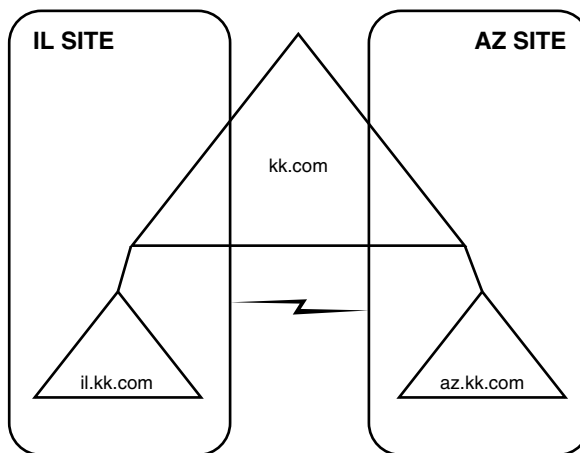
- Site Topology Fundamentals 226
- Active Directory Replication Model 239
- Replication Topology 244

The primary purpose of the Active Directory Sites and Services snap-in is to manage sites and the appropriate replication between them. Active Directory manages network traffic around its concept of sites.

## Site Topology Fundamentals

Sites map the physical structure of your network, whereas domains (if you use more than one) typically map the logical structure of your organization. Logical and physical structures are independent of each other. In Active Directory, there is no necessary correlation between sites (physical) and domain namespaces (logical).

Although physical structure and its domain structure are unrelated, many enterprises may configure their domains to reflect physical network structure. Because domains are partitions, and partitioning influences replication, partitioning the forest into multiple, smaller domains can reduce the amount of replication traffic. Active Directory and its additional network traffic may impact your current bandwidth. Active Directory lets multiple domains appear in a single site and a single domain appear in multiple sites, as shown in Figure 8.1.



**FIGURE 8.1**

*A single domain appearing in multiple sites.*

### CAUTION

You should consider your physical structure when planning your logical structure even though they are not directly related.

## Sites

A site is an area of your network with high bandwidth connectivity and by definition is a collection of well-connected IP subnets (10 Mbps or better). Because sites control how replication occurs, changes made with the Sites and Services snap-in affect the communication efficiency of DCs separated by longer distances within a domain.

A site represents the physical entity of the network, whereas domains represent the logical portion. Theoretically, a site may span multiple domains, and a domain may span multiple sites. Sites control replication of your domain information. Sites also help workstations locate nearby domain controllers (via subnet information) for authentication purposes.

When a site spans multiple domains, there is an increase in the replication within that site because additional domain databases must be replicated in addition to the forest schema, configuration, and GCs.

The Knowledge Consistency Checker (KCC) is built in to Windows 2000 Server and runs on all domain controllers. The KCC automatically establishes connections between domain controllers in the same site. These connections are referred to as Active Directory connection objects. You can add or remove connection objects, but if replication within a site becomes impaired, the KCC establishes new connection objects to re-establish Active Directory replication.

### Default-First-Site

The first site set up automatically when Windows 2000 Server was installed on the first domain controller in your enterprise is called Default-First-Site, as shown in Figure 8.2. If you choose, you can rename this site.



**FIGURE 8.2**

*The Default-First-Site.*

To rename a site, perform the following steps:

1. Open Active Directory Sites and Services.
2. Expand the Sites container, if necessary.
3. Right-click the site you want to rename, click Rename, and then enter the new name.

## Sites and Services MMC Snap-In

You can use the Sites and Services snap-in to display and manage the following:

- Sites
- Servers (domain controllers within the Site container)
- Replication connections and schedules (within the Server/NTDS Settings)
- Inter-Site Transports and links (includes IP(RPC) and SMTP)
- Subnets

## When to Create a New Site

When you have slow links between network segments, it is recommended that you create two sites and place domain controllers into the sites according to the following general rules:

- Deploy at least one Global Catalog (GC) per site.
- Deploy DNS servers on a site level.

The first domain controller in the forest is designated automatically as a GC server. When you create additional sites, you can use Active Directory Sites and Services to select the GC option in the properties for the NTDS Settings object of the server that you want to be the GC.

### NOTE

Placing a Global Catalog server in each site improves search performance because searches do not have to cross site boundaries. In addition, a GC server is required for domain login, so if a connection between sites is not available, logins will fail. However, if a GC server is not available in one site but there is another GC server in a remote site, that server may be used for the logon process. If no GC is available, a cached login will result.

Because the availability of DNS directly affects the availability of Active Directory, you should configure at least one DNS server in every site. Clients rely on DNS to locate domain controllers, and domain controllers rely on DNS to find other domain controllers.

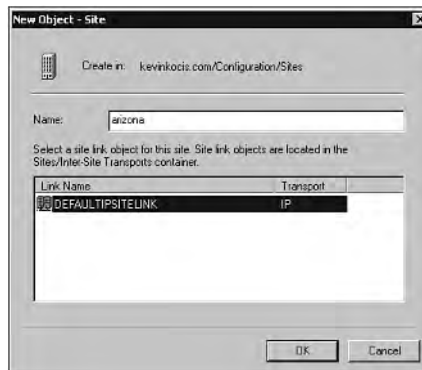


When you create sites and establish domain controllers in the sites, you need to create site links and configure them in accordance with network throughput and replication schedule.

To create a site, perform the following steps:

1. Open Active Directory Sites and Services.
2. Right-click the Sites folder and then click New Site.
3. In the Name box, type the name of the new site.
4. Click a site link object and then click OK.
5. Associate a subnet with a site for this newly created site.
6. Move a domain controller from an existing site into this new site, or install a new domain controller.
7. If you want to choose a specific licensing computer, other than the one automatically selected, select another licensing computer.
8. Delegate control of the site.

When finished, you should see something similar to Figure 8.3.



**FIGURE 8.3**

*Creating a new site.*

In Active Directory Sites and Services, you can delegate control for the Subnets, Inter-site Transports, Sites, and Server containers. Delegating control of an object allows you to specify who has permissions to access or modify that object or its child objects.

To delegate control, perform the following steps:

1. Open Active Directory Sites and Services.
2. Right-click the container whose control you want to delegate and then click Delegate Control to start the Delegation of Control Wizard.
3. Follow the instructions in the Delegation of Control Wizard.

In the case of an external acquisition of a site in your enterprise or a consolidation within, you may need to delete a site. To delete a site, perform the following steps:

1. Open Active Directory Sites and Services.
2. In the console tree, expand the Sites container if necessary.
3. Right-click the specific site container and then click delete.

#### NOTE

A site is a Container object, so deleting a site also deletes all directory objects contained within the site. You cannot delete the site called Default-First-Site.

## Subnets

A site must be associated with one or more subnets. Subnets are derived from the IP address and subnet mask of a computer existing in the particular subnet. You can view your computer's IP configuration by typing **ipconfig /all** at a command prompt.

Computers on TCP/IP networks are assigned to sites based on their location in a subnet or a set of subnets, which group computers to identify their physical network proximity. Subnet information is used during the authentication process to locate a domain controller in the same site as the computer's subnet. Replication is also dependent upon subnet information to determine the best routes between domain controllers.

If your network consists of a single local area network (LAN) or a set of LANs connected by a high-speed backbone, the entire network can be a single site. The first domain controller you install automatically creates the first site, known as the Default-First-Site-Name. Additional domain controllers are automatically added to the same site as the original domain controller, and can be moved when other sites are created later. The only exception is if, when you install a domain controller, its IP address falls within the subnet previously specified in an alternative site, the domain controller is then added to this alternative site.

To create a subnet, perform the following steps:

1. Open Active Directory Sites and Services.
2. In the console tree, expand the Sites container if necessary.
3. Right-click Subnets and then click New Subnet.
4. In the Address box, enter the subnet address. Any IP address within the subnet range is a valid entry.
5. In the Mask box, enter the subnet mask that describes the range of addresses included in this site's subnet.
6. Choose a site with which to associate this subnet and then click OK.

To associate a subnet with a site, perform the following steps:

1. Open Active Directory Sites and Services.
2. In the console tree, expand the subnet container with which you want to associate the site and then click Properties.
3. In the Site box, click a site with which to associate this subnet.

To delete a subnet, perform the following steps:

1. Open Active Directory Sites and Services.
2. In the console tree, expand the subnet container and select the subnet that you want to delete and then click Delete.

## Connections

A connection object represents a replication connection from one domain controller to another. The connection object is a child of the replication destination's NTDS Settings object and points from the source to the replication source.

Connection objects are created in two ways—either automatically by the KCC or manually by the administrator.

A connection is unidirectional. A bidirectional replication connection is represented as two connection objects under two different NTDS Settings objects.

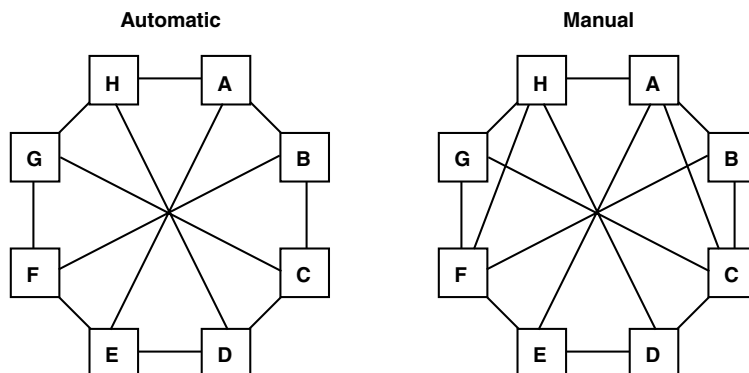
By creating site links and configuring their replication availability, relative cost, and replication frequency, you provide the KCC with information about what connection objects to create to facilitate replication of directory data. Active Directory uses site links as indicators for where it should create connection objects, and connection objects use the actual network connections to exchange directory information.

Replication is performed between naming context (NC) replicas. Domain controllers will often have several NCs in common (always have at least two—the Configuration NC and the Schema NC). The connection between domain controllers will be used to replicate as many NCs as needed.

### NOTE

You don't need to create multiple connections linking the same two domain controllers in the same direction.

The KCC automatically creates connections to maintain directory connectivity during failures. See Figure 8.4 for an illustration of how the KCC performs this function.



**FIGURE 8.4**

*Automatic (KCC) and manual (administrator) connections.*

The only situation where you would manually configure connections is when the KCC's connection fails to connect certain domain controllers that you believe should be connected.

For example, within a site, you can add connections to reduce intrasite replication latency. By default, an update takes at most three hops from where it originates in a site to any other domain controller in a site. Even though the extra connections could reduce the hop count to two or one, the consequences to this action are extra CPU cycles and disk reads due to replication.

You may also want to establish a replication schedule that the KCC couldn't create.

**NOTE**

If a manual connection matches one the KCC would normally create, the KCC will not create an additional connection. The KCC will never delete a connection created manually.

You can specify the replication period when you create a site link object. If this value is not created, a global default replication period (which can be modified) will be assumed for the site link. When the KCC creates a connection object, its replication period will be the maximum of the periods along the minimum-cost path of site link objects from one end of the connection to the other.

## Site Links

A site link object represents a group of sites that can communicate at uniform cost through an intersite transport. Site link objects are unidirectional connections between two or more sites, and will typically correspond to an actual WAN link.

Network connections between sites are represented by site links. A site link is a low-bandwidth or unreliable connection between sites. You should consider any two networks connected by a link that is slower than LAN speed to be connected by a site link. Also, a high-speed link that is near capacity has a low effective bandwidth and should also be considered a site link. When you have multiple sites, sites connected by site links become part of the replication topology.

You create a site link object for a specific intersite transport by specifying:

- A numeric cost—Higher cost numbers represent more expensive messages. And costs influence the frequency of replication on KCC-configured connections.
- Two or more sites—In order to configure a site link, you must have at least two sites configured in your enterprise.
- A schedule—The schedule determines the time periods when the link is available.

A site can be connected to other sites by any number of site link objects. Each site in a multi-site directory must be connected by at least one site link in order to replicate with domain controllers in other sites. Site links must be configured if two or more sites exist in your enterprise.

To create a site link, perform the following steps:

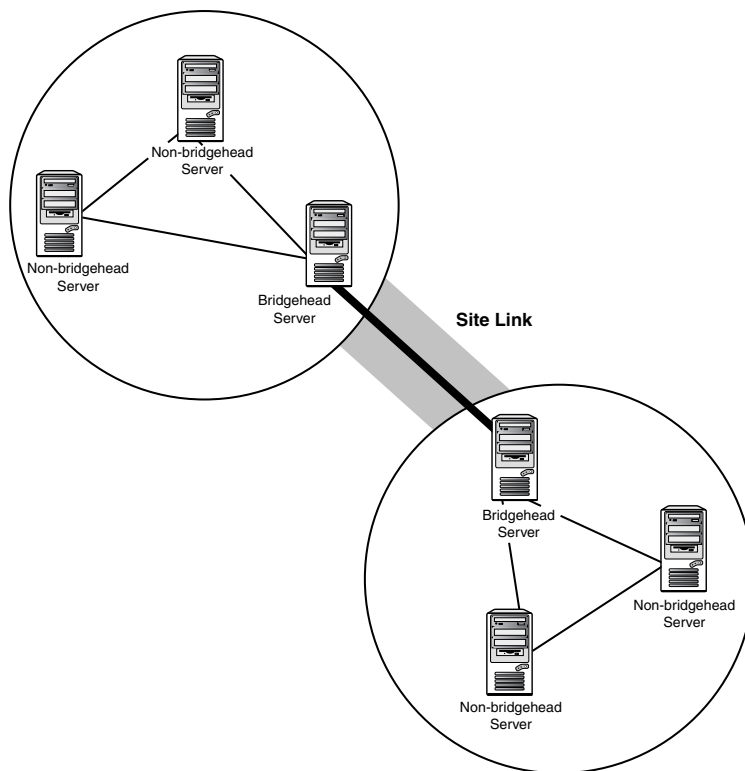
1. Open Active Directory Sites and Services.
2. In the console tree, expand the site and intersite transport containers hosting the site link.
3. Right-click the desired intersite transport protocol and then click New Site Link.

4. In the Name box, type the name to be given to the link.
5. Click two or more sites to connect and then click Add.
6. Configure the site link's cost, schedule, and replication frequency (see the next section on Site Link Attributes), and click OK.

Creating a site link influences replication topology. Active Directory uses the link's cost, schedule, and replication data to establish connection and schedules for optimal replication.

Site links are transitive by default, which means that a domain controller in one site can make replication connections with domain controllers in any other site. For example, if site A is connected to site B, and site B is connected to site C, then domain controllers in site A can communicate with domain controllers in site C.

Figure 8.5 shows two sites connected by a site link. Of the six domain controllers in the figure, two are bridgehead servers (the bridgehead server role is assigned automatically by the system).



**FIGURE 8.5**

*Two sites connected by a site link. Each site's preferred bridgehead server is used preferentially for intersite information exchange.*

The bridgehead servers are the preferred servers for replication, but you can also configure the other domain controllers in the site to replicate directory changes between sites.

In situations where an acquisition, merger, or division requires deletion of a site, you can perform the following steps:

1. Open Active Directory Sites and Services.
2. In the console tree, expand the site and intersite transport folder containers hosting the site link.
3. In the details pane, right-click the site link you want to delete and then click Delete.

## Site Link Attributes

You should provide availability, cost, and frequency information for all site links as part of the process of providing Active Directory with information about available intersite connections.

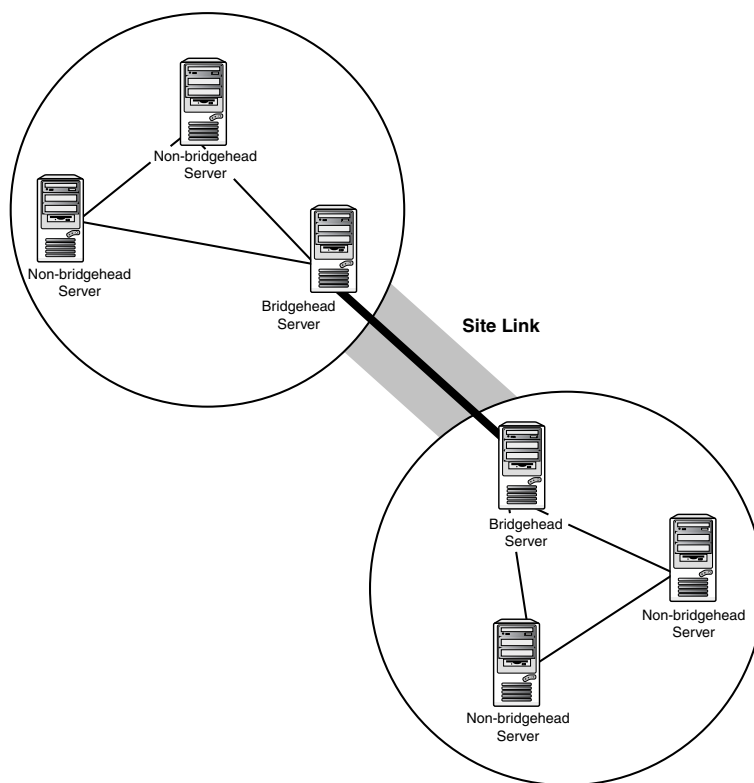
- Site availability—Configure the sites available for replication.
- Cost—Configure site link cost (similar to a routing cost) to assign a value for each available connection used for intersite replication. The lower the cost, the more preferred the link for Active Directory use. The higher the cost, the less desirable the link.
- Replication frequency—Configure site link replication frequency in terms of minutes to inform Active Directory how long it should wait before using this connection to check for replication updates. The replication interval must be at least 15 and no more than 10,080 minutes (one week).

## Bridgehead Servers

Bridgehead servers are dedicated domain controllers that manage intersite replication for the site, as shown in Figure 8.6. You can specify a preferred bridgehead server if you have a computer with appropriate bandwidth to transmit and receive information. If there's typically a high level of directory information exchange, a computer with more bandwidth can ensure that these exchanges are handled promptly. Typically, the same bridgehead server is used for all replication under a particular transport.

### NOTE

You can specify multiple preferred bridgehead servers, but only one will be the active preferred bridgehead server at any time.



**FIGURE 8.6**

*A site link and its corresponding bridgehead servers.*

If the active preferred bridgehead server fails, Active Directory selects another preferred bridgehead server to be the active preferred bridgehead server from the set you designate. If no active preferred bridgehead server is available and there are no other preferred bridgehead servers available for Active Directory to select, it selects another domain controller in the site to be the preferred bridgehead server. If no other domain controller is available, intersite replication will fail.

You must specify a preferred bridgehead server if your deployment uses a Windows 2000 Server firewall. Establish your firewall proxy server as the preferred bridgehead server, making it the contact point for exchanging information with servers outside the firewall.

If there are multiple servers in a site, the preferred bridgehead server for a protocol used in a site link will solely support intersite replication. Although domain controllers will still exchange directory information as needed, the preferred bridgehead server will be the primary choice for intersite replication.



The bridgehead server then distributes the directory information via intrasite replication.

To designate a preferred bridgehead server, perform the following steps:

1. Open Active Directory Sites and Services.
2. In the console tree, expand the Sites container.
3. Expand the domain containing the domain controller to be designated as the bridgehead server.
4. Expand the Servers container, if necessary.
5. Right-click the domain controller, and select Properties.
6. Select a transport (IP or SMTP), click the Add button, and then click OK.

### CAUTION

Establishing manual bridgehead servers may impact the KCC's ability to fail over to another bridgehead in the event of an outage.

## Site Link Bridges

A site link bridge object represents a set of site links, all of which communicate via an assigned transport. Site link bridges are user-specified collections of fully routable site links, which add transitive features to replication. Like bridges within a router, they connect multiple site links so that replication can pass along many sight links.

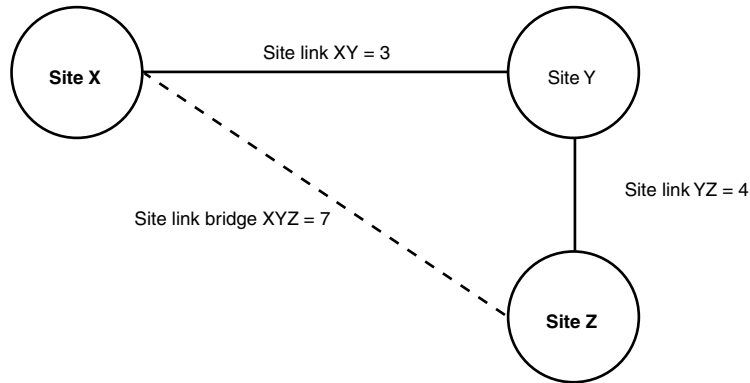
You create a site link bridge object for a specific intersite transport by specifying two or more site links for the specified intersite transport.

To understand what a site link bridge means, consider this example (as illustrated in Figure 8.7):

- Site link XY connects sites X and Y through IP with cost 3.
- Site link YZ connects sites Y and Z through IP with cost 4.
- Site link bridge XYZ connects XY and YZ.

The site link bridge XYZ implies that an IP message can be sent from site X to site Z with cost  $3+4 = 7$ . That is all the bridge does in this simple example.

Each site link “L” in a bridge should have some site in common with another site link in the bridge. Otherwise, the bridge cannot compute the cost from sites in link “L” to the sites in other links of the bridge.

**FIGURE 8.7**

*A site link bridge.*

Multiple site link bridges for the same transport work together to model multi-hop routing. Add the following objects to the previous example:

- Site link WX connects sites W and X through IP with cost 3.
- Site link bridge WXY connects WX and XY.

Now the site link bridges WXY and XYZ together imply that an IP message can be sent from site W to site Z with cost  $2+3+4=9$ .

If your IP network is not fully routed, you can turn off the transitive site link feature for the IP transport, in which case all IP site links will be considered intransitive, and you can configure site link bridges to model the actual routing behavior of your network.

To turn off the transitive site link for the IP transport, perform the following steps:

1. Open Active Directory Sites and Services.
2. In the console tree, expand the Sites and Inter-Site Transports containers, if necessary.
3. Right-click the IP container, and select Properties.
4. Clear the Bridge All Site Links check box.

After you have removed the transitive site link feature, you need to establish manual site link bridges.

Any network that you can describe by a combination of site links and site link bridges, you can also describe by site links alone. By using site link bridges, the network description is much smaller and easier for you to maintain because you don't need a site link to describe every possible path between pairs of sites.

To create a site link bridge, perform the following steps:

1. Open Active Directory Sites and Services.
2. In the console tree, open the Site and Intersite Transport containers containing the site link and then click New Site Link Bridge.
3. In the Name field, type a name for the site link bridge.
4. Click two or more site links to be bridged, click Add, and then OK.

**NOTE**

If you have enabled Bridge All Site Links, this procedure is redundant and will have no effect.

To delete a site link bridge, perform the following steps:

1. Open Active Directory Sites and Services.
2. In the console tree, open the Site and InterSite Transport containers hosting the site link and then click New Site Link Bridge.
3. In the details window, right-click the site link bridge you want to delete and click Delete.

**NOTE**

In a fully routed IP network, there is no need to create site link bridges.

## Active Directory Replication Model

The Active Directory replication model determines how changes are propagated and tracked among domain controllers. As mentioned in Chapter 2, “Active Directory Architecture,” each domain controller in a forest stores a replica of the directory partitions locally. Updates to replicas are synchronized among the domain controllers that store the same directory partitions during the process of replication.

The replication topology of sites on your network controls where and when replication occurs.

The first domain controller created in active directory will always migrate to the default-first-site container. (This is because you will have not yet gotten the chance to create any sites, subnets, and site-links.) This will continue to be the case for all additional domain controllers that you create unless you define the physical structure of your network. Once the physical structure has been defined in AD, the promotion of additional domain controllers will result in the new domain controllers migrating to their respective sites based on their IP address and the subnet to site associations defined in AD.

## Directory Partition Replicas

A directory partition replica can be a full (master) replica or a partial replica.

A full replica contains all attributes of all directory partition objects and is both readable and writable. Each domain controller stores at least three full, writable directory partition replicas as follows:

- The schema partition, which contains all class and attribute definitions for the forest. There is only one schema directory partition per forest.
- The configuration partition, which contains replication configuration information for the forest. There is only one configuration directory partition per forest.
- The domain partition, which contains all objects stored by one domain. There is one domain directory partition for each domain in the forest.

A full replica of a domain's partition is stored on each domain controller located in that domain (regardless of which site it is located in). A full replica of the configuration and schema partitions is stored on all domain controllers in the forest.

A partial replica (read only) is stored only on Global Catalog servers, and contains a subset of the attributes of all directory partition objects. Partial replicas form the Global Catalog and increase search efficiency throughout the forest.

Therefore, on a specific domain controller, a single database stores copies of those objects that are pertinent to only that domain, in addition to copies of the schema and the configuration objects, which apply to all domains in the forest.

## Benefits

According to Microsoft, some of the key benefits of the Active Directory replication model are as follows:

- Active Directory always replicates changes to the correct object and can differentiate between a deleted object and a new object that has the same distinguished name (also known as "DN"). This is possible because the process of replication is based on the globally unique identifiers (GUIDs) of directory objects, not on their distinguished names.
- Since only attribute changes are replicated (as opposed to the entire object), this replication model minimizes update conflicts and traffic.
- Wide area network (WAN) communication is minimized through support for store-and-forward replication and the compression of replication data between sites. Servers contain only a subset of the objects in the entire directory—those required for the forest and those specific to the server domain.

- Replication topology (including choice of transports) is flexible to make the best use of different network topologies.
- System configuration remains flexible because the sites are not tied to the partition structure of the directory.
- Speed over high-latency communication links is enhanced due to lowering the number of network round-trips by replication protocols.
- Minimized dependencies on other services, such as time synchronization (W32Time).

## Replication Components

The following mechanisms contribute to the overall replication system:

- Multi-master loose consistency with convergence, which maintains data integrity.
  - *Multi-master* means that a directory partition can have many writable replicas, or copies, that must maintain consistency between domain controllers. Changes or updates are then propagated to other domain controllers in the forest.
  - *Loose consistency* means that the replicas are not guaranteed to be consistent with each other at any particular point in time because changes can be applied to any full replica at any time.
  - *Convergence* means that if the system reaches a static state (after all updates have been completely replicated), all replicas are guaranteed to converge on the same set of values.
- Store-and-forward replication, which means that changes are distributed to only a subset of domain controllers. This subset of domain controllers then forwards the updates to other domain controllers, and so on, until the change has propagated to every domain controller.
- Pull replication, which means that domain controllers request (or pull) updates from replication partners.
- State-based replication, which means that instead of storing a full change log, each directory partition replica stores per-object and per-attribute data to support replication.

## Multi-Master Replication

Active Directory domain controllers support multi-master replication, synchronizing data on each domain controller, and ensuring consistency of information over time. Multi-master replication replicates Active Directory information among peer domain controllers, each of which has a read-and-write copy of the directory. This is a change from the Windows NT Server operating system, where only the PDC had a writable copy of the directory (the BDCs received read-only copies from the PDC). Once configured, replication is automatic and transparent.

## Store-and-Forward Replication

Store-and-forward replication is designed to reduce communication over slow WAN links. An update replicates first to nearby replicas, then expands to replicas that are farther away. Store-and-forward greatly reduces the WAN traffic produced by replication.

Active Directory can create the topology automatically after you have defined sites, site links, and site link bridges. Based on this model, Active Directory creates replication connections that allow Active Directory to perform replication. When failures occur, Active Directory modifies replication connections to keep replication going. Manually created connections coexist with automatically generated ones.

## Pull Replication

Active Directory uses pull replication, in which a destination replica requests information from a source replica. The request specifies the information that the destination needs, based on its knowledge of changes already received from the source and from all other domain controllers in the domain. When the destination receives information from the source, it applies that information, bringing itself more up-to-date. The destination's next request to the source excludes the information that has already been received and applied.

The alternative is push replication, in which a source sends information to a destination unsolicited, in an attempt to bring the destination more up-to-date. Push replication is challenging because it is difficult for the source to know what information the destination needs, especially if the destination has received identical information from another source.

## State-Based Replication

In state-based replication, each domain controller applies updates to its replica as they arrive, without maintaining a change log file. In a typical log-based replication system, each master keeps a log of the updates that it originated which it shares with other masters to ensure consistency.

Active Directory replication is driven not by logs stored with the source replica, but by the current "state" (the current values of all objects) of the source replica. This state includes information used to resolve conflicts and avoid sending the full replica during each replication. Each originating write operation is assigned a unique sequence number (USN) at its originating domain controller. The USN is a 64-bit number designed to manage replication consistency.

All replicas maintain information about how up-to-date they are with respect to all other replicas, and values in the directory are tagged with USNs of their originating write updates. By using this information, the replication source can filter the state changes that it replicates.

A state-based approach uses a single mechanism for incremental and full synchronization, and performs fewer database updates.

## Updates

Replication is triggered when an object is updated a domain controller. A timer is started and changes are collected for a predetermined period, after which the replication engine notifies adjacent domain controllers in the replication topology. After it has been notified that there are changes to be collected, the destination domain controller contacts the source domain controller to request the changes.

Replication between sites is typically performed on a scheduled basis. A domain controller requests changes from domain controllers in other sites according to a configurable schedule.

## Update Propagation and Update Sequence Numbers

Some directory services use time stamps to detect and propagate changes. Since time synchronization in a network is difficult, updates can be lost.

Active Directory replication is not time-dependent. Instead, it uses Update Sequence Numbers (USNs), which are 64-bit numbers maintained by each Active Directory domain controller to track updates. When the server writes to any attribute, or property, on an Active Directory object (including the originating write or a replicated write), the USN is advanced and stored with the updated property and with a property specific to the domain controller. The incrementing and storage of the USN and the write of the property value succeed or fail as a single unit.

Each Active Directory-based server maintains a table of USNs received from replication partners where only the highest USN is stored in this table. When a given partner notifies the directory server that replication is required, that server requests all changes with USNs greater than the last value received.

USNs allow for easier recovery in the event of a failure. To restart replication, a server requests all changes with USNs greater than the last valid entry in the table.

## Originating

A Lightweight Directory Access Protocol (LDAP) directory server supports the following four types of update requests:

- Add a directory object
- Modify (add, delete, or replace) object attribute values
- Move an object by changing the name or parent of the object
- Delete a directory object

An LDAP directory server processes each write request as an all-inclusive individual transaction. A write request either completely commits or fails before completion and has no effect on the directory.

A write request that commits is called an originating update.

If the update originated on one domain controller and is replicated to a second domain controller, the update to the second domain controller is called a replicated update and is distinguished by the replication system from an originating update.

## Replication Topology

The replication topology in Windows 2000 Active Directory is complex and could constitute a chapter within itself. A variety of components and factors create a positive replication topology. Let's begin with the protocols.

## Transports and Protocols

Directory information can be exchanged using the following network protocols:

- IP replication—IP replication uses remote procedure calls (RPC) for both intrasite and intersite replication. By default, intersite IP replication adheres to replication schedules, and does not require a certification authority (CA).
- SMTP replication—SMTP replication is used only for intersite replication. You cannot use SMTP replication to replicate between domain controllers in the same domain. SMTP replication can be used only for schema, configuration, and Global Catalog partial replica replication. SMTP replication observes the automatically generated replication schedule.

If you choose to use SMTP over site links, you must install and configure an enterprise certification authority (CA). The domain controllers obtain certificates from the CA, which the domain controllers then use to sign and encrypt the mail messages that contain directory replication information, ensuring the authenticity of directory updates. SMTP replication uses 56-bit encryption.

## Comparison of RPC to SMTP

Replication transports provide the wire protocols required for data transfer. Windows 2000 provides three levels of connectivity for replication of Active Directory information:

- Uniform high-speed, synchronous RPC over IP within a site
- Point-to-point, synchronous, low-speed RPC over IP between sites
- Low-speed, asynchronous SMTP between sites

The following rules apply to the replication transports:

- Replication within a site always uses RPC over IP.
- Replication between sites can use either RPC over IP or SMTP over IP.



- Replication between sites over SMTP is supported for only domain controllers of different domains. Domain controllers of the same domain must replicate by using the RPC over IP transport.

The Inter-Site Transports container allows you to map site links to the transport used by the link. When you create a site link object, you create it in either the IP container or the SMTP container.

The following characteristics apply to both SMTP and RPC with respect to Active Directory replication:

- For replication between sites, data replicated through both transports is compressed.
- Active Directory can respond with only a fixed (maximum) number of changes per change request, on the basis of the size of the replication packet. The size of the replication packet is configurable.
- Active Directory can have only a single change request outstanding for a specific directory partition to a specific replication partner.
- Changes are transported in one or many frames, based on the total number of changed or new values.
- TCP transports the data portion by using the same algorithm for both SMTP and RPC.
- If transmission of the data portion fails for either, complete retransmission is necessary.
- If bandwidth is limited, the same TCP retransmission characteristics apply. (RPC time-out is much longer than TCP time-out.)

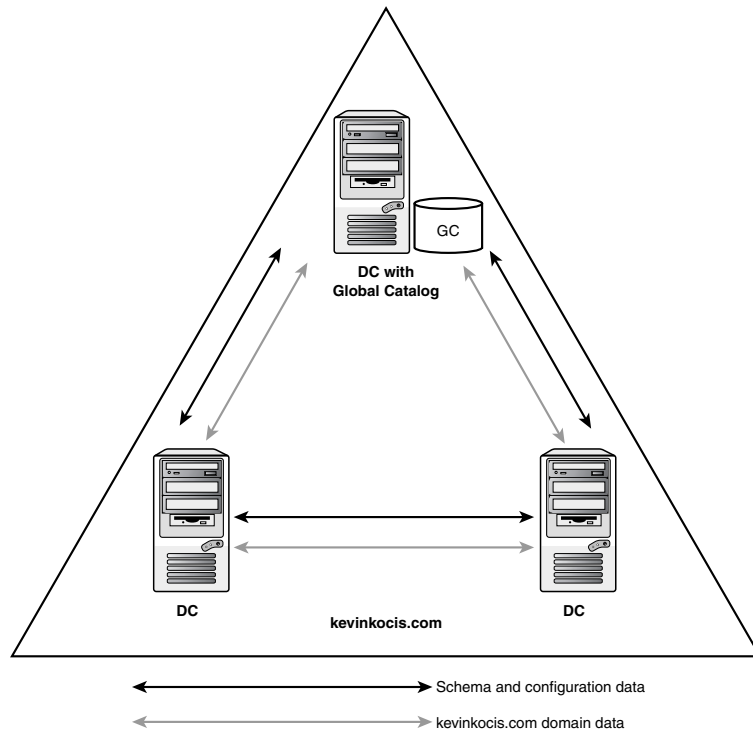
Because SMTP is not used for replication of domain directory partitions, Windows 2000 provides point-to-point synchronous RPC replication in addition to asynchronous SMTP replication between sites to allow the flexibility of having domains span multiple sites. RPC is best used between well-connected sites because it involves lower latency. SMTP is best used between sites where RPC over IP is not possible.

Active Directory replication uses both transports to implement a request-response mechanism. Active Directory issues requests for changes and replies to requests for changes. RPC maps these requests into RPC requests and RPC replies. SMTP, on the other hand, actually uses long-lived TCP connections to deliver streams of mail in each direction. Thus, RPC transport expects a response to any request more or less immediately and can have a maximum of one active inbound RPC connection to a directory partition replica at a time. The SMTP transport expects much longer delays between a request and a response. As a result, multiple inbound SMTP connections to a directory partition replica can be active at the same time, provided the requests are all for a different source domain controller or directory partition.

## Intrasite Replication

Directory information within a site is replicated frequently and automatically. Intrasite replication minimizes replication latency, to maintain data consistency. Intrasite directory updates are not compressed, which uses more network resources but requires less processing power.

Figure 8.8 illustrates replication within a site. Three domain controllers (one of which is a Global Catalog) replicate the forest's schema data and configuration data, as well as all directory objects (with a complete set of each object's attributes).



**FIGURE 8.8**

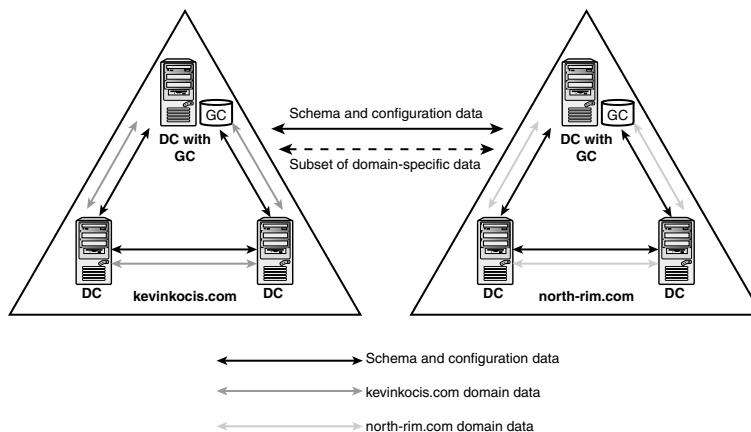
*Intrasite replication (replication within a site).*

The replication topology is automatically generated by the KCC, which attempts to establish a topology that allows at least two connections to every domain controller, so if a domain controller becomes unavailable, directory information can still reach all online domain controllers through the other connection.

The KCC continuously evaluates and modifies the replication topology to meet the changing state of the network. For example, when a domain controller is added to a site, the replication topology is modified to include the new server in the replication process.

If you expand your deployment from the first domain controller in one domain to multiple domain controllers in multiple domains, but in the same site, the replicated directory information changes to include the replication of the partial replica between Global Catalogs in different domains.

Figure 8.9 shows two domains, each containing three domain controllers. One domain controller in each domain is also a Global Catalog server. Within each domain, the domain controllers replicate the forest's schema data and configuration data, as well as all directory objects (with a complete set of each object's attributes), just as in Figure 8.8. In addition, each Global Catalog replicates the directory objects (with only a subset of their attributes) for its own domain to the other Global Catalog.



**FIGURE 8.9**

*Intrasite replication with two domains and two Global Catalogs.*

## Intersite Replication

Create multiple sites to optimize both server-to-server and client-to-server traffic over WAN links. In Active Directory, intersite replication automatically minimizes bandwidth consumption between sites.

Microsoft recommends the following practices when setting up multiple sites:

- **Geography**—Establish every geographic area requiring fast access to the updated directory information as a site.

- Domain controllers and Global Catalogs—Place at least one domain controller in every site and configure at least one domain controller in each site as a Global Catalog. This will alleviate dependency on other sites for directory information.
- Site Links—Keep all site links transitive and ignore replication schedules.

## Replication Tombstones

As mentioned earlier in this chapter, Windows 2000 uses a multiple master replication scheme. This scheme makes it somewhat difficult to remove data from the active directory database or to restore pieces of the active directory database and maintain data consistency at the same time. Windows 2000 uses a data removal method referred to as *tombstoning* to accomplish this task.

When a user deletes an object from Active Directory, the object is not actually deleted from the active directory database at that moment. Instead the object is removed from the user interface, and is marked in the database with a tombstone. The tombstone is an expiration attribute for the object. This tombstone attribute is then replicated to all other DC's. When the expiration time set in the tombstone is reached, the object will be removed from all copies of the active directory database at approximately the same time.

## Replication Tools

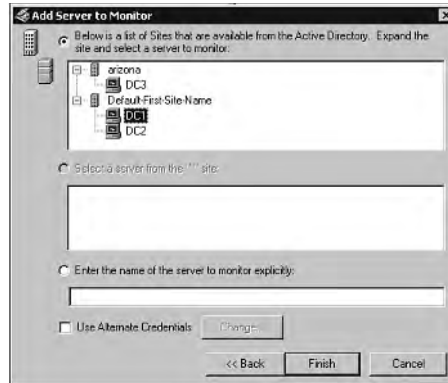
There are a few tools and utilities that Microsoft provides to manage and monitor Active Directory replication. They are as follows:

- Replication Monitor (replmon.exe)
- Replication Administrator (repadmin.exe)
- dsastat

### Replication Monitor (REPLMON)

Replication Monitor is part of the support tools included with Windows 2000 Server. Its GUI uses icons to display server roles such as GCs and their status. You can specify naming contexts and sites, and track the replication traffic by creating log files for each DC. The log files hold statistics for each replication partner and topology.

Use the replication monitor after you've set up your site structure to verify its functionality. This GUI tool (as shown in Figure 8.10) will also help you determine whether the current topology is the best for your network by displaying a graphical layout of all connection objects between domain controllers. Obviously, this tool also serves well as a troubleshooting agent for issues related to replication and bandwidth because it saves network statistics in its log files.



**FIGURE 8.10**  
*The replmon.exe tool.*

## Replication Administrator (REPADMIN)

The Replication Administrator utility is also part of the support tools included with Windows 2000 Server. Although it provides functionality similar to replmon.exe, it operates as a command-line tool (instead of a GUI) and provides accurate statistics.

## dsastat

The dsastat utility is not geared toward sites or replication but assists with diagnosing naming context issues. You would use dsastat if there were no obvious issues in the replication monitor log files. dsastat compares different naming contexts on different DCs and can detect replica inconsistencies.

## Summary

This chapter addressed the purpose of sites and subnets, and how they relate to Active Directory replication. Sites constitute the physical aspect of AD, whereas the namespace and domains make up the logical. It is important to remember that although they are not related, they need to be built in tandem. The next chapter looks at managing flexible single master operations servers.



# Managing Updates with Flexible Single-Master Operations

CHAPTER

9

## IN THIS CHAPTER

- Flexible Single-Master Fundamentals 252
- Operations Master Roles and Placement 253
- Controlling Role Transfers 271
- Controlling Role Seizures 272
- Operations Masters Troubleshooting 273

Although Active Directory allows all domain controllers (DCs) to act as peers in an enterprise, and multi-master replication allows for consistency, you'll still find some exceptions to this peer group. Five roles must retain the responsibility of a single domain controller. These roles are called Flexible Single-Master Operations (FSMO). This chapter examines the importance of each role and how you can place these roles in your enterprise to ease administration.

## Flexible Single-Master Fundamentals

In Active Directory, Flexible Single-Master Operations (FSMOs, pronounced fizz-moes) prevent conflicting updates in domains and forests. It is important to understand the functions of the various single-master operations as well as their placement in your organization. Planning the management of FSMOs will help you maintain a productive Active Directory deployment.

### NOTE

In most cases you will see this technology referred to as flexible single-master operations, or FSMOs, but it is worth noting that in some Microsoft curricula this technology is simply referred to as Operations Masters. Either term is referring to the same thing.

As mentioned in Chapter 8, "Managing Sites, Replication, and Network Traffic," Active Directory uses multi-master updates to maintain updates among domain controllers. This capability is useful if a domain controller becomes disconnected from the network; the DC can be updated when network connectivity is restored. Multi-master updates are replicated throughout the forest.

For example, if two administrators make simultaneous or conflicting updates, Active Directory allows both updates to be replicated. When this situation occurs, the domain controllers eventually resolve their conflicting updates through conflict resolution. In most cases, the last domain controller to write the update wins. However, it is better to prevent conflicts than to attempt to resolve them. In Active Directory, loss of critical data may result if conflicts are resolved in this matter. In this case, you will see how FSMOs effectively prevent AD-critical update conflicts.

## FSMO and Directory Schema Updates

FSMOs are crucial to Active Directory administration because they prevent update conflicts. One special situation in which you need to prevent conflicts is with directory schema updates. Because the schema is updated very infrequently (or should be) and is critical to the functionality of Active Directory, you would not want to use the standard conflict resolution in the event of simultaneous updates.



**NOTE**

As mentioned in Chapter 7, “Managing and Modifying Active Directory Schema,” you should not need to edit or modify the schema regularly. This process should follow an approved and documented procedure requiring sign-off from your implementation team and senior management. Refer to the aforementioned section for more details.

For this reason, Active Directory performs certain updates in a single-master fashion to prevent conflicts. With schema updates, only the designated domain controller in the forest (the one holding the schema master role) accepts updates to schema objects. The schema master role is one example of a Flexible Single-Master Operation role, also referred to as an operations master role or FSMO role. For each role that controls a specific set of directory changes, only the domain controller holding that role can make the necessary directory changes.

**NOTE**

FSMOs perform roles similar to those of primary domain controllers in an NT4 environment. They perform specific functions required by a single domain controller in the forest or domain.

**NOTE**

FSMOs are not a consideration in small Active Directory deployments with a single domain controller.

If you have deployed a complex Active Directory structure, you need to consider which domain controllers will hold operations master roles, where they will be located or relocated, and what to do in the event of a server unavailability or failure.

This chapter discusses all the operations master roles, their placements, security, and troubleshooting.

## Operations Master Roles and Placement

Active Directory identifies five operations master roles:

- Schema master
- Domain naming master

- Relative identifier (RID) master
- Primary domain controller emulator
- Infrastructure master

The schema master and domain naming master are per-forest roles, meaning that only one schema master and only one domain naming master exist in the entire forest. The other operations master roles are per-domain roles, meaning that each domain in a forest has its own RID master, primary domain controller emulator, and infrastructure master. Table 9.1 demonstrates this.

**TABLE 9.1** Roles Per Forest and Domain

<i>Per Forest</i>	<i>Per Domain</i>
Schema master	RID master
Domain naming master	Primary domain controller emulator
	Infrastructure master

Consequently, a forest with only one domain has five operations master roles. A forest with more than one domain has more than five roles because the per-domain roles need to exist in each domain. The basic formula for determining the number of FSMO roles in your enterprise is as follows:

$$2+(3-\text{number of domains})$$

So, if your enterprise consists of eight domains  $[2+(3-8)]$ , you will have 26 FSMO roles. The 2 consistent roles are the schema master and domain naming master (per-forest roles). Make sure these roles are maintained on a single DC in the forest. The other 24 roles are spread out 3 per domain (eight domains). You'll learn more about the placement of these roles later in the chapter.

#### NOTE

These roles can be hosted only on Windows 2000 domain controllers, regardless of the domain mode (mixed or native). By default, all five roles are installed on the first domain controller installed in Active Directory.

Let's consider the example of Kevinkocis.com. If you set up domains in North America and South America, you could use the following domain naming structure:

- Kevinkocis.com (root domain)
- Na.kevinkocis.com (North American domain)
- Sa.kevinkocis.com (South American domain)

Remember that the schema master and domain naming master are forest roles, so only one of each exists in the entire enterprise:

- Schema master (forest): Kevinkocis.com
- Domain naming master (forest): Kevinkocis.com

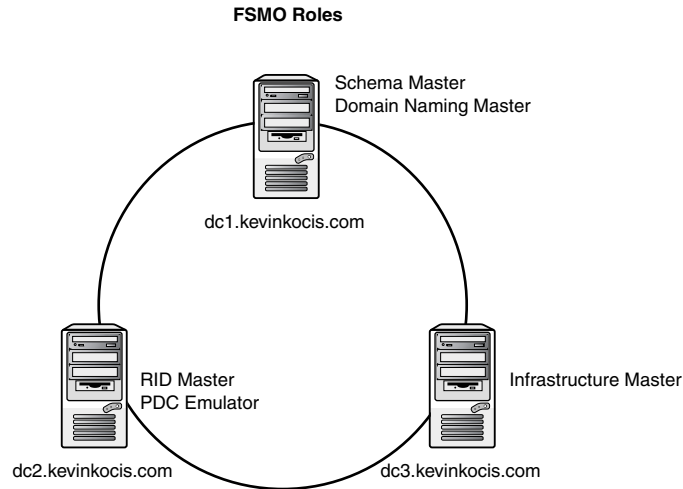
However, because three domains require one of each of the three per-domain roles, an additional nine roles are added to the enterprise:

- RID master (domain): Kevinkocis.com
- RID master (domain): Na.Kevinkocis.com
- RID master (domain): Sa.Kevinkocis.com
- Primary domain controller emulator (domain): Kevinkocis.com
- Primary domain controller emulator (domain): Na.Kevinkocis.com
- Primary domain controller emulator (domain): Sa.Kevinkocis.com
- Infrastructure master (domain): Kevinkocis.com
- Infrastructure master (domain): Na.Kevinkocis.com
- Infrastructure master (domain): Sa.Kevinkocis.com

#### NOTE

FSMO roles are not numerically equivalent to domain controllers. Multiple roles may be placed on the same DC. Some recommendations and restrictions are mentioned in the "Operations Master Roles and Placement" section of this chapter.

Any domain controller located in any domain in the forest can hold the schema or domain naming role. Per-domain roles must be hosted on domain controllers in their respective domains. Because a single domain controller can host up to 5 operations master roles, including 1 of each role, the 11 roles might be hosted by as few as 3 domain controllers or as many as 11. Let's look at Figure 9.1.

**FIGURE 9.1**

*A hypothetical perspective of how the roles can be distributed across servers in a domain that is also the forest root.*

In this example, three domain controllers host all the necessary roles for the enterprise. Remember the first controller to be installed in a Windows 2000 forest has all five roles assigned to it by Active Directory. The first domain controller installed in a new domain in the existing forest has all three per-domain roles assigned to it. However, these roles can be transferred, depending on several factors. I'll go into more detail about placement and transferring roles later in the chapter.

For now, let's look at the responsibilities of each of these roles in greater detail.

## Schema Master

The schema master role is responsible for writing updates to the directory schema, including the creation of new classes or attributes. Because only one schema exists in the forest, only one domain controller in the forest can be assigned this role. Schema updates are then replicated from the schema master to all other domain controllers in the forest.

Updating the directory schema requires you to connect to the domain controller holding the forest's schema master role. You can transfer roles between other domain controllers if necessary.

To determine the schema master role in a forest, open the Active Directory Schema console and right-click Active Directory Schema in the top-left pane. Then select Operations Masters to view the server holding the schema master role.

**NOTE**

If you make changes to the schema on a domain controller that is not a schema master role domain controller, the changes are not saved, and you receive the following error:

The displayable status could not be changed

## The Active Directory Schema Snap-In

You can manage the schema master from the Active Directory Schema snap-in.

To determine whether the Active Directory Schema snap-in is connected to the schema master, perform the following steps:

1. In the console tree, right-click the schema node and select Operations Master.
2. If the Current Focus and Current Operations Master values are the same, you are connected to the schema master.
3. Also, to enable schema changes, you must select the check box labeled The Schema May Be Modified on This Domain Controller (as shown in Figure 9.2).

After you complete these steps and check the box for schema modifications in the Operations Master window, you can check the schema master computer for the following Registry key:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS\Parameters\  
➤ Schema Update Allowed REG_DWORD 0x00000001
```



**FIGURE 9.2**

*In this example, dc1.kevinkocis.com hosts the schema master role. The check box allows for modifications locally at this server. Here, you are also connected locally because the Current Focus and Current Operation values are the same.*

## Domain Naming Master

The domain controller that has the domain naming master role is the only domain controller that can add and remove domains in the forest. It can also add or remove cross-reference objects to external directories.

By connecting to the domain controller holding the domain naming master role, you can add a domain to or remove it from the forest. If the domain naming master is unavailable, you cannot add or remove domains.

### NOTE

See “Performing Operations Master Role Transfers” later in this chapter for detailed instructions and cautions regarding role transfers in forests and domains.

When the domain naming master creates an object representing a new domain, it must ensure no other object has the same name. For this reason, the domain naming master should also be a Global Catalog server. The domain naming master verifies duplicates by running on a Global Catalog server, which contains a partial replica of every object in the forest.

## The Active Directory Installation Wizard

To add a domain to a forest, you can use either the Active Directory Installation Wizard or the `ntdsutil` command-line tool. When you use the Active Directory Installation Wizard, the wizard contacts the domain naming master by RPC to create the domain. You must be an enterprise administrator to create a domain.

If the domain naming master is unavailable, a message similar to the following appears:

```
Active Directory Installation Failed
The operation has failed because <reason for failure>
To perform the requested operation, the Directory Service needs to contact the
domain naming master (server dc1.Kevinkocis.com). The attempt to contact it
failed.
The error was: "The specified server cannot perform the requested operation."
```

In this example, `dc1.Kevinkocis.com` is the domain naming master.

## The `ntdsutil` Command-Line Tool

The other option for adding or removing a domain to or from a forest is to use the `ntdsutil` command-line tool.

**NOTE**

You can find ntdsutil in the Windows 2000 Resource Kit from Microsoft.

Under the Domain Management option in ntdsutil, type **precreate** followed by the correct DC and DNS to create the new object. Then use the Active Directory Installation Wizard. You can connect to the domain naming master using the ntdsutil tool to create a cross-reference object that names the new domain. The cross-reference object is located in the Partitions container of the Configuration directory partition. After the cross-reference object is replicated throughout the forest, you can run the Active Directory Installation Wizard to create the new domain using the newly created domain name. When you precreate the cross-reference object, the Active Directory Installation Wizard does not require a connection to the domain naming master to create the first domain controller of the domain. You must have sufficient access permissions to create a domain.

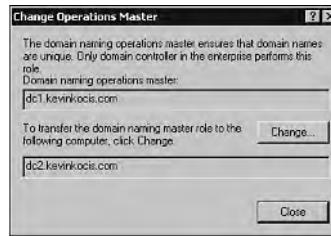
If the domain naming master is unavailable when the ntdsutil tool attempts to connect to it, a message similar to the following appears, with user input shown in bold type:

```
C:\>ntdsutil
ntdsutil: domain management
domain management: connections
server connections: connect to dc1.Kevinkocis.com
binding to dc1.Kevinkocis.com ...
DsBindW error 0x6ba(The RPC server is unavailable.)
```

To determine the domain naming FSMO holder in a forest, perform the following steps:

1. Open Active Directory Domains and Trusts.
2. Right-click the Active Directory Domains and Trusts node and then select Operations Master to view the domain controller holding the domain naming master role in the forest.

Figure 9.3 shows the Change Operations Master window for the domain naming operations role. Note that the current master and the local machine are the same; therefore, you cannot transfer the role.

**FIGURE 9.3**

*The Change Operations Master window.*

## Relative Identifier Master

You can create a new security principal object (User, Group, or Computer) on any domain controller. However, after you create several hundred security principal objects, a domain controller must communicate by means of RPC with the domain controller holding the domain's RID master role before creating the next security principal object. Then another several hundred security principal objects can be created, and when this set of objects has been created, the process of contacting the RID master is repeated. If a domain controller's RID pool is empty, and the RID master is unavailable, you cannot create new security principal objects on that domain controller. For this reason, it is imperative that a RID master be available or online when you are creating a significant number of security principal objects.

When you use the Active Directory Users and Computers snap-in to create new objects, an error message appears when the domain controller's RID pool is empty and the domain's RID master is unavailable. To avoid this error, you should ensure that the DC with this role is online and operational prior to adding a large number of security principal objects.

To move objects from one domain (the source domain) to another (the destination domain) using the `movetree.exe` command-line tool, you must connect to the domain controller holding the source domain's RID master role. If the RID master is unavailable, objects cannot be moved to other domains.

### NOTE

`Movetree.exe` can be found in the `support.cab` file, which is located in the `Support/Tools` directory on the Windows 2000 Server CD. For more information about `movetree.exe`, see the Appendix, "Common Active Directory Utilities."



If you attempt to move an object from one domain to another using the `movetree.exe` tool and you specify a source domain controller that is not the RID master, you get an unspecific `Movetree failed` error message. Cross-domain object moves originate on the RID master to prevent Active Directory from creating two objects in different domains with the same unique identifier. (This situation could occur if an object were simultaneously moved from two domain controllers to two different domains.)

Each Windows 2000 domain controller in a domain has a pool of RIDs it is allowed to assign to security principals it creates. In addition, the domain has a pool of RIDs that have never been assigned to a domain controller. When the number of RIDs in a domain controller's RID pool falls below a threshold, that domain controller submits background requests for additional RIDs from the domain's RID master. The domain's RID master removes RIDs from the domain's RID pool and assigns these RIDs to the pool of the requesting domain controller.

**NOTE**

Refer to the monitoring performance in Chapter 8 for a review of RID pool management and monitoring.

**NOTE**

In mixed mode with backup domain controllers (BDCs) still around, RID and PDCE must be on the same machine. I'll address the role of the primary domain controller emulator next.

## Primary Domain Controller Emulator (PDCE)

Windows 2000 interoperates with Windows NT 3.51 and 4.0 workstations, member servers, and domain controllers. Therefore, one domain controller in a Windows 2000 system must serve as primary domain controller for backward compatibility with these systems. The domain controller with this capability is called the primary domain controller emulator role.

Active Directory uses multi-master replication for most directory updates. This means that unavailability of the primary domain controller emulator does not have the same impact as unavailability of the primary domain controller in Windows NT 3.51 and 4.0. If the primary domain controller emulator is unavailable, you may experience issues in the following areas:

- When a user attempts a password change using a Windows NT Workstation 3.51-based computer or a computer running Windows NT Workstation 4.0, Windows 95, or Windows 98 without the Active Directory client installed, that user sees a message similar to the following: Unable to change password on this account. Please contact your system administrator.
- In a mixed-mode domain, the event logs of Windows NT 3.51 or 4.0 backup domain controllers contain entries showing failed replication attempts.
- In a mixed-mode domain, trying to start User Manager on a Windows NT 3.51 or 4.0 backup domain controller results in a domain unavailable error message. If User Manager is already running, you see an RPC server unavailable message. Attempting to create an account using the `net user /add` command results in a could not find domain controller for this domain message. When you run Server Manager, you see a message similar to the following: Cannot find the primary domain controller for <domain name>. You may administer this domain, but certain domain-wide operations will be disabled.

As you upgrade systems to Windows 2000 or install the Active Directory client for Windows 9x (a Windows NT 4.0 client is not available yet), they no longer depend on the primary domain controller. Instead of making password changes at the primary domain controller emulator, clients update passwords at any domain controller in the domain. The upgraded clients also use Active Directory instead of the Windows NT Computer Browser service to locate network resources. When all backup domain controllers in a domain are upgraded to Windows 2000, the primary domain controller emulator does not receive any Windows NT 3.51 or 4.0 replication requests.

Even after all systems are upgraded to Windows 2000, the domain controller holding the primary domain controller emulator role still performs the following functions:

- Password changes performed by other domain controllers in the domain are sent to the primary domain controller emulator first. This is called preferential replication.
- When an authentication fails with an invalid password at other domain controllers in the domain, the authentication request is retried at the primary domain controller emulator before failing. If a recent password update has reached the primary domain controller emulator, the retried authentication request should succeed.
- When an authentication succeeds on an account for which the most recent authentication attempt at the domain controller failed, the domain controller communicates this fact (zero lockout count) to the primary domain controller emulator.

Therefore, when the primary domain controller emulator is unavailable, you may experience password difficulties from legacy clients.

## Infrastructure Master

The domain controller hosting the infrastructure master role for the domain is responsible for updating cross-domain group-to-user references. The infrastructure master updates these references locally and uses replication to bring all other replicas of the domain up-to-date. For example, if you add a user to a group in the same domain by using the Active Directory Users and Computers snap-in, you can immediately view the group membership and see the user you just added. If you rename the user object (change its `cn` attribute) and then display the group membership again, you instantly see the user's new name in the list of group members.

However, if the user and group are located in different domains, a time lag occurs between the time you rename a user object and the time a group containing that user displays the user's new name. Until it is updated, the user's old name appears. This time lag is inevitable in a distributed environment with locally administered sites.

### NOTE

The infrastructure master role must not be hosted on a Global Catalog (GC) server because this role looks to the Global Catalog for object references it might not contain. However, in a single domain enterprise, the infrastructure master role serves no function, so it doesn't matter if it resides on a GC server.

For example, when an object on one domain controller references an object that is not on that domain controller, it represents that reference as a record containing the globally unique identifier (GUID), the security identifier (SID) for security principals, and the distinguished name of the object being referenced. If the referenced object moves, its GUID does not change, its SID changes if the move is across domains, and its distinguished name always changes.

The infrastructure master occasionally examines the references of its directory replica against objects not held on that domain controller. It queries a Global Catalog server for current information about the distinguished name and SID of each referenced object. If this information has changed, the infrastructure master makes the change in its local replica and also replicates the new values to other domain controllers within the domain.

If the infrastructure master runs on a Global Catalog server, it never updates anything because it does not contain any references to objects that it does not hold. Remember, a Global Catalog server only holds a partial replica of every object in the forest.

To determine the RID, PDC, and Infrastructure FSMO holders of a selected domain, perform the following:

1. Open Active Directory Users and Computers.
2. In the console tree, right-click the domain object and then click Operations Masters.
3. Click the RID tab to view the name of the server holding the RID master role (see Figure 9.4).
4. Click the PDC tab to view the name of the server holding the PDC master role.
5. Select the Infrastructure tab to view the name of the server holding the infrastructure master role.

Figure 9.4 shows the domain operations roles (RID, PDC, and Infrastructure). Selecting the various tabs displays information about the current role masters and transfer information.



**FIGURE 9.4**

*The domain operations roles.*

## NOTE

You may need to set Advanced Features in the Active Directory Users and Computers to view the Operation Masters option. Simply check the Advanced Features option under the View menu in the console.

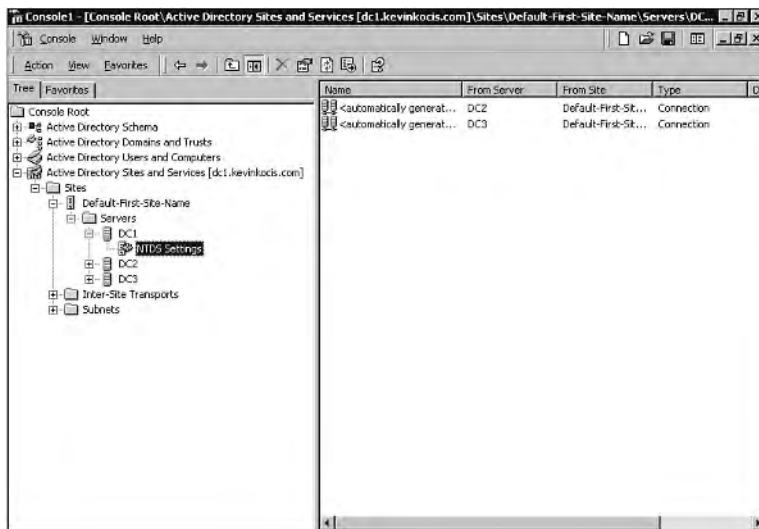
## Placing Flexible Single-Master Operations

Windows 2000 performs an initial placement of operations master roles on domain controllers. This placement works well for a forest deployed on a few domain controllers in a single site. In a forest with more domain controllers or multiple sites, you need to plan the placement of operations master roles to match your replication and network topologies.

Start planning your FSMO placements on a domain-by-domain basis. Obviously, if you're working in a single domain environment with only one domain controller, that domain controller holds all the per-domain roles. If your domain has several domain controllers, you can transfer roles to domain controllers that are direct replication partners. If you're still running in a mixed-mode domain, those two domain controllers should be located within the same site.

### The Active Directory Sites and Services Snap-In

To locate the direct replication partners of a domain controller, use the Active Directory Sites and Services snap-in. Open the sites container and locate the domain controller. Expand the Server object to view the NTDS Settings object beneath it and click the NTDS Settings object. The details pane displays a list of Connection objects. The From Server attribute of each Connection object identifies the direct replication partner of the domain controller. Figure 9.5 shows the Active Directory Sites and Services snap-in. Note the intrasite replication partners for dc1 (dc2 and dc3) found in the NTDS Settings object.



**FIGURE 9.5**

*The Active Directory Sites and Services snap-in.*

You can name one of the two domain controllers you have chosen as the Operations master domain controller for the domain and another the Standby operations master domain controller for the domain.

## Per-Domain Role Placements

In small domains, you should host both the RID master and primary domain controller emulator roles on the Operations master domain controller. In a large domain, on the other hand, you can reduce the workload on the primary domain controller emulator by hosting the RID master and primary domain controller emulator roles on separate domain controllers. These domain controllers need to be direct replication partners of the Standby operations master domain controller. Remember to keep the two roles together unless the workload on your Operations master domain controller justifies the extra management burden of separating the roles.

### NOTE

Remember that the infrastructure master role should not be hosted on a domain controller that is a Global Catalog server. Also, it will require a good network connection to a Global Catalog server from any domain, although having it within the same site is ideal. If the infrastructure master role is held by a domain controller that is a Global Catalog server, cross-domain object references in that domain are not updated. If all domain controllers in a domain are Global Catalog servers, it does not matter which domain controller holds the infrastructure master role.

## Per-Forest Role Placements

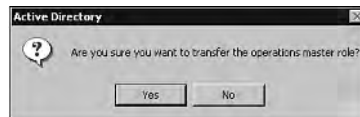
After you plan all the per-domain roles, you should then plan the per-forest roles. The key to remember is that the schema master and domain naming master roles should always be placed on the same domain controller. This domain controller should be a Global Catalog server. To simplify management, you can place these roles on the Operations master domain controller of a domain.

## Performing Operations Master Role Transfers

To transfer an operations master role is to move it with the cooperation of its current owner. Given a role placement plan, you need to transfer each role from its default location to its planned location. Depending on the role, you can transfer roles by using one of three Active Directory snap-ins:

- Active Directory Schema for the schema master
- Active Directory Domains and Trusts for the domain naming master
- Active Directory Users and Computers for per-domain roles

To transfer a role, first focus the Active Directory snap-in on the domain controller that needs to receive the role. Then right-click the snap-in node in the console tree and select Operations Master. For per-domain roles, you then select the tab corresponding to the specific role you want to transfer. The property page displays the Current Focus (the domain controller on which the snap-in is focused), the Current Operations Master (the domain controller that is the current role owner), and the online/offline status of the current role owner. Click Change and then click Yes to complete the operation. Figure 9.6 shows the transfer confirmation window. In this example, the RID operations master role is being transferred from dc1 to dc3.



**FIGURE 9.6**

*The transfer confirmation window.*

If the current role owner is available, the transfer is completed within a few seconds. If the transfer is not completed within a short period of time, the domain controller is not available. If this situation occurs, refer to the recommendations for failure responses explained in the next section.

Additionally, you can use the Active Directory snap-ins to view the actual roles that a domain controller owns. To do so, you choose one of the Active Directory snap-ins, right-click the root node of the snap-in in the console tree, and select Operations Master. The Operations dialog box displays the name of the domain controller that has the current focus and shows its status.

In mixed-mode domains that contain backup domain controllers, the Standby operations master domain controller should be in the same site as the primary domain controller emulator. If you keep both domain controllers in the same site, you can avoid having the system perform a full synchronization with the backup domain controllers (in the event you must seize the PDC emulator role to the Standby operations master domain controller).

## Using the ntdsutil Tool for Role Placement

The ntdsutil tool allows you to transfer and seize operations master roles. The ntdsutil tool might be more convenient for operations master transfers and seizures than the graphical user interface tools because entering commands is simpler and quicker than using multiple windows.

**NOTE**

To perform seizures of the schema master, domain naming master, and RID master roles, you are required to use the ntdsutil tool.

When you use the ntdsutil command-line tool to seize an operations master role, the tool attempts a transfer from the current role owner first. Then, if the existing operations master is unavailable, it performs the seizure.

The ntdsutil tool provides help information when you type a question mark (?). Figure 9.7 shows the ntdsutil command-line help screen with the help (?) options.



```

C:\WINNT\system32\ntdsutil.exe: ?

?               - Print this help information
Authoritative restore - Authoritatively restore the DIT database
Domain management - Prepare for new domain creation
Files           - Manage NTDS database files
Help           - Print this help information
IPDeny List    - Manage LDAP IP Deny List
LDAP policies  - Manage LDAP protocol policies
Metadata cleanup - Clean up objects of decommissioned servers
Popups %s      - <en/dis>able popups with "on" or "off"
Quit           - Quit the utility
Roles          - Manage NTDS role owner tokens
Security account management - Manage Security Account Database - Duplicate SID Cleanup
Semantic database analysis - Semantic Checker

C:\WINNT\system32\ntdsutil.exe:

```

**FIGURE 9.7**

*The ntdsutil command-line help screen with the help (?) options.*

On this screen, the available ntdsutil tool commands are displayed after you enter a question mark (?). To transfer an operations master role, you enter the **roles** command, which displays the FSMO maintenance menu. Entering a question mark (?) displays the subcommands within the FSMO maintenance menu. Before transferring the operations master role, you must connect to the domain controller that will receive the role (reskit1 in the previous example) by entering the **connect to server** subcommand. Then, after leaving the server connections mode by entering **quit**, you issue the transfer domain naming master command. A confirmation pop-up window appears for the transfer domain naming master operation.



**NOTE**

You must have sufficient permissions to execute commands using the `ntdsutil` tool. You also can view the current operations master role owner by using the `ntdsutil` command-line tool from the Select Operation Target menu located under the Roles option. If you choose the List Roles for Connected Server command, you see a list of all the current operations master role owners.

## Using the `ntdsutil` Tool for Role Location

`ntdsutil` is the exclusive tool to reveal all the FSMO role hosts. To view all the current FSMO roles in your enterprise, perform the following:

1. Click Start, Run. Then type `cmd` and press Enter.
2. Type `ntdsutil` and press Enter.
3. Type `domain management` and press Enter.
4. Type `connections` and press Enter.
5. Type `connect to server <ServerName>`, where `<ServerName>` is a domain controller, and then press Enter.
6. Type `quit` and press Enter.
7. Type `select operation target` and press Enter.
8. Type `list roles for connected server` and press Enter.

You then see output similar to the following:

```
C:\WINNT\system32\ntdsutil.exe: domain management
domain management: connections
server connections: connect to server dc1
Binding to dc1 ...
Connected to dc1 using credentials of locally logged on user
server connections: quit
domain management: select operation target
select operation target: list roles for connected server
Server "dc1" knows about 5 roles
Schema - CN=NTDS Settings,CN=DC1,CN=Servers,CN=Default-First-Site-Name,
➤CN=Sites,CN=Configuration,DC=kevinkocis,DC=com
Domain - CN=NTDS Settings,CN=DC1,CN=Servers,CN=Default-First-Site-Name,
➤CN=Sites, CN=Configuration, DC=kevinkocis, DC=com
PDC - CN=NTDS Settings,CN=DC1,CN=Servers,CN=Default-First-Site-Name,CN=Sites,
➤CN=Configuration,
DC=kevinkocis,DC=com
```

RID - CN=NTDS Settings,CN=DC1,CN=Servers,CN=Default-First-Site-Name,CN=Sites,  
➔CN=Configuration,  
DC=kevinkocis,DC=com  
Infrastructure - CN=NTDS Settings,CN=DC3,CN=Servers,CN=Default-First-Site-Name,  
➔CN=Sites, CN=Configuration, DC=kevinkocis, DC=com

## Schema Master Permission Changes

Schema Admins is the only group of users with privileges to change the schema master FSMO role. This right can be changed in one of the following two ways:

- Open the Schema Manager snap-in, right-click Active Directory Schema Manager, and then click Permissions. Use the Change Schema Master permission to designate rights.
- Using the adsiedit tool from the Windows 2000 Resource Kit, you can change the rights by right-clicking Schema Naming Context and then clicking Properties. Use the Change Schema Master permission to designate rights.

## Domain Naming Master Permission Changes

Members of the Enterprise Admins group are the only users with privileges to change the domain naming master. This right can be changed by using the adsiedit tool from the Windows 2000 Resource Kit. Change the rights by right-clicking CN=Partitions under Configuration Context and then clicking Properties. Use the Change Domain Master permission to designate rights.

## PDCE Permission Changes

Members of the Domain Admins group are the only users with privileges to change the primary domain controller emulator (PDCE). This right can be changed by using the adsiedit tool from the Windows 2000 Resource Kit. Change the rights by right-clicking DC=dc1,DC=kevinkocis,DC=com (for dc1.kevinkocis.com) under the Domain context and then clicking Properties. Use the Change PDC permission to designate rights.

## Infrastructure Master Permission Changes

Members of the Domain Admins group are the only group of users with privileges to change the infrastructure master. This right can be changed by using the adsiedit tool from the Windows 2000 Resource Kit. Change the rights by right-clicking CN=Infrastructure for the folder under the Domain context and then clicking Properties. Use the Change Infrastructure Master permission to designate rights.

## RID Master Permission Changes

Members of the Domain Admins group are the only group of users with privileges to change the RID master. This right can be changed by using the adsiedit tool from the Windows 2000 Resource Kit. Change the rights by right-clicking CN=RID Manager\$ in the CN=System folder under the Domain context and then clicking Properties. Use the Change RID Master permission to designate rights.

You can also change the RID master, PDC emulator, and infrastructure master in the Active Directory Users and Computers snap-in by right-clicking the domain item and then clicking Operations Master.

## Controlling Role Transfers

As previously defined, an operations master role transfer is the movement of a role with the cooperation of its current owner. If you want to perform a role transfer, both domain controllers must be available and networked.

The ability to perform a role transfer is controlled through a special object permission on the role object itself at the new role owner. Each role has a different object permission, granted by default to a particular group of administrators, as indicated in Table 9.2.

**TABLE 9.2** FSMO Permission Groups

<i>FSMO</i>	<i>Permission</i>	<i>Default Group</i>
Schema master	Change schema master	Schema admins
Domain naming master	Change domain master	Enterprise admins
RID master	Change RID master	Domain admins
PDC emulator	Change PDC permission	Domain admins
Infrastructure master	Change infrastructure master	Domain admins

If you attempt to perform a role transfer and do not have sufficient permissions, an error occurs.

If the need arises, you can change the group of administrators that is able to perform specific role transfers. For example, you might decide to create a new group called Domain Naming Role Admins that has exclusive permission to transfer the domain naming master role. In this case, you would create the group and then use adsiedit to find the domain naming master role object. Next, you would display the object properties, remove the Change Domain Master permission for Enterprise Admins, and add the Change Domain Master permission for Domain Naming Role Admins. In this way, you can precisely control the set of administrators who can transfer the domain naming master role.

The act of changing who can transfer a role does not change who can *use* the role. In the preceding example, the Domain Naming Role Admins can transfer the domain naming master role, but they cannot create cross-reference objects; only Enterprise Admins can do that.

#### NOTE

In a properly configured directory, only a small number of administrators should have the right to perform operations master role transfers.

## Controlling Role Seizures

As previously defined, role seizure is the movement of a role without the cooperation of its current owner. As a rule, role seizure should be avoided, but sometimes it is required.

A role seizure is controlled through the same per-role object permissions that control role transfers, plus the Write fsmoRoleOwner property permission at the new role owner.

#### NOTE

To seize a role, you need both the per-role object permission and the Write fsmoRoleOwner property permission. By default, the Write fsmoRoleOwner property permission is granted to the same groups that are granted the per-role object permissions.

If you are Visual Basic savvy, you can also place operations master role owners programmatically for both role transfers and seizures through Microsoft Visual Basic Script programs.

Active Directory operations master role transfers are exposed as LDAP update operations to a root DSE operational attribute of the domain controller taking the role. A root DSE operational attribute corresponds to each role:

- becomeSchemaMaster
- becomeDomainMaster
- becomeRidMaster
- becomePdc
- becomeInfrastructureMaster

For example, by running the following Visual Basic Script program using the `CScript` command on a domain controller, you can transfer the domain naming master role to that domain controller. The following is an example of the `CScript`:

```
Set dse = GetObject("LDAP://localhost/RootDSE")
dse.Put "becomeDomainMaster", 1
dse.SetInfo
```

Active Directory role seizures are exposed as LDAP update operations to the `FSMO-Role-Owner` attribute of the role object on the domain controller seizing the role.

For example, by running the following Visual Basic Script program using the `CScript` command on a domain controller, you can seize the domain naming master role to that domain controller. Here is another example:

```
Dim dse, roleObject, ntdsDsa
Set dse = GetObject("LDAP://localhost/RootDSE")
Set roleObject = GetObject("LDAP://localhost/" &
    "CN=Partitions," &
    dse.Get("configurationNamingContext"))
Set ntdsDsa = dse.Get("dsServiceName")
roleObject.Put "fSMORoleOwner", ntdsDsa
roleObject.SetInfo
```

## Operations Masters Troubleshooting

Although FMSO operations typically are not very complex, in some situations errors may require significant attention. In the event an operation master is unreachable or unavailable, you must quickly access the outage and determine an appropriate course of corrective action. Let's take a closer look at these situations.

### Responding to Operations Master Failures

The first step in responding to the unavailability of a domain controller that is an operations master role owner is to determine the anticipated duration of the outage.

If the outage is expected to be brief, wait for the role owner to become available before performing a role-related function. If the outlook is grim, you might want to seize the operations master role from a domain controller. When you seize a role, you are transferring it without the cooperation of its current owner.

#### TIP

It is best to avoid seizing roles. The decision to seize an operations master role depends on the role and the expected length of the outage.

## Primary Domain Controller Emulator Failures

The loss of a domain controller that is the primary domain controller emulator role can be visible to any user or administrator. Down-level clients without the Active Directory client cannot change their passwords without communicating with the primary domain controller emulator. If a user's password has expired, that user cannot log on.

In this situation, if the primary domain controller emulator is offline for a significant period of time and the domain does not have domain controllers running earlier versions of Windows NT, you should seize the primary domain controller emulator role to the Standby operations master domain controller.

Agree to the seizure confirmation only if you know the current primary domain controller emulator will be offline for a significant period. If the original primary domain controller emulator comes back online, you can then transfer the role back to the original role owner.

## Infrastructure Master Failures

Temporary loss of a domain's infrastructure master is not apparent unless you recently moved or renamed a significant number of accounts.

While a temporary loss of the infrastructure master is not a problem worth fixing, you may need to take action if the outage of a domain's infrastructure master is long term or permanent. Select a domain controller that is not a GC server and that has good network connectivity to a GC server located in any domain (but preferably in the same site). After you select the domain controller, seize the infrastructure master role to this domain controller.

Agree to the seizure confirmation only if you know that the current infrastructure master will be offline for a very long period. When the original infrastructure master comes back online, you can transfer the role back.

## Other Operations Master Failures

Temporary loss of the schema master, domain naming master, or RID master is ordinarily not apparent and does not usually inhibit your work as an administrator.

If you anticipate a significant and permanent outage of the domain controller holding one of these roles (like failed hardware), you can seize that role to the Standby operations master domain controller.

**CAUTION**

You should proceed with caution because seizing any of these roles is a drastic step. A domain controller whose schema master, domain naming master, or RID master role is seized must never come back online. Before you proceed with the role seizure, disconnect the domain controller from the network.

The domain controller that seizes the role should be current with respect to updates performed on the previous role owner. Because of replication latency, the domain controller might not be up-to-date.

To check the status of updates for a domain controller, you can use the repadmin command-line tool, which is a Resource Kit tool that performs replication diagnostics. It is available on the Microsoft Windows 2000 Server installation CD. repadmin can determine whether a domain controller has the most current updates.

For example, to make sure a domain controller is fully up-to-date, suppose that dc1 is the RID master of the domain Kevinkocis.com, dc2 is the Standby operations master domain controller, and server1 is the only other domain controller in the Kevinkocis.com domain. Using the repadmin tool, you would issue the following commands as indicated by the bold type (I'll use rounded numbers for simplicity's sake):

```
C:\>repadmin /showvector dc=kevinkocis,dc=com server1.Kevinkocis.com
```

The resulting output would be

```
Chicago\dc1      @ USN 1800
Brazil\server1   @ USN 2400
```

Here's another example:

```
C:\>repadmin /showvector dc=kevinkocis,dc=com server1.Kevinkocis.com
```

The resulting output for this command is

```
Chicago\dc1      @ USN 1600
Phoenix\dc2      @ USN 2700
```

Ignore all output lines except those for dc1. dc2's up-to-date status value with respect to dc1 (dc1 @ USN 1800) is larger than server1's up-to-date status value with respect to dc1 (dc1 @ USN 1600), making it safe for dc2 to seize the RID master role formerly held by dc1. If the up-to-date status value for dc2 is less than the value for server1, you wait for normal replication to update dc2, or use the repadmin tool's /sync/force commands to make the replication happen immediately.

After you determine that the role owner is fully up-to-date, you can seize the operations master role by using the ntdsutil tool.

Here is an example:

```
C:\>ntdsutil
ntdsutil: roles
fsmo maintenance: connections
server connections: connect to dc2.kevinkocis.com
binding to dc2.Kevinkocis.com ...
Connected to dc2.Kevinkocis.com
using credentials of locally logged on user
server connections: quit
fsmo maintenance: seize RID master
Server "dc2.Kevinkocis.com" knows about 5 roles
Schema - CN=NTDS Settings,CN=server2,CN=Servers,
CN=Chicago,CN=Sites,CN=Configuration,DC=kevinkocis,DC=com
Domain - CN=NTDS Settings,CN=server2,CN=Servers,
CN=Chicago,CN=Sites,CN=Configuration,DC=kevinkocis,DC=com
PDC - CN=NTDS Settings,CN=server3,CN=Servers,
CN=Tampa,CN=Sites,CN=Configuration,DC=kevinkocis,DC=com
RID - CN=NTDS Settings,CN=server10,CN=Servers,
CN=Tampa,CN=Sites,CN=Configuration,DC=kevinkocis,DC=com
Infrastructure - CN=NTDS Settings,CN=server1,CN=Servers,
CN=Phoenix,CN=Sites,CN=Configuration,DC=kevinkocis,DC=com
fsmo maintenance: quit
ntdsutil: quit
```

## Troubleshooting and Technical Details

You can use the following list to obtain more complete technical explanations concerning the management of operations masters:

- When you back up a domain controller, you back up the roles it owns. Subsequently, when you restore a domain controller from backup media, you restore the roles it owns.
- When you remove Active Directory (that is, dcpromo.exe demotion) from the domain controller that owns the operations master roles, the domain controller attempts to “abandon” its roles. For each role the domain controller holds, it locates another available domain controller for the role and transfers the role to it. If another domain controller is not available during the demotion, the demotion process will fail. If the domain controller is the last in the domain, dcpromo.exe will prompt you to this fact. In this case you should bring up another DC to inherit the roles.
- Make sure you transfer any roles before you begin the demotion process so that role placements are correct. Do not rely on the transfer feature when you’re removing Active Directory from a domain controller.



## Other FSMO Errors and Clarifications

One error can occur when a domain controller or Windows 2000 client attempts to synchronize with an external time server, known as an external time server error. When you use the `net time` command, you may receive the following error message:

```
Could not locate a time-server.  
More help is available by typing NET HELPMSG 3912
```

This error message occurs even if you set a valid Simple Network Time Protocol (SNTP) time server using the `net time /setsntp` command, and network connectivity to the external time server exists.

Windows 2000 servers that are not domain controllers and Windows 2000 clients attempt to locate a domain controller to synchronize the network time. Domain controllers attempt to contact the domain controller hosting the primary domain controller emulator FSMO role. Only the domain controller that holds the PDC FSMO role can query an external time source to set the time. To resolve this issue, make sure you query an external time source only from the PDC emulator FSMO.

### Infrastructure Error

The infrastructure error may be common and is usually an oversight in FSMO planning. As I mentioned earlier in the chapter, in a multi-domain enterprise, the infrastructure master role cannot be located on a Global Catalog server. Event error 1419 may be generated because there may be problems with an Infrastructure FSMO role holder performing its duties if it is also a Global Catalog server.

The error message in Event Viewer may be similar to the following sample:

```
Event ID: 1419  
Event Type:Error  
Event Source:NTDS General  
Event Category:Directory Access  
Event ID: 1419  
Date:3/13/2000  
Time:11:42:18 AM  
User:Everyone  
Computer:          Server1
```

In this case, this DC is both a Global Catalog and the infrastructure update master. These two roles are incompatible. If another machine exists in the domain, it should be made the infrastructure update master. The machine `CN=NTDS (Settings,CN=Server2,CN=Servers, CN=Chicago,CN=Sites,CN=Configuration,DC=PRODOM,DC=com)` is a good candidate for this role. If all domain controllers in this domain are Global Catalogs, then there are no Infrastructure Update tasks to complete, and this message may be ignored.

The error message is generated after another domain controller is installed in the domain. This is most likely to occur on the first domain controller in the forest because it holds all five FSMO roles and is also a Global Catalog server.

You can resolve this issue in two ways:

- Transfer the Infrastructure FSMO role to another domain controller in the domain.
- Enable another domain controller in the forest to be the Global Catalog server and disable this computer from being a Global Catalog server.

## RID Error

If you notice the following event occurring frequently in the NTDS event log, you may have a RID master role issue:

```
Event 1655  
MessageId=0x410A  
SymbolicName=SAMMSG_RID_INIT_FAILURE
```

Language=English

Here, the account-identifier allocator failed to initialize properly. The record data contains the NT error code that caused the failure. Windows 2000 will retry the initialization until it succeeds; until that time, account creation will be denied on this domain controller. Be sure to look for other SAM event logs that may indicate the exact reason for the failure.

This error may occur if the RID master FSMO is offline or is experiencing replication problems. The issue is that the domain controller cannot initialize the RID pool.

In this situation, you should verify that the RID master FSMO is online and then check the NTDS event log for any details indicating replication issues before looking into seizing the FSMO role.

## Summary

Flexible Single-Master Operations are dedicated roles assigned to specific Windows 2000 domain controllers in an enterprise. Of the five total roles, two are per-forest roles (schema and domain naming), and three are per-domain roles (RID, PDC, and infrastructure). The two per-forest roles need to be placed on the same domain controller, and the three other roles can go on another domain-specified controller. It is recommended that per-domain roles be hosted on the same domain controller, but the infrastructure role may be transferred if loads affect performance. Roles may be transferred, but you should avoid seizures because domain controllers hosting certain roles that have been seized cannot come back online.

# Active Directory Reliability and Optimization

CHAPTER

# 10

## IN THIS CHAPTER

- Utilities 280
- Active Directory Backup 280
- Performing an Active Directory Backup 281
- Active Directory Restore 284
- Monitoring Active Directory  
Performance 295

Because Active Directory is such a critical piece to your Windows 2000 implementation, making it as reliable and optimal as possible is imperative. Also, because AD is a mission-critical component of your Windows 2000 architecture, integrating a solid backup and recovery strategy into your environment is imperative. This chapter addresses Active Directory backup and restore, as well as optimization and monitoring.

## Utilities

The GUI utility, appropriately called Backup, backs up and restores Active Directory. Backup can perform live backups. The command-line utility called Ntbackup (addressed in previous chapters) partners with Backup to provide more granular control over Active Directory restores.

## Active Directory Backup

You need to back up Active Directory data for several reasons, and they are fairly straightforward. Backups can be critical in some of the following situations:

- In the event of hardware failure on a domain controller
- In the event of accidental or intentional data modification or deletion
- As a way to maintain historical information

The Backup tool can back up Active Directory

- While the domain controller is online
- With other system and data files (called the System State)
- With batch file commands (if desired)
- To removable media, network drives, or files

Active Directory supports only a normal backup, which backs up the entire system while the domain controller is online. A normal backup clears the Archive attribute of the file and truncates the log files of database applications. It also ensures that a restore requires only a single source (as opposed to multiple sources to restore from multiple incremental or differential backups).

The Backup tool automatically backs up other system components and services critical to Active Directory. These components and services (which include Active Directory) are collectively known as the System State data.

The System State data on a domain controller consists of the following files:

- The system startup files and Registry
- The class registration database of COM+
- The File Replication Service (the SYSVOL directory)
- The Certificate Services database (if installed)
- The Domain Name System (if installed)
- The Cluster service (if installed)
- Active Directory

Active Directory includes the following files:

- Ntds.dit—The Active Directory database
- Edb.chk—The checkpoint file
- Edb\*.log—The transaction logs, each 10MB in size
- Res1.log and Res2.log—Reserved transaction logs

#### NOTE

Active Directory cannot be backed up without the System State data.

## Performing an Active Directory Backup

To back up Active Directory, you start with accessing the Backup utility by choosing Start, Programs, Accessories, System Tools, Backup. You can also access Backup at the command prompt by typing **Ntbackup**. Remember, you can back up more than Active Directory and System State data. The Backup utility offers the option to back up additional local and network drives and files, and create an emergency repair disk (ERD).

#### NOTE

To back up the System State data, you must be either a Backup Operator or an Administrator.

## Using the Backup Utility's Backup Wizard

The Backup utility's Backup wizard will conveniently guide you through the backup process. You can also use the Backup utility to perform a manual backup if you don't favor wizards.

To back up System State data by using the Backup Wizard, perform the following steps:

1. On the Start menu, click Run and then type **Ntbackup**.
2. On the Tools menu, click Backup Wizard.
3. On the first wizard screen (see Figure 10.1), click Next. Click Only Back Up the System State Data and then click Next.
4. Designate where you want to save the System State data, designate a media or file name, and click Next.
5. At the Completing the Backup Wizard window, you can configure advanced options by clicking the Advanced button.
6. When you're done setting options, click Finish.



**FIGURE 10.1**

*The Backup Wizard screen.*

You can access advanced backup options by clicking Advanced on the final wizard screen. The advanced options include configuration options for data verification, hardware compression, media labels, job appending, and scheduling.

Although we only backed up the System State data in the example, you should consider backing up all data on your domain controller. For disaster recovery, backing up all local and mapped drives in addition to the System State data is recommended. You can do so by running the Backup utility and choosing Back Up Everything on My Computer on the What to Back Up screen found in Backup Wizard.

To back up System State data manually by using the GUI, perform the following steps:

1. On the Start menu, click Run and then type **Ntbackup**.
2. On the Backup tab, click the check box next to System State under My Computer.
3. In the Backup Destination box, choose File or the type of media you want to use to save the System State data.
4. Enter a tape or filename in the Backup Media or File Name box.
5. Click Start Backup, edit any backup job information that you want to, and then click Start Backup again. You should then see something like Figure 10.2.



**FIGURE 10.2**

*Backup-in-progress screen.*

## NOTE

System State data does not contain Active Directory unless the server on which you're backing up System State data is a domain controller.

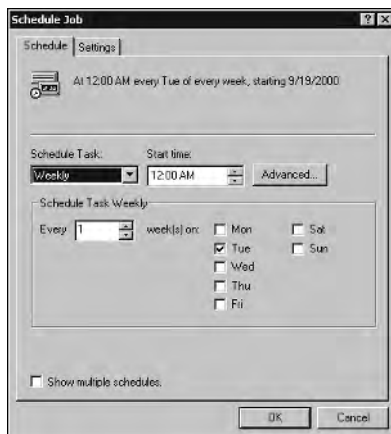
You must perform a backup on every domain controller in your enterprise to entirely back up Active Directory because you cannot back it up on a remote computer. This is a limitation of the Windows 2000 Backup utility. As a result, you may also want to consider using a third-party backup program to remotely back up and restore Active Directory.

## Scheduling Backups

You can design a backup schedule by using the Backup utility. It can be found under the Advanced option on the final wizard screen in the Backup Wizard. On the fifth and final window of the advanced options, you can schedule backups (see Figure 10.3).

**NOTE**

You must be a member of the Domain Admins or Backup Operators group to schedule backups.



**FIGURE 10.3**

*Scheduling backups in the Advanced Options Wizard.*

You can also view and schedule backups by clicking the Schedule Jobs tab in the Backup utility.

## Active Directory Restore

You can restore domain controller–replicated data in two basic ways, depending on the nature of the loss or failure. First, you can completely reinstall the operating system on the server, promote it to domain controller, and allow the normal replication process to provide Active Directory information from its replica partners. Second, you can install the operating system, and then use the Backup utility to restore replicated data from backup media without any additional configuration.

The two ways to perform Active Directory data restoration from backup media are nonauthoritative and authoritative. With a nonauthoritative restore, the metadata on a domain controller is restored and then updated through normal replication. A nonauthoritative restore is typically performed when a domain controller has completely failed because of hardware or software problems.



An authoritative restore is performed after a nonauthoritative restore. During an authoritative restore, an entire directory, subtree, or objects are selected to take priority over other copies of those objects on other domain controllers.

You would typically use an authoritative restore to revert a system to a previously known state—for example, if some Active Directory data became corrupt or was deleted. The Ntdsutil command-line utility enables you to authoritatively restore selected data.

**NOTE**

To restore System State data, you must be a Local Administrator.

## Restoring Active Directory Through Reinstallation and Replication

To restore Active Directory through reinstallation and replication, perform the following steps:

1. In Active Directory Sites and Services, delete any references to the old domain controller.
2. Reinstall Windows 2000 Server on the domain controller.
3. Promote the server to Active Directory to a domain controller (either through the wizard or dcpromo).
4. Allow Active Directory and SYSVOL to be updated via replication.

## Restoring Active Directory

You can also restore Active Directory by restoring the System State data from a file or backup media. One reason for a restore would be to revert to a premodification Active Directory state. If you are backing up files other than the System State, you may want to restore a modified or deleted file (such as an accidentally deleted Excel spreadsheet).

In the event of a severe hardware failure on the domain controller (such as failed NICs that needed to be replaced), you might need to manually reconfigure certain settings (such as network settings). In the case of failed drives, make sure the number and size of disk volumes are the same or larger than the previous system's, otherwise a restore failure will result.

The following sections examine the process of restoring Active Directory data both non-authoritatively and authoritatively.

## Nonauthoritatively Restoring Active Directory by Using the Backup Utility

The Backup utility operates in nonauthoritative restore mode. After you restore data and bring the domain controller back online, it detects that the restored data hasn't been updated since the backup was performed. The domain controller then receives and applies updates through normal replication from any domain controllers that remained online during the failure.

### NOTE

Any directory updates that occurred after the backup was created are applied after the restore as part of the normal replication process.

Replication rebuilds the data for the updates that originated on the restored domain controller between the time the domain controller was last backed up and when it was restored from backup.

## Using the Backup Utility to Restore Active Directory

If you want to restore a domain controller from backup media, Active Directory must be offline. To take Active Directory offline, you need to place the domain controller in Directory Services Restore Mode by following these steps:

1. Restart the domain controller.
2. Press F8 at the operating system selection screen.
3. Select Directory Services Restore Mode and then press Enter.

### NOTE

When you restart the computer in Directory Services Restore Mode, you must log on to the local computer as the Directory Services Restore Mode Administrator. You must use the administrator logon name and the Directory Services Restore Mode password set during the installation of Active Directory. You must use this account because local users and group accounts are never available on a domain controller, and because the netlogon service stops during safe-mode boot and account verification cannot occur for Active Directory accounts.

When the domain controller is offline, you can restore the System State data by using the Restore Wizard in the Backup utility as follows:

1. On the Start menu, click Run and then type **Ntbackup**.
2. On the Tools menu, click Restore Wizard.
3. Click Next, select the appropriate backup set for the restore, select System State, and then click Next.
4. Click Finish.

The System State data is restored to the system root by default, and replaces existing System State data on the domain controller. You can elect to change the restore locations by setting advanced restore options on the last screen of the Restore Wizard and then selecting the new location.

#### NOTE

When you restore the System State data, the location of your system root must be the same as when you backed up the System State data. If you choose an alternate location for restoring the System State data, only the system boot files, Registry files, and SYSVOL directory files are restored. Other database files and services are not restored.

Because the Backup utility restores the database and Registry settings, when it restores Active Directory, the Internet Protocol (IP) configuration is also restored. Additionally, the Domain Name System (DNS), the Certificate Services database files, and File Replication Service (FRS) are also restored.

After the restore, the File Replication Service (FRS) is reset in preparation for replication. The Active Directory is also verified. After the server reboots in normal operational mode, it checks Active Directory database files for consistency and re-indexes them. It also replicates FRS data and restores the Certificate Services database.

Due to Active Directory's association with other distributed services (such as Certificate Services and FRS), restoring Active Directory involves many steps.

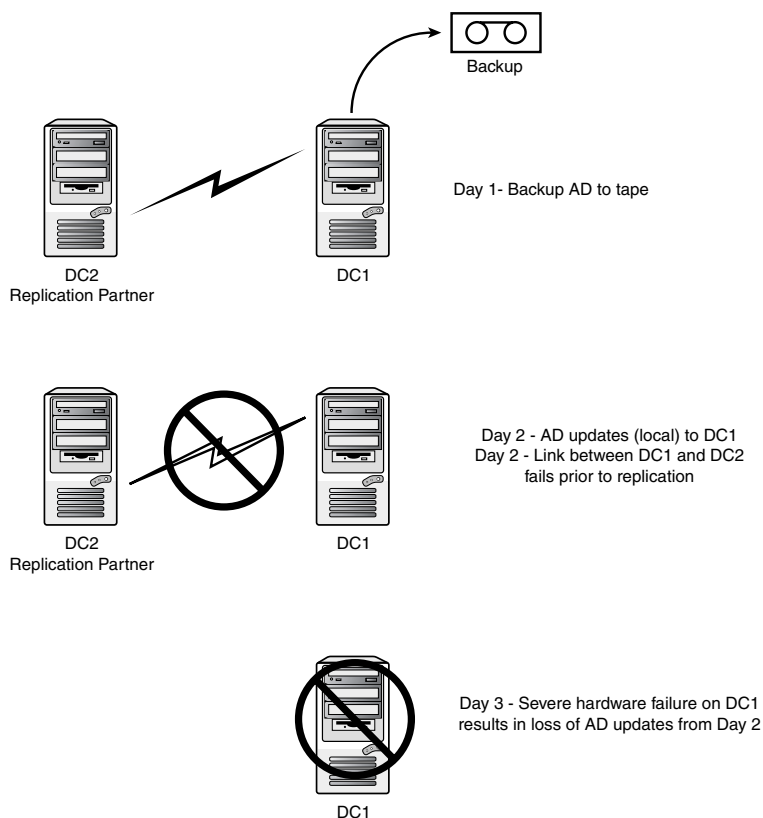
#### NOTE

Inconsistencies may result if all dependent services are not restored in the same mode and from the same backup media.

## Implications of a Nonauthoritative Restore

Using the Backup utility to restore a domain controller is considered nonauthoritative. This type of restore must still replicate with other domain controllers to update its database information.

You must consider one caveat when you have the potential for data loss. Say you backed up your domain controller, and then the following day you performed updates to some Active Directory objects. That same day you updated, however, an unexpected network interruption or outage caused the replication to fail. If the domain controller to which you made updates experienced an unrecoverable hardware failure, those changes would be lost; note this scenario is shown in Figure 10.4. Although this scenario may be a bit dramatic, it is possible, so you should be aware of this type of situation.



**FIGURE 10.4**

*Potential data loss scenario for nonauthoritative restore.*

Also, if the domain controller begins originating new updates before receiving updates from replication partners, either the domain controller declares itself fully up-to-date with respect to its own changes or not up-to-date. If it considers itself up-to-date, the domain controller never receives updates it originated after the backup; whereas if the domain controller considers itself not up-to-date, the new updates replicate back to the domain controller.

## Verifying the Nonauthoritative Restore

You can preverify the restore success by checking that Active Directory, Certificate Services, and File Replication Services are operational. Then check for files and services that existed before the restore.

### NOTE

You can perform this procedure only immediately after you restore the domain controller and before you start the domain controller in normal mode and bring it online.

To perform advanced verification of Active Directory after a Backup utility restore, perform the following steps:

1. Restart the computer in Directory Services Restore Mode. Press F8, and then from the Advanced Options menu, select the Windows 2000 operating system.
2. Log on to the local administrator's account on the server.
3. Open regedit or regedt32.
4. Locate the RestoreInProgress entry in the HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS subkey.

### NOTE

Because the restored database is not in a valid format for Active Directory, the Backup utility adds RestoreInProgress to the NTDS Registry subkey to make it valid. It then reads this entry during system initialization and subsequently deletes the entry; therefore, you should not make any changes to this key.

5. Close the Registry Editor and open Ntdsutil. Type **Files** and then **Info**. If the Active Directory database files are successfully recovered, Ntdsutil displays that information (see Figure 10.5).
6. Restart the domain controller in normal mode.

```

C:\WINNT\system32\ntdsutil.exe: Files
file maintenance: info

Drive Information:
C:\ NTFS (Fixed Drive ) free(4.7 Gb) total(3.1 Gb)
X:\ NTFS (Network Drive) free(8.3 Gb) total(14.3 Gb)

DS Path Information:
Database : C:\WINNT\NTDS\ntds.dit - 10.1 Mb
Backup dir : C:\WINNT\NTDS\dsadata.bak
Working dir: C:\WINNT\NTDS
Log dir : C:\WINNT\NTDS - 50.0 Mb total
          res2.log - 10.0 Mb
          res1.log - 10.0 Mb
          edb00003.log - 10.0 Mb
          edb00002.log - 10.0 Mb
          edb.log - 10.0 Mb

file maintenance:

```

**FIGURE 10.5**

*The Ntdsutil screen in Directory Services Restore Mode.*

## Restoring Active Directory to Dissimilar Hardware

If you are restoring Active Directory to a different computer, the new computers must have the same number of disk drives and minimum disk space as the former.

## Performing an Authoritative Restore

You perform an authoritative restore after you perform a nonauthoritative restore. After you restore from backup by using the Backup utility, you use the Ntdsutil utility to denote restored objects as authoritative so that they are replicated to the other domain controllers throughout your enterprise.

### NOTE

Only the domain and configuration domain directory partitions can be marked as authoritative. The schema cannot be authoritatively restored because it might endanger data integrity.

Authoritative restore is often used in situations in which objects are inadvertently deleted from Active Directory, and the deletions were propagated to other domain controllers. If this situation occurs, you can make an authoritative restore from a backup created prior to the object deletion.

After the domain controller is restored, but before the domain controller is restarted, you designate the deleted objects as authoritative. This way, when you bring the domain controller online, those objects are replicated to the other domain. Because the authoritatively restored objects have a higher version number, previously deleted objects are ignored during replication.

**NOTE**

Objects you created after the backup was restored—but before replication—are not affected by an authoritative restore.

You should use the authoritative restore feature of Ntdsutil sparingly because it restores the directory to an earlier state, causing the loss of any updates made after the saved state.

## Authoritatively Restoring Active Directory by Using Ntdsutil

After a nonauthoritative restore, you can perform an authoritative restore by using Ntdsutil. To do so, perform the following steps:

1. After restarting the domain controller, press F8 to display advanced startup options.
2. Select Directory Services Restore Mode.
3. At a command prompt, type **ntdsutil** and press Enter.
4. At the ntdsutil prompt, type **authoritative restore** and press Enter.
5. At the authoritative restore prompt, type **restore subtree**, followed by the LDAP name of the object to be authoritatively restored.

For example, to restore the Sales organizational unit (OU) in the kevinkocis.com domain, the commands are as follows:

```
ntdsutil  
authoritative restore  
Restore Subtree OU=Sales,DC=kevinkocis,DC=COM
```

**NOTE**

Ntdsutil opens the Ntds.dit file, increases version numbers, counts the records that need updating, verifies the number of records updated, and reports completion. If you do not specify an increased version number, Ntdsutil does so automatically.

6. In the Authoritative Restore dialog box, click Yes.
7. At the authoritative restore prompt, type **quit** and press Enter.
8. At the ntdsutil prompt, type **quit** and press Enter.
9. Restart the domain controller in normal mode.

Because the objects that are restored have the same objectGUID and objectSID, security remains intact, and object dependencies are maintained.

The preceding process is primarily for authoritative restoration of only Active Directory.

#### NOTE

Certain Active Directory objects such as OUs, domains, and site objects may have group policies as well, which are stored in the SYSVOL directory.

## Authoritatively Restoring the Entire Active Directory Database

When you authoritatively restore the entire Active Directory database, you must ensure that the proper elements are authoritatively restored. You do so by copying the SYSVOL directory from the alternate location over the existing SYSVOL directory. You need to do so to ensure the integrity of the Group Policy of the computer.

To restore the entire Active Directory database, perform the following steps:

1. Back up the System State data by using the Backup utility.
2. Restart the computer in Directory Services Restore Mode (press F8 while restarting).
3. Restore the System State data to its original location and to an alternate location.
4. By using Ntdsutil, mark the entire Active Directory database as authoritative.
5. Restart the computer in normal mode.
6. After the SYSVOL share is published, copy the SYSVOL directory on the alternate location over the existing one. Sysvol is normally secured for system access only if you are attempting to directly access the folder on the local hard drive. If you are accessing it through a share, then you may have the required access without resetting permissions. You can verify that the copy is complete by checking the contents of the Sysvol\domain directory. When the copy is complete, it contains a Scripts and Policies folder.

## Authoritatively Restoring Specific Active Directory Objects

When you authoritatively restore a portion of the Active Directory database, including Policy objects, you also must perform an additional procedure, described next, involving the



SYSVOL directory. To ensure the proper elements are authoritatively restored, use the following process:

1. Back up the System State data by using the Backup utility.
2. Restart the computer in Directory Services Restore Mode.
3. Restore the System State data to its original location and to an alternate location.
4. By using Ntdsutil, separately mark specific Active Directory objects as authoritative.
5. Restart the computer in normal mode.
6. After the SYSVOL share is published, copy only policy folders (identified by the GUID) corresponding to the restored Policy objects from the alternate location over the existing ones.

#### NOTE

Make sure you copy the SYSVOL and policy data from the alternate location after the SYSVOL share is published.

Publishing the SYSVOL share may take several minutes because it needs to synchronize with its replication partners.

When an object is deleted, a tombstone attribute for the object is replicated to all DC's, as discussed in Chapter 8, "Managing Sites, Replication, and Network Traffic." This can lead to problems when attempting to restore active directory objects from a backup. For example, suppose the domain contains three DC's: DC1, DC2, DC3. An administrator accidentally deletes an OU containing 2000 users objects on DC1. The OU will disappear from the AD console and be tombstoned for replication with DC2 and DC3. On the next replication cycle, all three DC's will have the tombstone for the OU. The administrator realizes the mistake and pulls out his backup tape from the previous night. He successfully restores the OU and reboots his DC to complete the restore process. After restarting the DC the administrator opens AD users and computers expecting to see the restored OU but does not. The reason is because of the tombstones on the replication partners. When a DC starts up, it will initiate replication with its configured partners to ensure that it has consistent data in its database. In this case DC2 or DC3 would have replicated the tombstone for the OU back to DC1.

It is for this reason Authoritative Restores exists.

## Verifying the Authoritative Restore

An authoritative restore positions every involved attribute to its current value. Even though the attribute values don't change, their metadata does change to indicate the time and location settings of each attribute's current value.

Authoritative restore metadata causes domain controllers to view that data as an update, which will override current values. To the authoritatively restored domain controller, the current value is the value at the backup time. To all the other domain controllers, the current value is the value following all the changes that are made (after the backup).

You can use the Repadmin utility to verify that the authoritative restore was successful by checking the version number increase. Do so by using the `show metadata` command, followed by the exact distinguished name of the directory or subtree that you authoritatively restored.

## Impact of Authoritative Restore on Trust Relationships and Network Connections

An authoritative restore may affect trust relationships and computer account passwords because they reside in the domain directory partition of Active Directory. This includes both parent and child trust relationships in Windows 2000 domains and Kerberos and NT LAN Manager (NTLM) trust relationships to other Windows NT 4.0 or Windows 2000 domains.

If you authoritatively restore an entire domain directory partition, computer passwords and trust relationship passwords are restored to the values at the time of the backup. If the password or trust was altered following the last backup, trust relationships and computer accounts may be nullified. As a result, trusts no longer work, and a workstation might not be able to connect to a domain controller.

### NOTE

Be very selective when you choose objects to authoritatively restore. You should restore only those portions of the domain directory partition critical to your business needs.

The more data you restore, the greater the opportunity to affect one of these situations.

Use the Active Directory Domains and Trusts snap-in to reset Windows 2000 trust relationship passwords and the Active Directory Users and Computers snap-in to reset the computer account passwords. You can also use the Netdom command-line utility to reset trust relationship and computer account passwords.

## Monitoring Active Directory Performance

Monitoring Active Directory performance is vital to making sure that Active Directory is meeting your business and networking goals. For example, one aspect of ensuring optimal performance is verifying that all network servers are getting directory replication updates and applying them in a timely manner. To monitor replication, as well as other activities, you have available to you Microsoft Management Console (MMC) snap-ins, Microsoft Windows 2000 Server Resource Kit command-line tools, Windows 2000 support tools (from the Server CD), and Visual Basic scripts.

## Monitoring Domain Controller Performance

You can use several utilities to monitor domain controller performance such as system metrics or network throughput at replication metrics. These utilities can assist with Active Directory performance issues. Let's take a closer look at these utilities (which were also available in Windows NT 4.0).

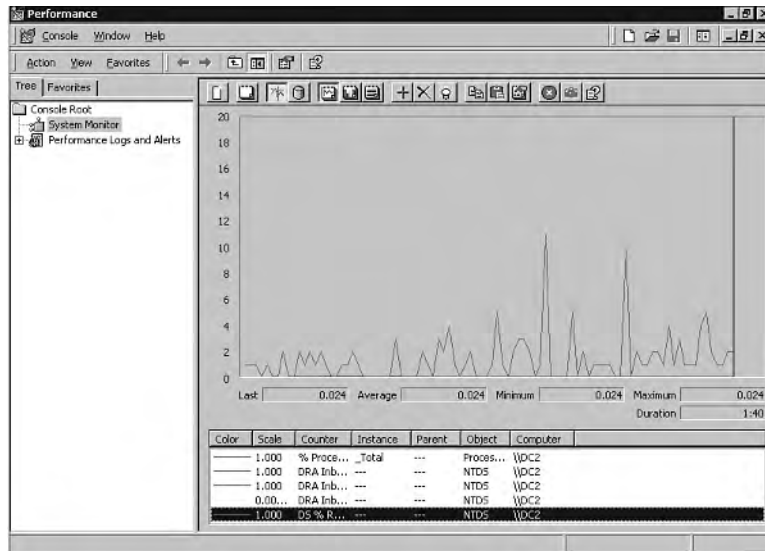
## System Monitor

The System Monitor utility enables you to monitor and collect performance data in real-time charts or reports and stores the data in logs. System Monitor also enables you to generate alerts to warn you when critical events occur. This information is classified as a performance object by the component (which can be a service, computer, or mechanism) creating the data. When you monitor Active Directory, you monitor the activity reported by the Windows NT Directory Services (NTDS) performance object.

To launch System Monitor, click Start, Programs, Administrative Tools, Performance in Administrative Tools (see Figure 10.6).

The first step in monitoring your system performance is to determine what you want or need to monitor. All the performance statistics associated with Windows 2000 can be categorized in three areas:

- **Objects**—Collections of various performance statistics that you can observe in System Monitor. The key object for Active Directory performance is NTDS. You'll learn more about NTDS monitoring in a moment.
- **Counters**—The actual parameters (grouped inside an object) that are measured by System Monitor. An example of a counter is % Processor Time, which tracks one type of information about the Processor Object.
- **Instances**—A further division of the measuring process, actually monitoring instances of a counter. For example, in the case of a multi-processor server (quad processor, for instance), a single instance % Processor Time can be added to create the %Total Utilization counter, which includes any determined processors.



**FIGURE 10.6**  
*The System Monitor console.*

To determine whether a server is receiving and applying directory replication updates effectively, select counters from the NTDS object. The DRA Pending Replication Synchronizations counter from the NTDS object monitors the number of directory synchronizations queued for a server that remain to be processed.

## System Monitor Counters

Only counters relevant to installed system applications, services, and hardware are available in System Monitor. Therefore, when Active Directory is installed, performance counters in the NTDS object are created to provide statistics about directory activity. As with any system monitoring, you need to create a baseline to determine whether performance is actually lacking.

The following sections provide a closer look at some important Active Directory objects.

## NTDS Object

The NTDS object contains performance counters that provide statistics about Active Directory performance. For example, several counters are associated with the Directory Replication Agent (DRA), which monitors replication activity. Due to the numerous performance counters available for DRA, you need to identify the critical objects you need to monitor first.

The NTDS object counters that assist with monitoring Active Directory fall into several categories (alphabetically as follows):

- Address Book (AB)
- Directory Replication Agent (DRA)
- Directory Service (DS)
- Key Distribution Center (KDC)
- Lightweight Directory Access Protocol (LDAP)
- NTLM Authentications (from down-level clients)
- Security Accounts Manager (SAM)
- Extended Directory Services (XDS)

The portions of Active Directory you are looking to monitor will determine the counters you will select. The scope of each counter for each object is beyond the scope of this chapter (and book, for that matter). The best way to learn about the various counters is to add them to the System Monitor and view them graphically as a chart or save it in a log file.

Let's look at adding counters to the NTDS object.

### Adding NTDS Counters

To monitor Active Directory performance counters, perform the following steps:

1. Open Performance in Administrative Tools.
2. Right-click the details pane, and select Add Counters or click the + in the menu bar over the display pane.
3. In the Select Counters from Computer field, make sure the name of the local computer is displayed (it is the default).

#### NOTE

Local monitoring may affect the results and the performance of the machine whose performance you are measuring, but will conserve network bandwidth. It is recommended that monitoring of a machine take place from a remote workstation when network bandwidth is not an issue.

4. In the Performance Object field, select NTDS.

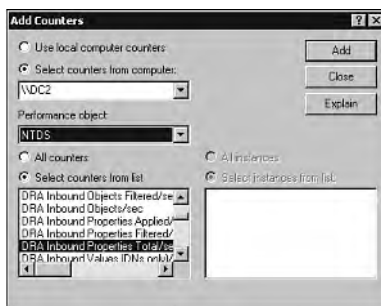
**NOTE**

To monitor objects other than NTDS, select them from the Performance Object list. NTDS is a commonly monitored object for Active Directory performance, so it is selected for this example.

5. In the Select Counters from List field, select the counters you want to use for monitoring (see Figure 10.7). You may select multiple counters by holding down the shift or control keys while selecting with your mouse pointer.
6. Click Add and then Close after you have added all the necessary counters.

**CAUTION**

The more counters you add to System Monitor, the more the domain controller's performance is affected. Therefore, selecting the All Counters radio button greatly affects server performance and is not a practical solution.



**FIGURE 10.7**

*Adding counters to System Monitor.*

To delete counters from System Monitor, perform the following steps:

1. Open Performance in Administrative Tools (or restore/maximize your System Monitor console).
2. In the System Monitor details pane, click to highlight the counter name in the legend.
3. Press Delete.

If you need to get more detailed information about the various counters, perform the following steps:

1. Open Performance.
2. Right-click the System Monitor details pane and click Add Counters (or the + icon at the top of the pane).
3. In the Performance Object field, select an object.
4. In the Select Counters from List field, select a counter.
5. Click Explain.

You can now select any object and counter to get more information. Click Add or Close to exit this option.

### Manually Loading and Unloading NTDS Counters

You may encounter a situation in which the NTDS object fails to load and is not available when you open System Monitor. In this case, you need to load the NTDS object manually. To load the counter information into the Registry, type **lodctr.exe ntdsctrs.ini** at the command prompt, and then restart the computer.

After you complete these steps, you can use System Monitor to view the counters for the NTDS object.

To unload the NTDS object manually from the Registry, type **unlodctr.exe ntds** at the command prompt.

### Database Object

The Database object relates to the Extensible Storage Engine (ESENT), the transacted database system that stores all Active Directory objects. This performance object is not installed by default, and you cannot automatically install the performance dynamic link library (DLL), *Esentprf.dll*, in Windows 2000.

To load the Database object, perform the following steps:

1. Copy the performance DLL (*Esentprf.dll*) located in <SystemRoot>\System32 to another directory.
2. Run *Regedt32.exe* or *Regedit.exe*, and make sure that the following Registry subkeys exist:

HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\ESENT

HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\ESENT\Performance

If they do not exist, you need to create them.

3. Make sure that, under the Performance subkey, Registry values with the following settings exist:  
data type REG\_SZ : OpenPerformanceData  
data type REG\_SZ : CollectPerformanceData  
data type REG\_SZ : ClosePerformanceData  
data type REG\_SZ : c:\perf\esentprf.dll
4. Change directory to %SystemRoot%\Winnt\System32 or to another folder that contains the files Esentperf.ini and Esentperf.hxx, which were generated when Eseperfmt.dll was compiled.
5. If you want to verify that previous counter information is not present in the Registry, type **unlodctr.exe ESENT** at the command prompt.
6. To load the counter information into the Registry, run **Lodctr.exe Esentperf.ini**.

To view the counters for the Database object, restart System Monitor.

## Using System Monitor to Select Performance Counters

Similar to the System Monitor utility, the Performance utility enables you to select performance counters in a performance object, such as NTDS, for purposes of tracking certain types of activity. Furthermore, you can select the time intervals during which to log the activity, and you can print the logs or view them online by using System Monitor.

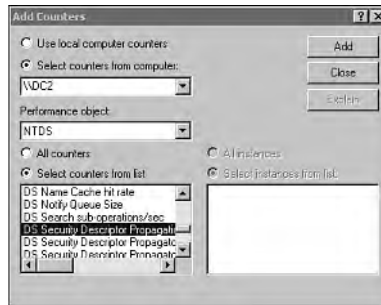
You can select the performance counters that you want to monitor in System Monitor. You can view a graphical representation of the counters by selecting a type of display: chart, histogram, or log file data.

To select Active Directory performance counters to monitor, perform the following steps:

1. Open Performance from Administrative Tools.
2. Click the add counter button (+) to display the Add Counters window.
3. Select the name of the computer that you want to monitor: either the local computer or another domain controller.
4. Select the NTDS or Database performance object.
5. Select the performance counters that you want to add and then click Add.
6. Click Close.

If you need a detailed explanation of the counter, click Explain. The Explain button allows you to view all the details about a selected counter, and the description remains open until you click Add or Close (see Figure 10.8).



**FIGURE 10.8**

*The Explain feature in the Performance utility.*

The counters that you selected appear in the lower part of the screen. System Monitor displays each counter in a unique color.

When you're creating a monitoring console for export, make sure to select Use Local Computer Counters. Otherwise, System Monitor obtains data from the computer named in the text box, regardless of the place where the console file is installed.

## NOTE

When creating a monitoring console for export, you may choose local computer counters or counters from a remote system you wish to monitor. If you choose a remote machine to monitor, System Monitor obtains data from the computer named in the text box, regardless of where the console file is installed.

Choosing to remotely monitor may be preferred depending on your available network bandwidth and whether you keep the console as part of your roaming profile.

## Performance Logs and Alerts

The Performance Logs and Alerts tool enables you to log performance counters and system alerts. You can optionally collect data using the Windows Management Instrumentation (WMI) interface for hardware resources installed on the system. WMI can trace data available in Active Directory for core directory services such as the Lightweight Directory Access Protocol (LDAP), Key Distribution Center (KDC), Security Accounts Manager (SAM), Local Security Authority (LSA), and Net Logon. When enabled, trace logging continuously captures key events such as network logons, authentications, LDAP operations, and SAM operations, and it also records the CPU time, timestamp, and thread identifier. You can enable or disable trace logging by using the Performance Logs and Alerts utility. To produce transaction-level costing information trace data, you must use the trace application programming interfaces (APIs).

## Counter Logs

Counter logs track performance of objects, counters, and instances in System Monitor based on time intervals. These statistics are saved to a file for you to analyze later.

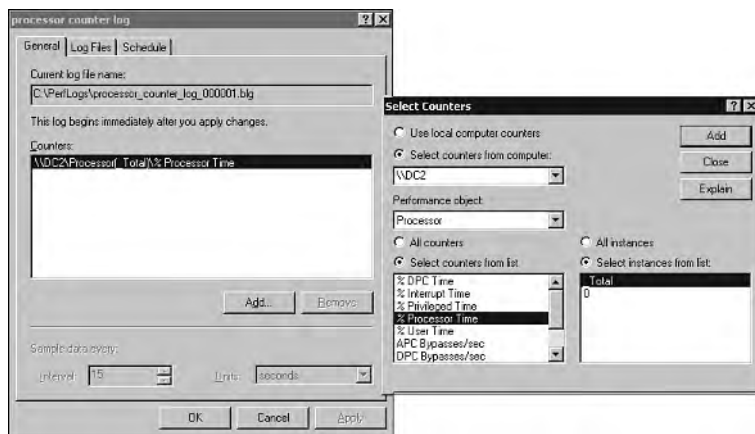
To create a counter log, perform the following steps:

1. Open Performance in Administrative Tools.
2. Double-click Performance Logs and Alerts and then click Counter Logs. If you have previously existing logs, they appear in the details pane.

### NOTE

A green icon indicates that the log is running. A red icon on the log indicates that it has been stopped.

3. Right-click in the details pane and click New Log Settings.
4. In the Name box, input the log name and click OK.
5. On the General tab, click Add. Then select the performance object and the associated counters you want to log as shown in Figure 10.9.
6. If you want to change the default file and schedule information, make the changes on the Log Files tab and the Schedule tab.



**FIGURE 10.9**

*Creating a counter log to monitor processor utilization.*

To create or modify a log, you must have Full Control permission for the following Registry key, which controls the Performance Logs and Alerts service:

HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\SysmonLog\Log Queries

Administrators usually have this permission by default. Administrators can grant permission to users by using the Security menu in Regedt32.exe.

To run the service (which runs in the background when you configure a log), you must have permission to start or otherwise configure services on the system. Administrators have this right by default and can grant it to users by using Group Policy.

To add counters to a log, perform the following steps:

1. Open Performance in Administrative Tools.
2. Double-click Performance Logs and Alerts and then click Counter Logs.
3. In the details pane, double-click the name of the log you want to modify.
4. On the General tab, click Add.
5. In the Performance Object field, select the desired object.
6. In the Select Counters from List field, select the counters you want to use for monitoring.
7. Click Add and then Close after you have added all the necessary counters.

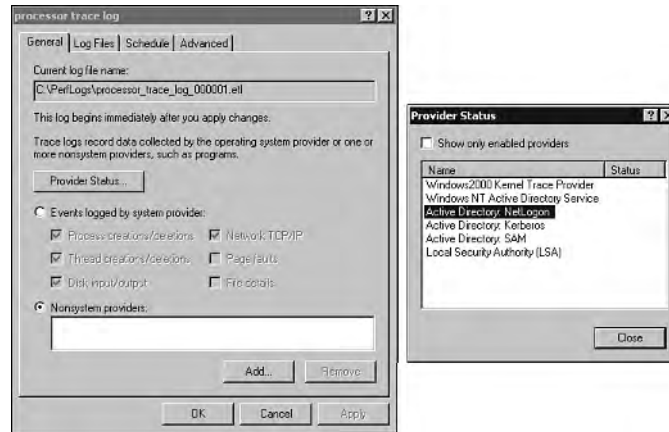
## Trace Logs

Trace logs track performance of objects, counters, and instances in System Monitor based on system events. Unlike counter logs, these logs are based on specific events as opposed to times; therefore, these logs may not be effective if you are troubleshooting sporadic issues.

To create a trace log, perform the following steps:

1. Open Performance.
2. Double-click Performance Logs and Alerts and then click Trace Logs.
3. Right-click in the details pane or on the trace log name in the console pane; then click New Log Settings.
4. In the Name box, input the trace log name and click OK.

The trace log file is created in the PerfLogs folder in your root directory. A sequence number and the .etl extension are attached to the file. Use the Log Files and Advanced tabs to modify these parameters or define other parameters for your log. To define providers and events to log, use the General tab (see Figure 10.10). To specify when you want logging to occur, use the Schedule tab.



**FIGURE 10.10**

*Creating a trace log to monitor processor utilization.*

To define trace log providers and events, perform the following steps:

1. Open Performance In Administrative Tools.
2. Double-click Performance Logs and Alerts and then click Trace Logs.
3. In the details pane, double-click the name of the log.
4. Click Provider Status for a list of the installed providers and their status (enabled or not). The Nonsystem Providers option is selected by default to minimize trace-logging overhead.
5. If you click Events Logged by System Provider, a default provider (the Windows kernel trace provider) will monitor processes, threads, and other activity. To define events for logging, click the check boxes as appropriate.
6. If you click Nonsystem Providers, you can select the data providers you want—for example, if you have written your own providers. Use the Add or Remove buttons as needed.

To define trace log buffers, perform the following steps:

1. Open Performance.
2. Double-click Performance Logs and Alerts and click Trace Logs.
3. In the details pane, double-click the name of the log.
4. Click the Advanced tab.
5. In the Buffer Size box, specify the size of buffer in kilobytes you want to use for trace data.

6. In the Minimum box, specify the smallest number of buffers you want used for trace data.
7. In the Maximum box, specify the largest number of buffers you want used for trace data.
8. To have the trace provider periodically flush the buffers, select the Transfer Data from Buffers to Log File Every check box and specify the transfer interval in seconds.

## Alerts

Alerts allow you to define a counter value that is deemed unacceptable based on your established baseline values. Alerts can trigger actions such as sending a network message, running a program, or starting a log. Alerts are useful when you want to be notified if a particular counter threshold value exceeds or falls below a specified value. You can then take action to resolve the variance.

To create an alert, perform the following steps:

1. Open Performance.
2. Double-click Performance Logs and Alerts and then click Alerts.
3. Right-click in the details pane and click New Alert Settings.
4. In the Name box, input the alert name and click OK.
5. To define a comment for your alert, along with counters, alert thresholds, and the sample interval, use the General tab (see Figure 10.11). To define actions that should occur when counter data triggers an alert, use the Action tab, and to define when the service should begin scanning for alerts, use the Schedule tab.



**FIGURE 10.11**

*Creating an alert when processor utilization exceeds 50% over a two-hour period.*

To define counters and thresholds for an alert, perform the following steps:

1. Open Performance.
2. Double-click Performance Logs and Alerts and then click Alerts.
3. In the details pane, double-click the name of the alert.
4. In the Comment box, type a comment to describe the alert as needed.
5. Click Add.
6. For each counter or group of counters that you want to add to the log, add the performance objects, counters, or instances as described previously. Then click Add.
7. In the Alert When the Value Is box, specify Under or Over, and in the Limit box, specify the value that triggers the alert.
8. In the Sample Data Every box, specify the amount and the unit of measure for the update interval.
9. Complete the alert configuration using the Action and Schedule tabs.

To define actions for an alert, perform the following steps:

1. Open Performance in Administrative Tools.
2. Double-click Performance Logs and Alerts and then click Alerts.
3. In the details pane, double-click the name of the alert.
4. Click the Action tab.
5. To have the Performance Logs and Alerts service create an entry visible in Event Viewer, select Log an Entry in the Application Event Log.
6. To have the service trigger the messenger service to send a message, select Send a Network Message To and type the name of the computer on which the alert message should be displayed.
7. To run a counter log when an alert occurs, select Start Performance Data Log and specify the counter log you want to run.
8. To have a program run when an alert occurs, select Run This Program and type the file path and name or click Browse to locate the file. When an alert occurs, the service creates a process and runs the specified command file. The service also copies any command-line arguments you define to the command line that is used to run the file. Click Command Line Arguments and select the appropriate check boxes for arguments to include when the program is run.

To remove counters from a log or alert, perform the following steps:

1. Open Performance in Administrative Tools.
2. Double-click Performance Logs and Alerts and then click Counter Logs or Alerts.
3. In the details pane, double-click the name of the log or alert.
4. Under Counters, click the counter you want to remove and then click Remove.

To define start or stop parameters for a log or alert, perform the following steps:

1. Open Performance in Administrative Tools.
2. Double-click Performance Logs and Alerts and then click Counter Logs, Trace Logs, or Alerts.
3. In the details pane, double-click the name of the log or alert.
4. Click the Schedule tab.
5. Under Start Log, select whether you want to run the log manually or automatically and set the parameters.
6. Under Stop Log, select the parameters for stopping the alert (options include manually, after a specified duration, at a certain time and date, or when the log becomes full).

#### NOTE

Consider available disk space and any disk quotas because you may encounter errors (such as STOP screen errors) if your disk runs out of disk space because of logging.

7. Complete the properties as appropriate for logs or alerts (such as circular, or continuous logging).

## Task Manager

Task Manager provides information about applications currently running on your system, the processes and memory usage or other data about those processes, and statistics about memory and processor performance.

Although useful as a quick reference to system operation and performance, Task Manager does not possess the logging and alert capabilities of the Performance console. Task Manager also does not have access to the scope of information available from all installed counters.

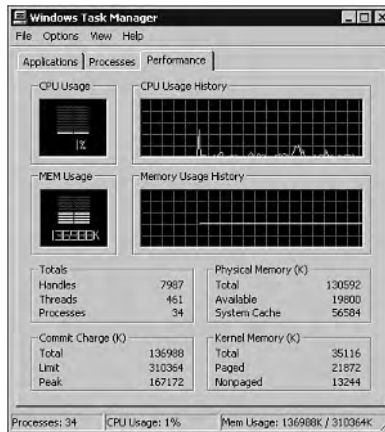
However, Task Manager provides capabilities not available with the Performance console, such as the capability to stop processes that are currently running on the system, change the base priority of a process, and set affinity (on multiprocessor systems) for a process to a particular processor.

To start Task Manager, use any of these methods:

- Press Ctrl+Shift+Esc.
- Right-click the taskbar and then click Task Manager.
- Press Ctrl+Alt+Del and then click Task Manager.
- Click Start, Run and type **taskmgr**.

The main Task Manager display has three tabs:

- Applications—Lists all the programs running on your system, along with the status of each. You can use the information in this window to find out whether an application is in trouble because it is described in this window as Not Responding instead of bearing the normal description of Running.
- Processes—Lists all the processes running on your system by name, process ID (PID) number, processor times, and memory usage.
- Performance—Displays constantly updated totals for current and historic processor and memory usage (see Figure 10.12).



**FIGURE 10.12**

*The Performance tab in Task Manager.*

Task Manager also includes physical memory allocation as well as information on handles, processes, and threads running on your server.



## Event Logs

Using the event logs in Event Viewer, you can gather information about hardware, software, and system problems, and you can monitor Windows 2000 security events.

Windows 2000 provides the Event Viewer snap-in as a way to monitor system events, such as application or system errors and the verification of running services. These events are recorded in event logs. For example, if you need detailed information about the times directory partitions are replicated, you would use Event Viewer to study the event log.

Also, if you suspect any problem with the directory operation, such as information not being replicated, it is recommended that you first investigate the event logs to determine the cause of the problem. By using information from the event logs, you can better understand the sequence and types of events that led to the performance problem.

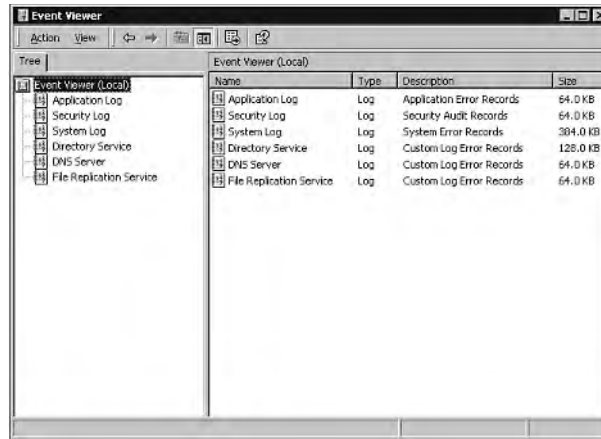
Just like in Windows NT 4.0, Windows 2000 records events in three types of logs:

- **Application**—Contains events logged by applications or programs. For example, a Web program would record a file error in the application log. The program developer decides which events to record.
- **System**—Contains events logged by Windows 2000. For example, a driver issue would be recorded in the system log. The event types logged by system components are predetermined by Windows 2000.
- **Security**—Records security events such as valid and invalid logon attempts as well as events related to resource use such as creating, opening, or deleting files. By enabling auditing, you can specify which events are recorded in the security log.

Event Viewer (as shown in Figure 10.13) displays the following types of events:

- Error
- Warning
- Information
- Success Audit
- Failure Audit

The EventLog service starts automatically when you start Windows 2000. All users can view application and system logs. Only administrators can gain access to security logs.

**FIGURE 10.13**

*The Event Viewer utility.*

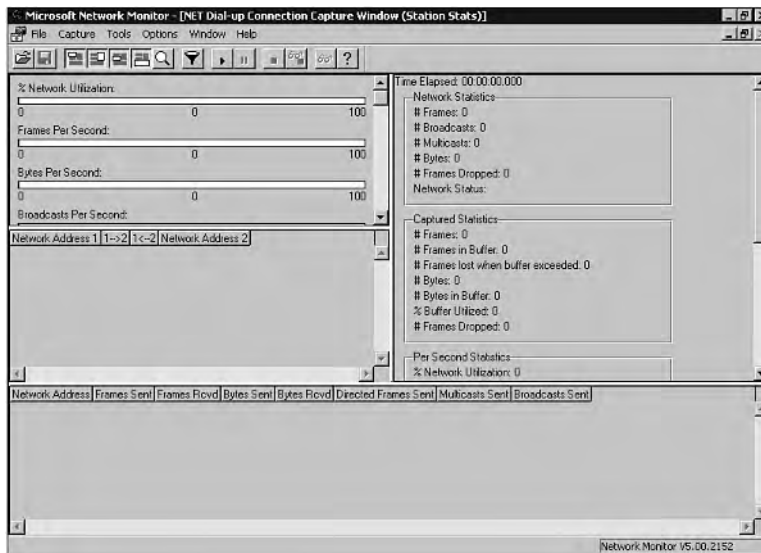
You can open the Event Viewer by double-clicking the Event Viewer icon in Administrative Tools.

By default, security logging is turned off. You can use Group Policy to enable security logging (see Chapter 6, “Administering Group Policy,” for more details). The administrator can also set auditing policies in the Registry that cause the system to halt when the security log is full.

## Network Monitor

You use Network Monitor (see Figure 10.14) to capture and display the frames (also called packets) running between Windows 2000 computers on your local area network. You also can use Network Monitor to detect and troubleshoot networking problems. The frames you capture in Network Monitor can be saved to a file for further analysis.

Microsoft Systems Management Server (SMS) includes a full version of Network Monitor. In addition to the functionality in Windows 2000 Network Monitor, Systems Management Server Network Monitor can capture frames sent to and from all computers in a network segment, as well as edit and transmit frames.

**FIGURE 10.14**

*The Network Monitor interface.*

If you did not install Network Monitor during the Windows 2000 setup, you can install it by performing the following steps:

1. Open Add/Remove Programs from Control Panel.
2. In Add/Remove Programs, click Add/Remove Windows Components.
3. In the Windows Components Wizard, select Management and Monitoring Tools and then click Details.
4. In the Management and Monitoring Tools window, select the Network Monitor check box and then click OK.
5. If you are prompted for additional files, insert your Windows 2000 Server CD or select the appropriate network path to the files.

You can install the Network Monitor driver only on Windows 2000 computers. Network Monitor drivers for operating systems other than Windows 2000 are available in Microsoft Systems Management Server.

To capture network frames, perform the following steps:

1. Open Network Monitor in Administrative Tools.
2. If you are prompted for a default network on which to capture frames, select the local network from which you want to capture data by default.

3. On the Capture menu, click Start.
4. If you are prompted for additional files, insert your Windows 2000 Server CD or select the appropriate network path to the files.

You can protect your network from unauthorized monitoring because Network Monitor can detect other installations of itself running on your local network segment. When Network Monitor detects other Network Monitor installations running on the network, it displays the following information about them:

- The name of the computer
- The name of the user logged on at the computer
- The state of Network Monitor on the remote computer (running, capturing, or transmitting)
- The adapter address of the remote computer
- The version number of Network Monitor on the remote computer

The full version of Network Monitor (included with Microsoft Systems Management Server) supports additional protocol parsers.

You can use a display filter to determine which frames to display. You can filter a frame by its source or destination address, the protocols used to send it, and the properties and values it contains.

Network Monitor simplifies data analysis by interpreting raw data collected during the capture and displaying it in the Frame Viewer window. To display captured information in the Frame Viewer window, from the Capture menu, click Stop and View while the capture is running. Or open a capture file (.cap).

#### NOTE

Although the Windows 2000 Network Monitor may assist to some degree with network monitoring, using the version that comes with Microsoft Systems Management Server is recommended if you are seeking high-level monitoring.

## Summary

This chapter described the importance of maintaining Active Directory reliability with the Backup utility and Ntfsutil for backups and restores. You looked at some of the caveats of using the Windows 2000 built-in Backup utility and the need to consider third-party utilities for enhanced features. You also looked at monitoring Active Directory performance with MMC snap-ins and the Performance tool, which includes System Monitor and the logs and alert files.

# Common Active Directory Utilities

## APPENDIX

# A

## IN THIS APPENDIX

- **ADSI (Active Directory Service Interface) 314**
- **Comma-Separated Value Directory Exchange (CSVDE) 314**
- **LDAP Data Interchange Format Directory Exchange (LDIFDE) 316**
- **Movetree 319**

## ADSI (Active Directory Service Interface)

**Location:** Support\Tools folder on the Windows 2000 Server operating system CD.

**Function:** Provides applications and scripts with a protocol-independent interface to a directory service.

**Description:** ADSI uses the Component Object Model (COM), which allows programs and scripts to access ADSI to modify LDAP directories such as Active Directory, Windows NT 4.0 directories, NetWare NDS directories, and NetWare 3 binderies. ADSI provides access to multiple directory services via an open set of interfaces and allows developers to create applications and scripts using Visual Basic, Java, and C/C++. ADSI interfaces allow Windows 9x, NT, and 2000 to access a directory service.

### NOTE

ADSI must be installed from the Windows 2000 Server CD before it can be added to an MMC console.

The following example uses a Visual Basic script to show all objects in a given organizational unit:

```
Set ou = GetObject("LDAP://dc1/OU=Sales, DC=kevinkocis,DC=COM")
For each obj in ou
    Debug.Print obj.Name
Next
```

## Comma-Separated Value Directory Exchange (CSVDE)

**Location:** Installed with Windows 2000 Server operating system.

**Function:** A command-line utility that allows objects stored in Microsoft's comma-separated value (CSV) file format to be imported or exported to and from Active Directory.

**Description:** Although CSVDE does not allow you to modify or delete directory objects, you can create and manage data by importing and exporting it as a CSV file between applications. CSV is a fairly restrictive format, used primarily by Microsoft applications including Microsoft Excel and Exchange. For a more standards-based directory exchange solution, use LDIFDE (see next section). The CSVDE parameters are listed in Table A.1.

**TABLE A.1** CSVDE Parameters

<i>Parameter and Value</i>	<i>Description</i>
-i	Specifies import mode. The default mode is export.
-f <i>filename</i>	Identifies the import or export filename.
-s <i>server name</i>	Specifies the domain controller to perform the operation.
-c <i>string1 string2</i>	Replaces all occurrences of <i>string1</i> with <i>string2</i> . Replaces the distinguished name of the export domain with that of the import domain when you're importing data from one domain to another.
-v	Sets verbose mode.
-j <i>path</i>	Sets the log file location. The default is the current path.
-t <i>port number</i>	Specifies an LDAP port number. The default is port 389; the global catalog port is 3268.
-d <i>baseDN</i>	Sets the distinguished name of the search base for data export.
-r <i>LDAP-filter</i>	Creates an LDAP search filter for data export.
-p <i>scope</i>	Sets the search scope; may be one of Base, OneLevel, or SubTree.
-l <i>LDAP-attribute-list</i>	Sets the list of attributes to return in the results of an export query. If this parameter is omitted, all attributes are returned. For example, to retrieve only the distinguished name, common name, first name, surname, and telephone number of the returned objects, you would specify the following attribute list:  -l "distinguishedName, cn, givenName, sn, telephone"
-o	Sets the list of attributes to be omitted from the results of an export query. You typically use this parameter when you're exporting objects from the Active Directory and then importing them into another LDAP-compliant directory that does not support all of Active Directory's attributes.

**TABLE A.1** Continued

<i>Parameter and Value</i>	<i>Description</i>
-g	Disallows paged searches.
-m	Omits attributes that apply only to Active Directory objects such as the <code>ObjectGUID</code> , <code>objectSID</code> , <code>pwdLastSet</code> , and <code>samAccountType</code> attributes.
-n	Disallows binary value export.
-k	Skips errors during the import operation and continues processing.
-a <i>user-distinguished-name password</i>	Sets the command to run using the supplied user-distinguished name and password.
-b <i>username domain password</i>	Sets the command to run as <i>username domain password</i> . The default is the current user.
-?	Displays online help.
-u	Supports Unicode.

**NOTE**

CSVDE can be run on a Windows 2000 server or copied to a Windows 2000 Professional workstation.

# LDAP Data Interchange Format Directory Exchange (LDIFDE)

**Location:** Installed with Windows 2000 Server operating system.

**Function:** A command-line utility that allows you to create, modify, and delete Active Directory objects. It can also be used to extend the schema and populate Active Directory by importing from other LDAP-compliant directory services.

**Description:** LDIFDE uses the LDAP Data Interchange Format (LDIF) to modify Active Directory objects. LDIFDE is similar in functionality to CSVDE, but utilizes a more Internet standards-based format (LDAP). LDIF files are composed of a series of records containing entry modifications which are divided by line separations. You can use a text editor to modify the file if you need to make changes prior to importing or exporting. The basic parameters for LDIFDE, LDIFDE export-specific parameters, and LDIFDE import-specific and credential parameters are shown in Tables A.2, A.3, and A.4.



**TABLE A.2** LDIFDE Basic Parameters

<i>Parameter and Value</i>	<i>Description</i>
-i	Specifies import mode. The default mode is export.
-f <i>filename</i>	Identifies the import or export filename.
-s <i>server name</i>	Specifies the domain controller to perform the operation.
-c <i>string1 string2</i>	Replaces all occurrences of <i>string1</i> with <i>string2</i> . Replaces the distinguished name of the export domain with that of the import domain when you're importing data from one domain to another.
-v	Sets verbose mode.
-t <i>port number</i>	Specifies an LDAP port number. The default is port 389; the global catalog port is 3268.
-?	Displays online help.

**TABLE A.3** LDIFDE Export-Specific Parameters

<i>Parameter</i>	<i>Description</i>
-d <i>baseDN</i>	Sets the distinguished name of the search base for data export.
-r <i>LDAP-filter</i>	Creates an LDAP search filter for data export.
-p <i>scope</i>	Sets the search scope; may be one of Base, OneLevel, or SubTree.
-l <i>LDAP-attribute-list</i>	Sets the list of attributes to return in the results of an export query. If this parameter is omitted, all attributes are returned. For example, to retrieve only the distinguished name, common name, first name, surname, and telephone number of the returned objects, you would specify the following attribute list: -l "distinguishedName, cn, givenName, sn, telephone"
-o	Sets the list of attributes to be omitted from the results of an export query. You typically use this parameter when you're exporting objects from the Active Directory and then importing them into another LDAP-compliant directory that does not support all of Active Directory's attributes.
-g	Disallows paged searches.

**TABLE A.3** Continued

<i>Parameter</i>	<i>Description</i>
-j <i>path</i>	Sets the log file location. The default is the current path.
-m	Omits attributes that apply only to Active Directory objects such as the ObjectGUID, objectSID, pwdLastSet, and samAccountType attributes.
-n	Disallows binary value export.
-u	Supports Unicode.
-y	Enables lazy commit to directory; it stores the change in cache and commits to the directory as a background process.
-?	Displays online help.

**TABLE A.4** LDIFDE Import-Specific and Credential Parameters

<i>Parameter</i>	<i>Description</i>
-k	Skips errors during the import operation and continues processing.
-a <i>user-distinguished-name password</i>	Sets the command to run using the supplied user-distinguished name and password.
-b <i>username domain password</i>	Sets the command to run as <i>username domain password</i> . The default is the current user.

Here's an example of an LDIF import file:

```
dn: CN=lilySmith, CN=Users, DC=na, DC=kevinkocis, DC=com
changetype: add
cn: lilySmith
description: importing user Lily Smith
objectClass: user
sAMAccountName: lilySmith
```

The command to import this file from the current directory is as follows:

```
Ldifde -I -f import.ldf -v
```

## Movetree

**Location:** Windows 2000 Server Resource Kit.

**Function:** A command-line utility that allows you to move users, groups, and organizational units from one domain to another in the same forest.

### NOTE

The source domain can be mixed or native, but the target domain must be a Windows 2000 native domain.

## Movetree Syntax

The syntax used with the Movetree command tool is shown here. Table A.5 further explains the parameters in the syntax line.

```
Movetree [/start | /continue] [/s SrcDSA] [/d DstDSA] [/sdn SrcDN] [/ddn DstDN]  
➤[/u Domain\Username] [/p Password] [/quiet]
```

**TABLE A.5** Movetree Syntax Parameters

<i>Parameter</i>	<i>Description</i>
<i>/start</i>	Starts the Movetree operation. <i>/check</i> is the default. <i>/nocheck</i> can be alternately used.
<i>/continue</i>	Continues a failed operation.
<i>/s SrcDSA</i>	Sets the source server's FQDN. Enter <b>FQDN</b> as <i>SrcDSA</i> .
<i>/d DstDSA</i>	Sets the destination server's FQDN. Enter <b>FQDN</b> as <i>DstDSA</i> .
<i>/sdn SrcDN</i>	Sets the source subtree's root DN.
<i>/ddn DstDN</i>	Sets the destination subtree's root DN.
<i>/u Domain\UserName</i>	Sets the domain name and user account name.
<i>/quiet</i>	Sets quiet mode to run without any screen output.



# INDEX

## SYMBOLS

- \ (backslash), 42**
- / (forward slash), 42**
- ? parameter**
  - CSVDE (Comma-Separated Value Directory Exchange), 316
  - LDIFDE (LDAP Data Interchange Format Directory Exchange), 317-318
- 88 classes, 211**

## A

- a parameter**
  - CSVDE (Comma-Separated Value Directory Exchange), 316
  - LDIFDE (LDAP Data Interchange Format Directory Exchange), 318
- A (address) resource records, 73**
- abstract classes, 211**
- access control, 168-169**
- Access Control Entries (ACEs), 109, 156**
- Access Control Lists. See ACLs**
- access tokens, 108, 156**
- account policies (special policies), 194**
- accounts**
  - computer
    - adding to groups, 134*
    - creating, 133*
    - deleting, 134*
    - disabling, 136*
    - DNS (Domain Name System) names, 137-138*
    - editing, 135*

- enabling*, 137
  - finding*, 135
  - moving*, 136
  - properties*, 135
  - resetting*, 136
- group
  - adding members to*, 131
  - converting group type*, 129
  - creating*, 128
  - deleting*, 129
  - distribution groups*, 120
  - editing properties*, 130
  - effect of domain mode*, 127
  - finding*, 130
  - nesting*, 127-128
  - removing members from*, 131-132
  - renaming*, 132
  - replication conflicts*, 132-133
  - scope and replication traffic*, 126-130
  - security groups*, 120-126
- user
  - copying*, 112
  - creating*, 111
  - deleting*, 113
  - disabling*, 114
  - editing properties*, 113
  - enabling*, 114
  - finding*, 116
  - group memberships*, 115
  - moving*, 116
  - passwords*, 114-115
  - predefined accounts*, 110
  - renaming*, 113-114
  - UPN (user principal name) suffixes*, 115
- ACEs (Access Control Entries)**, 109, 156
- ACLs (Access Control Lists)**, 109, 155-156
  - ACEs (Access Control Entries), 156
  - DACLs (Discretionary Access Control Lists), 109
  - SACLs (System Access Control Lists), 109
  - SIDs (security identifiers), 156-157
- Action menu commands, New Delegation**, 82
- actions**
  - alerts, 306
  - IPSec filter actions
    - creating*, 152-153
    - editing*, 153
    - Permit*, 152
    - Request Security*, 152
    - Require Security*, 152
- activating**
  - attributes, 223-224
  - classes, 223-224
  - computer accounts, 137
  - DHCP (Dynamic Host Configuration Protocol), 98
  - objects, 223
  - schema modification, 214
  - user accounts, 114
- Active Directory (overview)**, 4
  - compared to Novell 5, 19
  - catalogs*, 21-22
  - Internet standards support*, 22
  - partitions*, 20-21
- domains, 16
- forests, 16
- interoperability, 12
  - ADC (Active Directory Connectors), 14
  - DEA (directory-enabled application) platform, 14
  - DEN (directory-enabled networking) platform, 14
  - DNS (Domain Name System) standards, 12
  - extensible schema*, 13
  - MSDSS (Microsoft Directory Synchronization Services), 13
  - native LDAP (Lightweight Directory Access Protocol)*, 12-13
  - open APIs (application programming interfaces)*, 14
- manageability, 4
  - ADSI (Active Directory Services Interfaces), 9
  - backward compatibility*, 9-10
  - centralized management*, 5-6
  - FSMOs (flexible single-master operations)*, 9
  - GC (Global Catalog)*, 7-8
  - group policy*, 7
  - multi-master replication*, 10

- software*
  - distribution*, 7
  - transitive trusts*, 8
- namespaces, 14-15
- OUs (organizational units), 17
- sites, 17
- trees, 15-16
- Active Directory Connectors (ADC), 314**
- Active Directory Domains and Trusts snap-in**
  - changing domain
    - modes, 63
  - creating trusts, 66-67
  - revoking trusts, 68
  - verifying trusts, 68
- Active Directory schema, 204-206**
  - Active Directory Schema snap-in, 207-209
  - attributes, 209
    - adding*, 217-218
    - deactivating*, 224
    - defined*, 204
    - indexing*, 209
    - mandatory*, 210, 217-218
    - modifying*, 218-219
    - reactivating*, 223-224
    - syntax*, 210
  - classes
    - 88 classes*, 211
    - abstract*, 211
    - adding*, 219-220
    - auxiliary*, 211
    - classSchema object*, 210-211
    - deactivating*, 224
    - defined*, 204
    - modifying*, 220-221
    - reactivating*, 223-224
    - structural*, 211
- DIT (Directory Information Tree), 206
- extending, 213
  - FSMO role*, 217
  - LDIF (LDAP Data Interchange Format)*, 216
  - methods*, 213-214
  - OIDs (object identifiers)*, 215
  - order of processing*, 216
  - planning process*, 212-213
  - potential problems*, 222
  - Schema Admins membership*, 215-216
- modifying, 212
  - consistency checks*, 222
  - enabling modification*, 214
  - safety checks*, 222
- objects, 223
- syntax, 204
- verifying modifications to, 221-222
- Active Directory Schema snap-in, 207-209**
- Active Directory Services Interface (ASDI), 9, 32, 314**
- Active Directory Sites and Services snap-in, 228**
  - delegating control, 230
  - designating bridgehead servers, 237
  - site link bridges, 239
- site links
  - creating*, 233-234
  - deleting*, 235
  - turning off*, 238
- sites
  - creating*, 229
  - deleting*, 230
  - renaming*, 228
- subnets, 231
- Active Directory Users and Computers administration. See administration**
- ADC (Active Directory Connectors), 14**
- Add a Script dialog box, 189**
- Add Filter Wizard, 150-151**
- address (A) resource records, 73**
- address pools (DHCP), 104**
- administration. See also security**
  - administrative templates (Group Policy), 183-184
  - computer accounts
    - adding to groups*, 134
    - creating*, 133
    - deleting*, 134
    - disabling*, 136
    - DNS (Domain Name System) names*, 137-138
    - editing*, 135
    - enabling*, 137
    - finding*, 135
    - moving*, 136
    - properties*, 135
    - resetting*, 136
  - delegated, 12

- domains, 61-62
- Group Policy administrative requirements, 173-174
- group accounts
  - adding members to, 131
  - converting group type, 129
  - creating, 128
  - deleting, 129
  - distribution groups, 120
  - editing properties, 130
  - effect of domain mode, 127
  - finding, 130
  - nesting, 127-128
  - removing members from, 131-132
  - renaming, 132
  - replication conflicts, 132-133
  - scope and replication traffic, 126-130
  - security groups, 120-126
- home directories, 119
- NT 4.0 vs. Windows 2000, 18-19
- user accounts
  - copying, 112
  - creating, 111
  - deleting, 113
  - disabling, 114
  - editing properties, 113
  - enabling, 114
  - finding, 116
  - group memberships, 115
  - moving, 116
  - passwords, 114-115
  - predefined accounts, 110

- renaming, 113-114
- UPN (user principal name) suffixes, 115
- user profiles
  - advantages, 117
  - local, 117
  - mandatory, 118
  - roaming, 117-118
- administrative templates (Group Policy), 183-184**
- ADSI (Active Directory Services Interface), 9, 32, 314**
- ADSI Edit, 47-48**
- alerts**
  - actions, 306
  - adding counters to, 306
  - creating, 305
  - removing counters from, 307
  - start/stop parameters, 307
  - thresholds, 306
- All ICMP Traffic filter list (IPSec), 150**
- All IP Traffic filter list (IPSec), 150**
- APIs (application programming interfaces), 14**
- application setup files, 185**
- applications**
  - assigning, 186
  - publishing, 186-187
  - uninstalling, 188
  - upgrading, 187-188
- architecture, 26**
  - directory services, 28-30
    - database layer, 30
    - DSA (directory system agent), 30
    - ESE (Extensible Storage Engine), 31

- logical structure, 33
  - DNs (distinguished names), 38-40
  - domain hierarchy, 33-34
  - domain names, 35-36
  - forests, 37-38
  - GUIDs (globally unique identifiers), 41
  - logon names, 43-44
  - name formats, 42
  - name mapping, 43
  - naming attributes, 41
  - RDNs (relative distinguished names), 40
  - trees, 36-37
- physical structure, 44-45
  - data storage, 55-56
  - directory contents, 45-46
  - partitions. *See* partitions
  - sites, 54-55
- protocols, 31
  - ADSI (Active Directory Services Interface), 32
  - LDAP (Lightweight Directory Access Protocol), 32
  - replication, 33
  - X.500, 31
- subsystem, 26, 28
- ASCII character support, 69**
- assigning**
  - applications, 186
  - logon and logoff scripts, 189
  - object permissions, 161
  - ownership, 167-168
  - startup and shutdown scripts, 189



associating subnets with sites, 231

attribute-level security, 11

attributelD attribute, 218

attributes, 209

adding, 217-218

deactivating, 224

defined, 204

indexing, 209

mandatory, 210, 217-218

modifying, 218-219

naming attributes, 41

partition attributes, 47-48

reactivating, 223-224

site links, 235

syntax, 210

attributeSyntax

attribute, 218

authentication

Kerberos, 10, 140

*Active Directory and,*  
143

*advantages,* 141

*authentication*

*process,* 142-143

*clients,* 141

*history of,* 142

*KDC (Key*

*Distribution Center),*  
141

*keys,* 141

*limitations,* 146

*policies,* 144

*preauthentication,* 143

*servers,* 141

*Windows 2000*

*interoperability,* 145

PKI (public key

infrastructure), 146

*advantages,* 147

*CAs (certificate*

*authorities),* 147

*components,* 146-147

*SmartCards,* 148

required, 11

author mode (MMC), 5

authoritative restores,  
290-291

entire Active Directory

database, 292

impact on trust

relationships and

network connections,

294

Ntdsutil utility, 291-292

specific Active Directory

objects, 292-293

verifying, 294

authorizing DHCP

(Dynamic Host

Configuration

Protocol), 98

auxiliary classes, 211

auxiliaryClass attribute,  
220

## B

-b parameter

CSVDE (Comma-

Separated Value

Directory Exchange),

316

LDIFDE (LDAP Data

Interchange Format

Directory Exchange),

318

backslash (\), 42

Backup utility, 280-281

creating backups,

282-283

scheduling backups,

283-284

starting, 281

Backup wizard, 282-283

backups, 280

creating, 281-283

restoring data from. *See*

restoring Active

Directory

scheduling, 283-284

System State data, 281

backward compatibility,  
9-10

BIND (Berkeley Internet  
Name Domain) DNS  
servers, 88-91

BIND as primary DNS,  
92-94

DNS integration

options, 91

Windows 2000 DNS as

primary DNS, 91-92

zone transfers, 94-95

Block Policy Inheritance  
option (Group Policy),  
195-197

blocking policy

inheritance, 195-197

bridgehead servers,  
235-236

designating preferred  
servers, 237

multiple, 235

bridges, site link,  
237-239

browsers, Internet  
Explorer, 192

buffers (trace log),  
304-305

## C

-c parameter

CSVDE (Comma-

Separated Value

Directory Exchange),

315

- LDIFDE (LDAP Data Interchange Format Directory Exchange), 317
- caching-only servers, 72**
- canonical name (CNAME) resource records, 73**
- canonical names, 42**
- capturing network frames, 311-312**
- Catalog (Global), 7-8, 228**
- centralized management, 5-6**
- certificates, 10-11**
- changing. See editing**
- child domains, 60**
- Class field (SRV records), 74**
- classes**
  - 88 classes, 211
  - abstract, 211
  - adding, 219-220
  - auxiliary, 211
  - classSchema object, 210-211
  - deactivating, 224
  - defined, 204
  - modifying, 220-221
  - reactivating, 223-224
  - structural, 211
- classSchema object, 210-211**
- client keys, 141**
- cn attribute, 219**
- CNAME (canonical name) resource records, 73**
- Comma-Separated Value Directory Exchange (CSVDE), 314-316**
- commands, Action menu, 82. See also utilities**
- Common.adm template, 184**
- computer accounts**
  - adding to groups, 134
  - creating, 133
  - deleting, 134
  - disabling, 136
  - DNS (Domain Name System) names, 137-138
  - editing, 135
  - enabling, 137
  - finding, 135
  - moving, 136
  - properties, 135
  - resetting, 136
- computer policy, 180-181**
- configuration partitions, 46, 49-50**
- configuring**
  - DHCP (Dynamic Host Configuration Protocol)
    - address pools, 104*
    - DDNS (dynamic DNS) updates, 104*
    - exclusion ranges, 104*
    - reservations, 104*
    - scopes, 103*
  - DNS (Domain Name System) zones
    - adding zones, 83-84*
    - converting, 85-86*
    - deleting zones, 84*
    - integrated zones, 84*
    - troubleshooting, 86-87*
  - domain security policies, 144
  - Group Policy, 177-179
  - home directories, 119
  - partitions, 48-49
  - RIS (Remote Installation Services), 191
- conflicts (replication), 132-133**
- connection objects, 231-233**
- consistency checks, 222**
- containers, 45**
  - Group Policy, 200
  - System, 52-53
- /continue parameter (Movetree), 319**
- controllers, 53**
- convergence, 241**
- converting**
  - group type, 129
  - zones
    - integrated zones to standard zones, 86*
    - preventing problems, 86-87*
    - standard zones to integrated zones, 85*
- copying user accounts, 112**
- counter logs**
  - adding counters to, 303
  - creating, 302
- counters (System Monitor)**
  - adding, 297-298
  - Database object, 299-300
  - deleting, 298
  - loading, 299
  - NTDS object, 296-297
  - returning information about, 299
  - selecting, 300-301
- Create All Child Objects permission, 161**
- cross-link trusts, 66**
- CSVDE (Comma-Separated Value Directory Exchange), 314-316**

## D

**/d parameter (Movetree), 319**

**-d parameter**

CSVDE (Comma-Separated Value Directory Exchange), 315

LDIFDE (LDAP Data Interchange Format Directory Exchange), 317

**DACLs (Discretionary Access Control Lists), 109, 159**

**data restoration. See restoring Active Directory**

**data storage, 31, 55-56**

**database layer, 30**

**Database object, 299-300**

**databases**

Ntds.dit database files, 55  
restoring. *See* restoring Active directory

**/ddn parameter (Movetree), 319**

**DEA (directory-enabled application) platform, 14**

**deactivating.**

*See* disabling

**Default-First-Site, 227-228**

**defaultSecurityDescriptor attribute, 220**

**delegation, 168, 230**

administration, 12, 197-198

authentication, 141

**Delegation of Control Wizard, 198, 230**

**Delete All Child Objects permission, 161**

**deleting**

computer accounts, 134  
group members, 131-132  
groups, 129  
integrated zones, 84  
site link bridges, 239  
site links, 235  
sites, 230  
subnets, 231  
System Monitor counters, 298  
user accounts, 113  
zones, 84

**DEN (directory-enabled networking) platform, 14**

**denial of service attacks, 146**

**Deploy Software dialog box, 186**

**descriptors (security), 158-160**

**designating preferred bridgehead servers, 237**

**DHCP (Dynamic Host Configuration Protocol), 98**

advantages, 98-99  
authorizing, 98  
configuring  
    *address pools, 104*  
    *DDNS (dynamic DNS) updates, 104*  
    *exclusion ranges, 104*  
    *reservations, 104*  
    *scopes, 103*

integrating with DNS (Domain Name System), 101-103

lease process, 100-101

new features, 99

**dialog boxes**

Add a Script, 189  
Deploy Software, 186  
Filter Properties, 151  
IP Filter List, 151  
Remote Install, 191  
Test Attribute Properties, 224

**directories**

containers, 45  
DIT (Directory Information Tree), 206  
home directories, 119  
objects, 45  
OUs (organizational units), 45-46

**Directory Information Tree. See DIT, 206**

**directory partition replicas, 240**

**directory service architecture, 28-30**

database layer, 30  
DSA (directory system agent), 30  
ESE (Extensible Storage Engine), 31

**Directory Services Restore Mode, 286**

**directory-enabled application (DEA) platform, 14**

**directory-enabled networking (DEN) platform, 14**

**disabling**

attributes, 224  
classes, 224  
computer accounts, 136  
Kerberos preauthentication, 143  
objects, 223  
transitive site links, 238  
user accounts, 114

**Discover phase (DHCP leases), 100**

**Discretionary Access Control Lists (DACLs), 109, 159**

**disjointed**

namespaces, 16

**distinguished names (DNs), 38-40**

**distribution (software), 7, 120**

**DIT (Directory Information Tree), 206**

**DNs (distinguished names), 38-40**

**DNS (Domain Name System), 12, 68**

dynamic DNS, 76, 78

installing, 81-83

integrating with Active Directory, 80-81

integrating with DHCP (Dynamic Host Configuration Protocol), 101-103

mixed environments

*non-RFC-compliant data, 88*

*split DNS*

*configuration, 95-97*

*UNIX/BIND, 88-95*

*UTF-8 characters, 87*

*WINS and WINSR records, 87*

naming conventions, 35-36

*computer accounts, 137-138*

*name restrictions, 70-71*

*standards, 68-70*

RRs (resource records), 72-73

*A (address), 73*

*CNAME (canonical name), 73*

*MX (mail exchange), 74*

*NS (name server), 73*

*PTR (pointer), 73*

*SOA (Start of Authority), 73*

*SRV (service), 74-75*

server roles

*caching-only*

*servers, 72*

*forwarders, 72*

*primary name*

*servers, 71*

*secondary name*

*servers, 71*

*slaves, 72*

**WINS (Windows Internet Name Service), 97-98**

zone transfers, 78

*full, 79-80*

*incremental, 80*

zones, 75-76, 83

*adding, 83-84*

*converting, 85-86*

*deleting, 84*

*forward lookup*

*zones, 76*

*integrated zones, 84*

*reverse lookup*

*zones, 76*

*troubleshooting, 86-87*

*zone files, 75-76*

**domain controllers, 53**

monitoring performance of, 295-301

taking office, 286

**domain directory**

**partitions, 47, 51-53**

**domain GPOs (Group Policy Objects), 179**

**domain local security groups, 122-124**

**Domain Name System.**

**See DNS**

**domain naming masters (FSMOs), 258-259**

**domains, 16, 60-61**

child, 60

compared to sites, 55

creating, 62

hierarchical structure, 33-34

managing, 61-62

migrating to Windows 2000, 62

modes, 63

naming conventions, 35-36

root, 60

scalability, 61

security policies, 144

trust relationships, 64

*creating, 66-67*

*cross-link, 66*

*one-way, 65*

*revoking, 68*

*transitive, 65-66*

*verifying, 68*

**dsastat command, 249**

**dynamic DNS (Domain Name Service), 76, 78**

**Dynamic Host**

**Configuration Protocol.**

**See DHCP**

**Dynamic Update**

**Protocol, 89**

## E

**Edb.chk file, 281**

**editing**

attributes, 218-219

classes, 220-221

- computer accounts, 135
- domain modes, 63
- groups
  - memberships*, 115
  - group type*, 129
  - properties*, 130
  - scope*, 129-130
- IPSec filter actions, 153
- IPSec filter lists, 151
- partition attributes, 47-48
- schema, 212
  - enabling modification*, 214
  - verifying modifications*, 221-222
- server roles, 54
- software installation, 187
- user account properties, 113

## enabling

- attributes, 223-224
- classes, 223-224
- computer accounts, 137
- DHCP (Dynamic Host Configuration Protocol), 98
- objects, 223
- schema modification, 214
- user accounts, 114

## enforcing policies, 196-197

## errors.

**See troubleshooting**

## ESE (Extensible Storage Engine), 31

## event logs, 309-310

## Event Viewer, 309-310

## exclusion ranges (DHCP), 104

## explicit trusts, 66-67

## export-specific parameters (LDIFDE), 317-318

## extending schema, 13, 213

- FSMO role, 217
- LDIF (LDAP Data Interchange Format), 216
- methods, 213-214
- OIDs (object identifiers), 215
- order of processing, 216
- planning process, 212-213
- potential problems, 222
- Schema Admins membership, 215-216

## Extensible Storage Engine (ESE), 31

# F

## -f parameter

- CSVDE (Comma-Separated Value Directory Exchange), 315
- LDIFDE (LDAP Data Interchange Format Directory Exchange), 317

## fields

- security descriptors, 159
- SRV (Service) resource records, 74

## filename extensions, 185

## files

- log. *See* log files
- ownership, 167-168
- zone files, 75-76

## Filter Action Wizard, 152-153

## Filter Properties dialog box, 151

## filtering

- GPOs (Group Policy Objects), 200
- IPSec
  - filter actions*, 152-153
  - filter lists*, 149-151

## finding

- computer accounts, 135
- groups, 130
- users, 116

## flexible single-master operations. **See** FSMOs

## folder redirection, 192-193

## folders

- Group Policy template folder, 200-201
- ownership, 167-168
- redirecting, 192-193
- shared folders, 164-166

## forests, 16, 37-38

## forward lookup zones, 76

## forward slash (/), 42

## forwarders, 72

## FSMOs (Flexible Single-Master Operations), 9, 217, 252

- Directory schema updates, 252-253
- ntdsutil utility, 267
  - help*, 268
  - viewing FSMO roles*, 269-270
- operations master roles, 253-256
  - domain naming masters*, 258-259
  - infrastructure masters*, 263-264
  - PDCEs (primary domain controller emulators)*, 261-262

- relative identifier masters*, 260-261
- schema masters*, 256-257
- transferring*, 266-267, 271-272
- permission changes
  - domain naming master*, 270
  - infrastructure master*, 270
  - PDCE*, 270
  - RID master*, 271
- placing
  - per-domain role placements*, 266
  - per-forest role placements*, 266
  - Sites and Services snap-in*, 265-266
- role seizures, 272
- role transfers, 266-267, 271-272
- troubleshooting, 273
  - infrastructure errors*, 277-278
  - infrastructure master failures*, 274
  - master failures*, 273
  - other operations master failures*, 274-276
  - primary domain controller emulator failures*, 274
  - RID errors*, 278
  - technical explanations*, 276
- Full Control permission**, 161
- full replicas**, 240
- full zone transfers**, 79-80

## G

### -g parameter

- CSVDE (Comma-Separated Value Directory Exchange), 316
- LDIFDE (LDAP Data Interchange Format Directory Exchange), 317

### GC (Global Catalog), 7-8, 228

### global groups, 122, 124-125

### globally unique identifiers (GUIDs), 41

### governorID attribute, 219

### GPOs (Group Policy Objects), 174-176

- creating, 176-177
- domain, 179
- filtering scope of, 200
- inheritance, 174
- linking, 194-195
- local, 176, 179
- multiple, 201
- non-local, 176
- organizational unit, 179
- site, 179

### Group Policy, 7, 172

- Active Directory-based storage, 200
- administrative requirements, 173-174
- administrative templates, 183-184
- blocking policies, 195-197
- compared to System Policy Editor, 172-173
- configuring, 177-178
- containers, 200

- delegating administration, 197-198
- enforcing policies, 196-197
- folder redirection, 192-193
- GPOs (Group Policy Objects), 174-176
  - creating*, 176-177
  - domain*, 179
  - filtering scope of*, 200
  - inheritance*, 174
  - linking*, 194-195
  - local*, 176, 179
  - multiple*, 201
  - non-local*, 176
  - organizational unit*, 179
  - site*, 179
- Group Policy template folder, 200-201
- Internet Explorer maintenance, 192
- IPSec, 149
  - adding rules to*, 154-155
  - creating*, 153-154
  - defined*, 149
- Kerberos, 144
- mixed mode
  - permissions*, 199-200
  - upgrading computer accounts*, 199
- MMC snap-in, 180
  - Computer Configuration settings*, 180-181
  - User Configuration settings*, 181-182
- order of policy implementation, 178-179

## RIS (Remote Installation Services)

*configuring*, 191  
*installing*, 190  
*managing client installation images*, 191-192

## scripts

*logon and logoff scripts*, 189  
*startup and shutdown scripts*, 189  
*types*, 189  
*WSH (Windows Script Host)*, 188

## Security Settings

extension, 184-185

## Software Installation

snap-in, 185

*assigning applications*, 186  
*modifying software installation*, 187  
*publishing applications*, 186-187

*setting permissions*, 188

*uninstalling*

*applications*, 188

*upgrading*

*applications*, 187-188

special policies (account policies), 194

## Group Policy Objects.

See GPOs

## groups, 120

adding members to, 131  
 changing, 115  
 converting group type, 129  
 creating, 128

deleting, 129

distribution groups, 120

editing properties, 130

effect of domain mode, 127

finding, 130

nesting, 127-128

policies, 7

removing members from, 131-132

renaming, 132

replication conflicts, 132-133

scope and replication traffic, 126-130

security groups, 120-121  
*domain local groups*, 122-124

*global groups*, 122-125

*implementing*, 121-122

*machine local groups*, 126

*universal groups*, 122, 125-126

spanning security groups, 12

**guessing passwords**, 146

**GUIDs (globally unique identifiers)**, 41

## H

**hard drive partitions**, 20, 46

Active Directory vs.

Novell 5, 20-21

changing attributes of, 47-48

configuration partitions, 46, 49-50

configuring, 48-49

domain partitions, 47, 51-53

partial replica of domain directory partitions, 48  
 schema partitions, 47, 50-51

**headers**, 159

**help (ntdsutil utility)**, 268

**heterogeneous environments (DNS)**, 87

non-RFC-compliant data, 88

split DNS configuration, 95-97

UNIX/BIND, 88-91

*BIND as primary*

*DNS*, 92-94

*DNS integration options*, 91

*Windows 2000 DNS as primary DNS*, 91-92

*zone transfers*, 94-95

UTF-8 characters, 87

WINS and WINSR

records, 87

**hierarchy (domains)**, 33-34

**history of Kerberos**, 142

**home directories**, 119

## I

**-i parameter**

CSVDE (Comma-Separated Value Directory Exchange), 315

LDIFDE (LDAP Data Interchange Format Directory Exchange), 317



**IDs**

- GUIDs (globally unique identifiers), 41
- OIDs (object identifiers), 215
- SIDs (security identifiers), 108, 156-157

**import-specific parameters (LDIFDE), 318**

**Incremental Zone Transfer Protocol, 90**

**incremental zone transfers, 80**

**indexing, 209**

**Inetres.adm template, 184**

**infrastructure masters (FSMOs), 263-264**

**inheritance**

- GPOs (Group Policy Objects), 174
- Group Policy, 195
  - blocking policies, 195-197*
  - enforcing policies, 196-197*
- permissions, 168

**Installation Wizard, 258**

**installing**

- DNS (Domain Name System), 81-83
- Network Monitor, 311
- RIS (Remote Installation Services), 190

**integrated zones**

- converting to standard zones, 86
- creating, 84
- deleting, 84

**Intellimirror, 6**

**Internet, 22**

**Internet Explorer, 192**

**Internet Protocol. See IP interoperability, 12**

- ADC (Active Directory Connectors), 14
- DEA (directory-enabled application) platform, 14
- DEN (directory-enabled networking) platform, 14
- DNS (Domain Name System) standards, 12
- extensible schema, 13
- MSDSS (Microsoft Directory Synchronization Services), 13
- native LDAP (Lightweight Directory Access Protocol), 12-13
- open APIs (application programming interfaces), 14

**intersite replication, 33, 247-248**

**intrasite replication, 33, 246-247**

**IP (Internet Protocol)**

- IPSec, 148
  - advantages/*
  - disadvantages, 148*
  - filter actions, 152-153*
  - filter lists, 149-151*
  - policies, 149, 153-155*
- replication, 244

**IP Filter List dialog box, 151**

**ipconfig command, 230**

**IPSec, 148**

- advantages/disadvantages, 148
- filter actions
  - creating, 152-153*
  - editing, 153*

*Permit, 152*

*Request Security, 152*

*Require Security, 152*

filter lists, 149-151

*All ICMP Traffic, 150*

*All IP Traffic, 150*

*creating, 150-151*

*editing, 151*

policies

*adding rules to, 154-155*

*creating, 153-154*

*defined, 149*

**isMemberOfPartialReplicaSet attribute, 218**

**isSingleValued attribute, 218**

## J-K

**-j parameter**

- CSVDE (Comma-Separated Value Directory Exchange), 315
- LDIFDE (LDAP Data Interchange Format Directory Exchange), 318

**-k parameter**

- CSVDE (Comma-Separated Value Directory Exchange), 316
- LDIFDE (LDAP Data Interchange Format Directory Exchange), 318

**KCC (Knowledge Consistency Checker), 227**



**KDC (Key Distribution Center), 141****Kdcsvc.dll file (LSA), 28****Kerberos, 10, 28, 140**

- Active Directory and, 143
- advantages, 141
- authentication process, 142-143
- clients, 141
- history of, 142
- KDC (Key Distribution Center), 141
- keys, 141
- limitations, 146
- policies, 144
- preauthentication, 143
- servers, 141
- Windows 2000 interoperability, 145

**Kerberos.dll file (LSA), 28****key discovery, 146****Key Distribution Center (KDC), 141****keys, 141**

- client, 141
- KDC (Key Distribution Center), 141
- key discovery, 146
- PKI (public key infrastructure), 146
  - advantages, 147*
  - CAs (certification authorities), 147*
  - components, 146-147*
  - SmartCards, 148*
- server, 141
- session, 141

**Knowledge Consistency Checker (KCC), 227****L****-l parameter**

- CSVDE (Comma-Separated Value Directory Exchange), 315
- LDIFDE (LDAP Data Interchange Format Directory Exchange), 317

**LDAP (Lightweight Directory Access Protocol), 12-13, 32**

- LDAP over SSL, 11
- LDIF (LDAP Data Interchange Format), 216
- LDIFDE (LDAP Data Interchange Format Directory Exchange), 316-318

**LDAPDisplayName attribute, 218, 220****LDIF (LDAP Data Interchange Format), 216****LDIFDE (LDAP Data Interchange Format Directory Exchange), 316-318****leaf nodes, 15****leases (DHCP), 100-101****Lightweight Directory Access Protocol. *See* LDAP****linking GPOs (Group Policy Objects), 194-195****links**

- site link bridges, 237-239
- site links
  - attributes, 235*
  - creating, 233-234*

- deleting, 235*

- turning off, 238*

**lists**

- ACLs (Access Control Lists), 155-156
  - ACEs (Access Control Entries), 156*
  - SIDs (security identifiers), 156-157*
- IPSec filter lists, 149-151
  - All ICMP Traffic, 150*
  - All IP Traffic, 150*
  - creating, 150-151*
  - editing, 151*

**loading System Monitor counters, 299****local GPOs (Group Policy Objects), 176, 179****local profiles, 117****Local Security Authority (LSA), 27-28****local security groups**

- domain local, 123-124
- machine local, 126

**locating. *See* finding****log files, 301**

- counter logs
  - adding counters to, 303*
  - creating, 302*
- event logs, 309-310
- trace logs
  - buffers, 304-305*
  - creating, 303*
  - providers and events, 304*

**logical structure, 33**

- DNs (distinguished names), 38-40
- domain hierarchy, 33-34
- domain names, 35-36
- forests, 37-38
- GUIDs (globally unique identifiers), 41

- logon names, 43-44
- name formats, 42
- name mapping, 43
- naming attributes, 41
- NT 4.0 vs. Windows 2000, 17-18
- RDNs (relative distinguished names), 40
- trees, 36-37
- logoff scripts, 189**
- logon names, 43-44, 115**
- logon scripts, 189**
- lookup zones (DNS), 76**
- loop zones (DNS), 83-84**
- loose consistency, 241**
- LSA (Local Security Authority), 27-28**
- Lsasrv.dll file (LSA), 28**

## M

### -m parameter

- CSVDE (Comma-Separated Value Directory Exchange), 316
- LDIFDE (LDAP Data Interchange Format Directory Exchange), 318
- mail exchange (MX) resource records, 74**
- manageability of Active Directory, 4**
  - ADSI (Active Directory Services Interfaces), 9
  - backward compatibility, 9-10
  - centralized management, 5-6
  - FSMOs (flexible single-master operations), 9

- GC (Global Catalog), 7-8
- group policy, 7
- multi-master replication, 10
- software distribution, 7
- transitive trusts, 8
- management. See administration**
- mandatory attributes, 210, 217-218**
- mandatory profiles, 118**
- mapping names, 43**
- master roles (FSMOs), 9, 253-256**
  - domain naming masters, 258-259
  - infrastructure masters, 263-264
  - multi-master replication, 10
  - PDCEs (primary domain controller emulators), 261-262
  - relative identifier masters, 260-261
  - schema masters, 256-257
  - transferring, 266-267, 271-272
- member servers, 53-54**
- members**
  - adding to groups, 131
  - removing from groups, 131-132
- Meta-directory Services, 19**
- Microsoft Directory Synchronization Services (MSDSS), 13**
- Microsoft Management Console. See MMC**
- migrating Windows NT domains to Windows 2000, 62**

### **mixed environments (DNS), 87**

- non-RFC-compliant data, 88
- split DNS configuration, 95-97
- UNIX/BIND, 88-91
  - BIND as primary DNS, 92-94*
  - DNS integration options, 91*
  - Windows 2000 DNS as primary DNS, 91-92*
  - zone transfers, 94-95*
- UTF-8 characters, 87
- WINS and WINSR records, 87

### **mixed mode, 63**

- permissions, 199-200
- upgrading computer accounts, 199

### **MMC (Microsoft Management Console), 5-6**

- Computer Configuration settings, 180-181
- User Configuration settings, 181-182

### **mmc command, 5**

### **modes**

- Directory Services Restore Mode, 286
- domain modes, 63
- MMS (Microsoft Management Console), 5

### **Modified Windows**

- Installer packages, 185**

### **modifying. See editing**

### **monitoring performance, 295**

- event logs, 309-310
- Network Monitor, 310-312

Performance Logs and  
Alerts utility, 301-307  
System Monitor, 295-301  
Task Manager, 307-308

## **Movetree, 319**

### **moving**

computer accounts, 136  
user accounts, 116

## **MSDSS (Microsoft**

### **Directory**

### **Synchronization**

### **Services), 13**

**.msi filename extension,**  
**185**

**.mst filename extension,**  
**185**

**Msvl.dll file (LSA), 28**

**multi-master replication,**  
**10, 241**

**Multibyte (UTF-) option**  
**(server properties), 71**

**multiple GPOs (Group**  
**Policy Objects), 201**

**mutual authentication,**  
**141**

**MX (mail exchange)**  
**resource records, 74**

## **N**

### **-n parameter**

CSVDE (Comma-  
Separated Value  
Directory Exchange),  
316  
LDIFDE (LDAP Data  
Interchange Format  
Directory Exchange),  
318

**Name field (SRV**  
**records), 74**

**name formats, 42**

**name mapping, 43**

**name resolution (DNS),**  
**68-71**

**name server (NS)**

**resource records, 73**

**namespaces, 14-15**

**naming attributes, 41**

**naming conventions**

DNs (distinguished  
names), 38-40

DNS (Domain Name  
System)

*name restrictions,*  
*70-71*

*standards, 68-70*

domains, 35-36

GUIDs (globally unique  
identifiers), 41

groups, 132

logon names, 43-44

name formats, 42

name mapping, 43

naming attributes, 41

objects, 45

RDNs (relative  
distinguished  
names), 40

sites, 228

**National Registration**  
**Authority (NRA), 215**

**native mode**

**(domains), 63**

**NDS (Novell Directory**  
**Services), 19**

catalogs, 22

Internet standards

support, 22

partitions, 20-21

**nesting groups, 127-128**

**Net Logon service, 28**

**Netlogon.dll file**  
**(LSA), 28**

**Network Monitor, 310**

capturing network

frames, 311-312

installing, 311

**networks. See also sites**

Network Monitor, 310

*capturing network*

*frames, 311-312*

*installing, 311*

subnets, 230-231

**New Delegation**

**command (Action**

**menu), 82**

**No Override option**

**(Group Policy), 196-197**

**Non RFC (ANSI) option**

**(server properties), 71**

**non-local GPOs (Group**  
**Policy Objects), 176**

**non-RFC-compliant**  
**data, 88**

**nonauthoritative**

**restores, 286**

implications of, 288-289

step-by-step process,  
286-287

verifying, 289

**noncontiguous**

**namespaces, 16**

**Novell 5**

compared to Active

Directory, 19

*catalogs, 21-22*

*Internet standards*

*support, 22*

*partitions, 20-21*

NDS (Novell Directory

Services), 19

*catalogs, 22*

*Internet standards*

*support, 22*

*partitions, 20-21*

**NRA (National Registration Authority), 215**  
**NS (name server)**  
 resource records, 73  
**NT 4.0.**  
 See Windows NT 4.0  
**NT LAN Manager (NTLM), 28**  
**Ntbackup command, 281**  
**NTDS object, 296-297**  
**Ntds.dit database files, 55, 281**  
**Ntdsa.dll file (LSA), 28**  
**ntdsutil utility, 258-259, 267**  
 authoritative restores, 291-292  
 help, 268  
 viewing FSMO roles, 269-270  
**NTLM (NT LAN Manager), 28**  
**nTSecurityDescriptor attribute, 218-220**  
**numbers, USNs (Update Sequence Numbers), 243**

## O

**-o parameter**  
 CSVDE (Comma-Separated Value Directory Exchange), 315  
 LDIFDE (LDAP Data Interchange Format Directory Exchange), 317  
**object identifiers (OIDs), 215**

**object-oriented security, 155**  
**objectClass attribute, 218**  
**objectClassCategory attribute, 219**  
**objects, 45, 160**  
 classSchema, 210-211  
 connection, 231-233  
 Database, 299-300  
 deactivating, 223  
 GPOs (Group Policy Objects), 174-176  
   *creating, 176-177*  
   *domain, 179*  
   *filtering scope of, 200*  
   *inheritance, 174*  
   *linking, 194-195*  
   *local, 176, 179*  
   *multiple, 201*  
   *non-local, 176*  
   *organizational unit, 179*  
   *site, 179*  
 naming conventions, 45  
 NTDS, 296-297  
 OIDs (object identifiers), 215  
 permissions  
   *assigning, 161*  
   *Create All Child Objects, 161*  
   *Delete All Child Objects, 161*  
   *Full Control, 161*  
   *propagation, 162-163*  
   *Read, 161*  
   *Write, 161*  
 reactivating, 223  
 security. *See* security  
 site link  
   *attributes, 235*  
   *creating, 233-234*

*deleting, 235*  
   *turning off, 238*  
   site link bridge, 237-239  
   types, 160  
**OIDs (object identifiers), 215**  
**oMSyntax attribute, 218**  
**one-way trusts, 65**  
**open APIs (application programming interfaces), 14**  
**operations master roles (FSMOs), 253-256**  
 domain naming masters, 258-259  
 infrastructure masters, 263-264  
 PDCEs (primary domain controller emulators), 261-262  
 relative identifier masters, 260-261  
 schema masters, 256-257  
 transferring, 266-267, 271-272  
**order of Group Policy implementation, 178-179**  
**organization units. See OUs, 17**  
**organizational unit GPOs (Group Policy Objects), 179**  
**organizational units (OUs), 17, 45-46**  
**originating updates, 243-244**  
**OU (organizational unit) GPOs, 179**  
**OUs (organizational units), 17, 45-46**  
**Owner field (security descriptors), 159**  
**ownership, 167-168**

## P

### -p parameter

CSVDE (Comma-Separated Value Directory Exchange), 315  
LDIFDE (LDAP Data Interchange Format Directory Exchange), 317

### parameters

CSVDE (Comma-Separated Value Directory Exchange), 314-316  
LDIFDE (LDAP Data Interchange Format Directory Exchange), 317

Movetree, 319

### partial replicas, 240

### partitions, 20, 46

Active Directory vs. Novell 5, 20-21  
changing attributes of, 47-48  
configuration partitions, 46, 49-50  
configuring, 48-49  
domain partitions, 47, 51-53  
partial replica of domain directory partitions, 48  
schema partitions, 47, 50-51

### passwords

password guessing, 146  
resetting, 114-115

### PDCEs (primary domain controller emulators), 261-262

### per-domain role placements, 266

### per-forest role

#### placements, 266

### performance logs, 301

counter logs  
    *adding counters to*, 303  
    *creating*, 302  
trace logs  
    *buffers*, 304-305  
    *creating*, 303  
    *providers and events*, 304

### Performance Logs and Alerts utility, 301

alerts  
    *actions*, 306  
    *adding counters to*, 306  
    *creating*, 305  
    *removing counters from*, 307  
    *start/stop parameters*, 307  
    *thresholds*, 306  
counter logs  
    *adding counters to*, 303  
    *creating*, 302  
trace logs  
    *buffers*, 304-305  
    *creating*, 303  
    *providers and events*, 304

### performance

#### monitoring, 295

event logs, 309-310  
Network Monitor, 310  
    *capturing network frames*, 311-312  
    *installing*, 311  
Performance Logs and Alerts utility, 301  
    *alerts*, 305-307  
    *counter logs*, 302-303  
    *trace logs*, 303-305

### System Monitor, 295-296

*adding counters*, 297-298  
    *Database object*, 299-300  
    *deleting counters*, 298  
    *loading and unloading counters*, 299  
    *NTDS object*, 296-297  
    *returning counter information*, 299  
    *selecting counters*, 300-301  
    *starting*, 295  
Task Manager, 307-308

### permissions, 157

conflicts with privileges, 168

### FSMOs (Flexible Single-Master Operations)

*domain naming master permission changes*, 270  
    *infrastructure master permission changes*, 270  
    *PDCE permission changes*, 270  
    *RID master permission changes*, 271

### Group Policy, 199-200

inheritance, 168

### object permissions

*assigning*, 161  
    *Create All Child Objects*, 161  
    *Delete All Child Objects*, 161  
    *Full Control*, 161  
    *propagation*, 162-163

- Read*, 161
- Write*, 161
- software installation, 188
- Permit filter action (IPSec), 152**
- physical structure, 44-45**
  - data storage, 55-56
  - directory contents
    - containers*, 45
    - objects*, 45
    - OUs (organizational units)*, 45-46
  - NT 4.0 vs. Windows 2000, 18
  - partitions, 46
    - changing attributes of*, 47-48
    - configuration partitions*, 46, 49-50
    - configuring*, 48-49
    - domain partitions*, 47, 51-53
    - partial replica of domain directory partitions*, 48
    - schema partitions*, 47, 50-51
  - sites, 54-55
- PKI (public key infrastructure), 10-11**
  - advantages, 147
  - CAs (certification authorities), 147
  - components, 146-147
  - SmartCards, 148
- placing FSMOs (Flexible Single-Master Operations), 265**
  - per-domain role
    - placements, 266
  - per-forest role
    - placements, 266
  - Sites and Services
    - snap-in, 265-266
- pointer (PTR) resource records, 73**
- policies. See Group Policy**
- Port field (SRV records), 74**
- posSuperiors attribute, 219**
- preauthentication (Kerberos), 143**
- predefined user accounts, 110**
- primary domain controller emulators (PDCEs), 261-262**
- Primary Group field (security descriptors), 159**
- primary name servers, 71**
- primary zones, 83**
- printers, publishing, 166-167**
- Priority field (SRV records), 74**
- privileges, 168**
- profiles (user)**
  - advantages, 117
  - local, 117
  - mandatory, 118
  - roaming, 117-118
- propagating object permissions, 162-163**
- properties**
  - computer accounts, 135
  - group properties, 130
  - user accounts, 113
- Proto field (SRV records), 74**
- protocols, 31**
  - ADSI (Active Directory Services Interface), 32
- DHCP (Dynamic Host Configuration Protocol), 98
  - address pools*, 104
  - advantages*, 98-99
  - authorizing*, 98
  - configuring*, 103-104
  - DDNS (dynamic DNS) updates*, 104
  - exclusion ranges*, 104
  - integrating with DNS*, 101-103
  - lease process*, 100-101
  - new features*, 99
  - reservations*, 104
  - scopes*, 103
- DNS (Domain Name System), 68
  - dynamic DNS*, 76-78
  - installing*, 81-83
  - integrating with Active Directory*, 80-81
  - integrating with DHCP*, 101-103
  - mixed environments*, 87-97
  - naming conventions*, 35-36, 68-70
  - naming restrictions*, 70-71
  - RRs (resource records)*, 72-75
  - server roles*, 71-72
  - WINS (Windows Internet Name Service)*, 97-98
  - zone transfers*, 78-80
  - zones*, 75-76, 83-87
- Dynamic Update Protocol, 89
- Incremental Zone Transfer Protocol, 90
- IP (Internet Protocol), 244

IPSec, 148  
     *advantages/*  
         *disadvantages, 148*  
     *filter actions, 152-153*  
     *filter lists, 149-151*  
     *policies, 149, 153-155*

Kerberos, 10, 28, 140  
     *Active Directory and,*  
         *143*  
     *advantages, 141*  
     *authentication*  
         *process, 142-143*  
     *clients, 141*  
     *KDC (Key*  
         *Distribution Center),*  
         *141*  
     *keys, 141*  
     *limitations, 146*  
     *policies, 144*  
     *preauthentication, 143*  
     *servers, 141*  
     *Windows 2000*  
         *interoperability, 145*

LDAP (Lightweight  
     Directory Access  
     Protocol), 12-13, 32  
     *LDAP over SSL, 11*  
     *LDIFDE (LDAP Data*  
         *Interchange Format*  
         *Directory Exchange),*  
         *316-318*

RPC (Remote Procedure  
     Calls), 244-245  
     SMTP, 244-245  
     X.500, 31

**PTR (pointer) resource  
     records, 73**

**public key  
     infrastructure. See PKI**  
**publishing**

    Active Directory  
         resources, 164  
         *guidelines, 167*  
         *printers, 166-167*

*shared folders,*  
             *164-166*  
     applications, 186-187  
**pull replication, 241-242**

## Q-R

**/quiet parameter  
     (Movetree), 319**

**-r parameter**  
     CSVDE (Comma-  
         Separated Value  
         Directory Exchange),  
         315  
     LDIFDE (LDAP Data  
         Interchange Format  
         Directory Exchange),  
         317

**rangeLower attribute,  
     218**

**rangeUpper attribute,  
     218**

**RDNs (relative  
     distinguished  
     names), 40**

**reactivating**  
     attributes, 223-224  
     classes, 223-224  
     objects, 223

**Read permission, 161**

**redirecting folders,  
     192-193**

**referral (WINS), 97-98**

**Registry, tattooing, 173**  
**relationships, trust, 64,  
     202**

    authoritative restores and,  
         294  
     creating, 66-67  
     cross-link, 66  
     one-way, 65

    revoking, 68  
     transitive, 8, 65-66  
     verifying, 68

**relative identifier  
     masters (FSMOs),  
     260-261**

**Remote Install dialog  
     box, 191**

**Remote Installation  
     Services (RIS), 190**  
     configuring, 191  
     installing, 190  
     managing client  
         installation images,  
         191-192

**Remote Installation  
     Services Setup Wizard,  
     190**

**removing. See deleting  
     renaming**

    groups, 132  
     sites, 228  
     user accounts, 113-114

**REPADMIN (Replication  
     Administrator), 249**

**Repadmin utility, 294**

**replication, 33, 239**  
     benefits of, 240-241  
     conflicts, 132-133  
     directory partition  
         replicas, 240  
         *full replicas, 240*  
         *partial replicas, 48,*  
         *240*

    intersite, 247-248  
     intrasite, 246-247  
     multi-master, 241  
     pull, 241-242  
     replication tombstones,  
         248  
     state-based, 241-242  
     store-and-forward,  
         241-242



- tools
  - dsastat*, 249
  - REPADMIN*
    - (*Replication Administrator*), 249
  - REPLMON*
    - (*Replication Monitor*), 248
  - transports and protocols
    - IP (Internet Protocol)*, 244
    - RPC (Remote Procedure Calls)*, 244-245
    - SMTP*, 244-245
  - updates
    - originating*, 243-244
    - update propagation*, 243
    - USNs (Update Sequence Numbers)*, 243
- Replication Administrator (REPADMIN), 249**
- Replication Monitor (REPLMON), 248**
- replication tombstones, 248**
- REPLMON (Replication Monitor), 248**
- Request Security filter action (IPSec), 152**
- Require Security filter action (IPSec), 152**
- reservations (DHCP), 104**
- resetting**
  - computer accounts, 136
  - user passwords, 114-115
- resolving names. See name resolution**
- resource records. See RRs**
- resources, publishing, 164**
  - guidelines, 167
  - printers, 166-167
  - shared folders, 164-166
- restoring Active Directory, 284-285**
  - authoritative restores, 290-291
    - entire Active Directory database*, 292
    - impact on trust relationships and network connections*, 294
    - Ntdsutil utility*, 291-292
    - specific Active Directory objects*, 292-293
    - verifying*, 294
  - nonauthoritative restores, 286
    - implications of*, 288-289
    - step-by-step process*, 286-287
    - verifying*, 289
  - through reinstallation and replication, 285
- restrictions (DNS names), 70-71**
- reverse lookup zones, 76, 83-84**
- revoking trusts, 68**
- rights, 108, 157**
- RIS (Remote Installation Services), 190**
  - configuring, 191
  - installing, 190
  - managing client installation images, 191-192
- roaming profiles, 117-118**
- roles**
  - operations master roles (FSMOs), 253-256
    - domain naming masters*, 258-259
    - infrastructure masters*, 263-264
    - PDCEs (primary domain controller emulators)*, 261-262
    - relative identifier masters*, 260-261
    - schema masters*, 256-257
    - transferring*, 266-267, 271-272
  - server roles, 54
- root domains, 60**
- RPC (Remote Procedure Calls), 244-245**
- RRs (resource records), 72-73**
  - A (address), 73
  - CNAME (canonical name), 73
  - MX (mail exchange), 74
  - NS (name server), 73
  - PTR (pointer), 73
  - SOA (Start of Authority), 73
  - SRV (service), 74-75

## S

**/s parameter (Movetree), 319**

**-s parameter**  
CSVDE (Comma-Separated Value Directory Exchange), 315



LDIFDE (LDAP Data Interchange Format Directory Exchange), 317

**SACLs (System Access Control Lists), 109, 159**

**safety checks, 222**

**SAM (Security Accounts Manager), 28**

**Samsrv.dll file (LSA), 28**

**scalability (domains), 61**

**Schannel.dll file (LSA), 28**

**schema (Active Directory), 204-206**

Active Directory Schema snap-in, 207-209

attributes, 209

adding, 217-218

deactivating, 224

defined, 204

indexing, 209

mandatory, 210,

217-218

modifying, 218-219

reactivating, 223-224

syntax, 210

classes

88 classes, 211

abstract, 211

adding, 219-220

auxiliary, 211

classSchema object, 210-211

deactivating, 224

defined, 204

modifying, 220-221

reactivating, 223-224

structural, 211

**DIT (Directory Information Tree), 206**

extending, 213

*FSMO role, 217*

*LDIF (LDAP Data*

*Interchange*

*Format), 216*

*methods, 213-214*

*OIDs (object identifiers), 215*

*order of processing, 216*

*planning process, 212-213*

*potential problems, 222*

*Schema Admins membership, 215-216*

modifying, 212, 222

consistency checks, 222

enabling modification, 214

safety checks, 222

objects, 223

partitions, 47, 50-51

syntax, 204

updates, 252-253

verifying modifications to, 221-222

**Schema Admins groups, 215-216**

**schema masters (FSMOs), 256-257**

**Schema snap-in, 257**

**schemalDGUID**

**attribute, 218-220**

**scope**

DHCP (Dynamic Host Configuration Protocol), 103

groups, 126, 129-130

**scripts**

logon and logoff scripts, 189

startup and shutdown scripts, 189

types, 189

WSH (Windows Script Host), 188

**/sdn parameter (Movetree), 319**

**searchFlags attribute, 218**

**searching. See finding secondary name**

**servers, 71**

**Secur32.dll file (LSA), 28**

**Secure Sockets Layer.**

**See SSL**

**security, 10, 140**

access control, 168-169

access tokens, 108, 156

ACEs (Access Control Entries), 109, 156

ACLs (Access Control Lists), 109, 155-156

*ACEs (Access Control Entries), 156*

*DACLs (Discretionary Access Control Lists), 109*

*SACLs (System Access Control Lists), 109*

*SIDs (security identifiers), 156-157*

attribute-level, 11

delegation, 12, 168

denial of service attacks, 146

IPSec, 148

*advantages/disadvantages, 148*

*filter actions, 152-153*

*filter lists, 149-151*

*policies, 149, 153-155*

Kerberos, 10, 28, 140

*Active Directory and, 143*

*advantages, 141*

*authentication*

*process, 142-143*

- clients*, 141
- history of*, 142
- KDC (Key Distribution Center)*, 141
- keys*, 141
- limitations*, 146
- policies*, 144
- preauthentication*, 143
- servers*, 141
- Windows 2000 interoperability*, 145
- key discovery, 146
- LDAP over SSL, 11
- LSA (Local Security Authority), 27-28
- Net Logon service, 28
- NTLM (NT LAN Manager), 28
- object-oriented, 155
- objects, 160-163
- ownership, 167-168
- passwords
  - guessing*, 146
  - resetting*, 114-115
- permissions, 157, 161-163
- PKI (public key infrastructure), 10-11, 146
  - advantages*, 147
  - CAs (certification authorities)*, 147
  - components*, 146-147
  - SmartCards*, 148
- policies. *See* Group Policy
- publishing Active Directory resources, 164
  - guidelines*, 167
  - printers*, 166-167
  - shared folders*, 164-166
- required authentication, 11
- rights, 157
- SAM (Security Accounts Manager), 28
- security descriptors, 158-160
  - DACLs (Discretionary Access Control Lists)*, 159
  - headers*, 159
  - Owner field*, 159
  - Primary Group field*, 159
  - SACLs (System Access Control Lists)*, 159
- security groups, 120-121
  - domain local groups*, 122-124
  - global groups*, 122-125
  - implementing*, 121-122
  - machine local groups*, 126
  - universal groups*, 122, 125-126
- security subsystem
  - architecture, 26
- SIDs (security identifiers), 108
- spanning security groups, 12
- user rights, 108
- Security Accounts Manager (SAM), 28**
- security descriptors, 158-160**
- security groups, 120-121**
  - domain local groups*, 122-124
  - filtering GPO (Group Policy Object) scope*, 200
  - global groups*, 122-125
  - implementing*, 121
  - machine local groups*, 126
  - universal groups*, 122, 125-126
- security identifiers (SIDs), 108, 156-157**
- security policies. *See* Group Policy**
- Security Settings extension (Group Policy), 184-185**
- seizures (role), 272**
- server keys, 141**
- servers**
  - bridgehead servers, 235-236
    - designating preferred servers*, 237
    - multiple*, 235
- DHCP (Dynamic Host Configuration Protocol), 98
  - address pools*, 104
  - advantages*, 98-99
  - authorizing*, 98
  - configuring*, 103-104
  - DDNS (dynamic DNS) updates*, 104
  - exclusion ranges*, 104
  - integrating with dynamic DNS (Domain Name System)*, 101-103
  - lease process*, 100-101
  - new features*, 99
  - reservations*, 104
  - scopes*, 103

## DNS (Domain Name Service)

- caching-only*, 72
- forwarders*, 72
- primary name servers*, 71
- secondary name servers*, 71
- slaves*, 72

- Kerberos, 141

- member servers, 53-54
- roles, 54

## service (SRV) resource records, 74-75

## Service field (SRV records), 74

## session keys, 141

## shared folders, 164-166

## shortcut trusts, 66

## shutdown scripts, 189

## SIDs (security identifiers), 108, 156-157

## single-master operations. See FSMOs (flexible single-master operations)

## site GPOs (Group Policy Objects), 179

## site link bridge objects, 237-239

## site link objects

- attributes, 235
- creating, 233-234
- deleting, 235
- turning off, 238

## sites, 17, 54

- Active Directory Sites and Services snap-in, 228

- bridgehead servers, 235-236

- designating preferred servers*, 237
- multiple*, 235

- compared to domains, 55

- connections, 231-233

- creating, 228-229

- Default-First-Site, 227-228

- defined, 227

- delegating control, 230

- deleting, 230

- renaming, 228

- replication. *See*

- replication

- site link bridges, 237-239

- site links

- attributes*, 235

- creating*, 233-234

- deleting*, 235

- turning off*, 238

- Sites and Services

- snap-in, 228

- subnets, 230-231

- topology, 226

## Sites and Services

- snap-in, 265-266

## slash (/), 42

## slaves, 72

## SmartCards, 148

## SMTP, 244-245

## sn attribute, 218

## snap-ins

- MMC snap-in extension model, 180

- Computer*

- Configuration*

- settings*, 180-181

- User Configuration*

- settings*, 181-182

- Schema, 207-209, 257

- Sites and Services, 228

- delegating control*, 230

- designating bridgehead servers*, 237

- site link bridges*, 239

- site links*, 233-238

- sites*, 228-230

- subnets*, 231

- Software Installation, 7, 185

- assigning*

- applications*, 186

- modifying software*

- installation*, 187

- publishing*

- applications*,

- 186-187

- setting permissions*, 188

- uninstalling*

- applications*, 188

- upgrading*

- applications*,

- 187-188

## SOA (Start of Authority)

- resource records, 73

## sockets, SSL (Secure Sockets Layer), 11, 28

## software

- distribution, 7

- REPADMIN (Replication Administrator), 249

- REPLMON (Replication Monitor), 248

## Software Installation

- snap-in, 7, 185

- assigning applications, 186

- modifying software

- installation, 187

- publishing applications, 186-187

- setting permissions, 188

- uninstalling applications, 188

- upgrading applications, 187-188

## spanning security groups, 12

## special policies, 194

**split DNS configuration,**  
95-97

**SRV (service) resource  
records, 74-75**

**SSL (Secure Sockets  
Layer), 11, 28**

**/start parameter  
(Movetree), 319**

**Start of Authority (SOA)  
resource records, 73**

**start/stop parameters  
(alerts), 307**

**starting**

Active Directory Schema  
snap-in, 207

Backup utility, 281

System Monitor, 295

Task Manager, 308

**startup scripts, 189**

**state-based replication,  
241-242**

**stop/start parameters  
(alerts), 307**

**storage, 31, 55-56**

**store-and-forward  
replication, 241-242**

**Strict RFC (ANSI) option  
(server properties), 71**

**structural classes, 211**  
**subClassOf attribute,  
219**

**subnets, 230-231**

**subsystem architecture,  
26, 28**

**syntax**

attribute syntax, 210

defined, 204

**System Access Control  
Lists (SACLs), 109, 159**

**system administration.**

See administration

**system checks, 222**

**System container, 52-53**

**System Monitor utility,  
295-296**

adding counters, 297-298

Database object, 299-300

deleting counters, 298

loading and unloading  
counters, 299

NTDS object, 296-297

returning counter  
information, 299

selecting counters,  
300-301

starting, 295

**System Policy Editor,  
172-173**

**System State data, 281**

**System.adm template,  
184**

## T

**-t parameter**

CSVDE (Comma-  
Separated Value  
Directory Exchange),  
315

LDIFDE (LDAP Data  
Interchange Format  
Directory Exchange),  
317

**taking Active Directory  
offline, 286**

**Target field (SRV  
records), 74**

**Task Manager, 307-308**

**tattooing (Registry), 173**  
**templates (Group**

**Policy), 183-184,  
200-201**

**Test Attribute Properties  
dialog box, 224**

**thresholds (alerts), 306**

**tokens, 108, 156**

**tombstones  
(replication), 248**

**tools. See utilities**

**topology (replication),  
226**

intersite replication,  
247-248

intrasite replication,  
246-247

replication tombstones,  
248

transports and protocols  
*IP (Internet Protocol),  
244*

*RPC (Remote  
Procedure Calls),  
244-245*

*SMTP, 244-245*

**trace logs, 303-305**

buffers, 304-305

creating, 303

providers and events, 304

**transferring**

operations master roles,  
266-267, 271-272

zone transfers, 78  
*full, 79-80*  
*incremental, 80*

**transitive site links, 238**

**transitive trusts, 8, 65-66**

**trees, 15-16, 36-37**

**troubleshooting**

FSMOs (Flexible Single-  
Master Operations),  
273, 277

*infrastructure errors,  
277-278*

*infrastructure master  
failures, 274*

*master failures, 273*

*other operations  
master failures,  
274-276*

- primary domain*
- controller emulator*
- failures, 274*
- RID errors, 278*
- technical*
- explanations, 276*
- zones, 86-87

## **trust relationships, 64, 202**

- authoritative restores and, 294
- creating, 66-67
- cross-link, 66
- one-way, 65
- revoking, 68
- transitive, 65-66
- transitive trusts, 8
- verifying, 68

## **TTL field (SRV records), 74**

### **turning off.**

**See disabling**

### **turning on. See enabling**

## **U**

### **/u parameter (Movetree), 319**

### **-u parameter**

- CSVDE (Comma-Separated Value Directory Exchange), 316
- LDIFDE (LDAP Data Interchange Format Directory Exchange), 318

### **Unicode character support, 69**

### **Uniform Resource Locators (URLs), 42**

### **uninstalling**

**applications, 188**

### **universal groups, 122, 125-126**

### **unloading System**

**Monitor counters, 299**

### **Update Sequence**

**Numbers (USNs), 243**

### **updates, 243**

- originating, 243-244
- update propagation, 243
- USNs (Update Sequence Numbers), 243
- schemas, 252-253

### **upgrading**

- applications, 187-188
- computer accounts, 199

### **UPNs (user principal names), 43, 115**

### **URLs (Uniform Resource Locators), 42**

### **user authentication, 140**

- Kerberos, 140
  - Active Directory and, 143*
  - advantages, 141*
  - authentication process, 142-143*
  - clients, 141*
  - history of, 142*
  - KDC (Key Distribution Center), 141*
  - keys, 141*
  - limitations, 146*
  - policies, 144*
  - preauthentication, 143*
  - servers, 141*
  - Windows 2000 interoperability, 145*
- PKI (public key infrastructure)
  - advantages, 147*
  - CAs (certification authorities), 147*

- components, 146-147*
- SmartCards, 148*

### **user mode (MMC), 5**

### **user policy, 181-182**

### **user principal names (UPNs), 43, 115**

### **users**

#### **accounts**

- copying, 112*
- creating, 111*
- deleting, 113*
- disabling, 114*
- editing properties, 113*
- enabling, 114*
- finding, 116*
- group memberships, 115*
- moving, 116*
- passwords, 114-115*
- predefined accounts, 110*
- renaming, 113-114*
- UPN (user principal name) suffixes, 115*

#### **groups**

- adding members to, 131*
- converting group type, 129*
- creating, 128*
- deleting, 129*
- distribution groups, 120*
- editing properties, 130*
- effect of domain mode, 127*
- finding, 130*
- nesting, 127-128*
- removing members from, 131-132*
- renaming, 132*
- replication conflicts, 132-133*

- scope and replication traffic, 126-130*
- security groups, 120-126*
- home directories, 119
- permissions, 157, 168
- profiles
  - advantages, 117*
  - local, 117*
  - mandatory, 118*
  - roaming, 117-118*
- rights, 108, 157
- special policies, 194
- USNs (Update Sequence Numbers), 243**
- UTF character support, 71, 87**
- utilities**
  - ADSI (Active Directory Service Interface), 47-48, 314
  - Backup, 280-281
    - creating backups, 282-283*
    - scheduling backups, 283-284*
    - starting, 281*
  - CSVDE (Comma-Separated Value Directory Exchange), 314-316
  - dsastat, 249
  - Event Viewer, 309-310
  - ipconfig, 230
  - LDIFDE (LDAP Data Interchange Format Directory Exchange), 316-318
  - mmc, 5
  - Movetree, 319
  - Network Monitor, 310
    - capturing network frames, 311-312*
    - installing, 311*

- Ntbackup, 281
- ntdsutil, 258-259, 267
  - authoritative restores, 291-292*
  - help, 268*
  - viewing FSMO roles, 269-270*
- Performance Logs and Alerts, 301
  - alerts, 305-307*
  - counter logs, 302-303*
  - trace logs, 303-305*
- Repadmin, 294
- System Monitor, 295-296
  - adding counters, 297-298*
  - Database object, 299-300*
  - deleting counters, 298*
  - loading and unloading counters, 299*
  - NTDS object, 296-297*
  - returning counter information, 299*
  - selecting counters, 300-301*
  - starting, 295*
- Task Manager, 307-308

## V

### -v parameter

- CSVDE (Comma-Separated Value Directory Exchange), 315
- LDIFDE (LDAP Data Interchange Format Directory Exchange), 317

## verifying

- authoritative restores, 294
- nonauthoritative restores, 289
- Schema Admins
  - membership, 215-216
- schema modifications, 221-222
- trusts, 68

## viewing

- ACEs (Access Control Entries), 156
- FSMO (Flexible Single-Master Operation)
  - roles, 269-270

## W

### Web browsers, Internet Explorer, 192

### Weight field (SRV records), 74

### Windows 2000

- compared to Windows NT 4.0, 17
  - administrative differences, 18-19*
  - logical differences, 17-18*
  - physical differences, 18*
- domains. *See* domains
- Kerberos interoperability, 145
- Windows 2000 DNS as primary DNS, 91-92

### Windows Installer packages, 185

### Windows Internet Name Service (WINS), 97-98

**Windows NT 4.0**

compared to Windows

2000, 17

*administrative*

*differences, 18-19*

*logical differences,*

*17-18*

*physical*

*differences, 18*

domains, migrating to

Windows 2000, 62

**Windows Script Host (WSH), 188****Windows.adm template, 184****Winnt.adm template, 184****WINS (Windows Internet Name Service), 87, 97-98****WINSR records, 87****wizards**

Add Filter Wizard,  
150-151

Backup Wizard, 282-283

Delegation of Control,  
230

Delegation of Control  
Wizard, 198

Filter Action Wizard, 153

Installation Wizard, 258

Remote Installation

Services Setup Wizard,  
190

**Write permission, 161****WSH (Windows Script Host), 188****X-Y-Z****X.500, 31****X.509 certificates, 10-11****y parameter (LDIFDE), 318****.zap filename extension, 185****zone transfers, 78**

between AD and BIND,  
94-95

full, 79-80

incremental, 80

**zones (DNS), 75-76, 83**

adding

*primary zones, 83*

*reverse lookup zones,*  
*83-84*

converting

*integrated zones to*

*standard zones, 86*

*standard zones to*

*integrated zones, 85*

deleting, 84

forward lookup zones, 76

integrated zones

*converting to standard*

*zones, 86*

*creating, 84*

*deleting, 84*

reverse lookup zones, 76

troubleshooting, 86-87

zone files, 75-76

zone transfers, 78

*between AD and*

*BIND, 94-95*

*full, 79-80*

*incremental, 80*