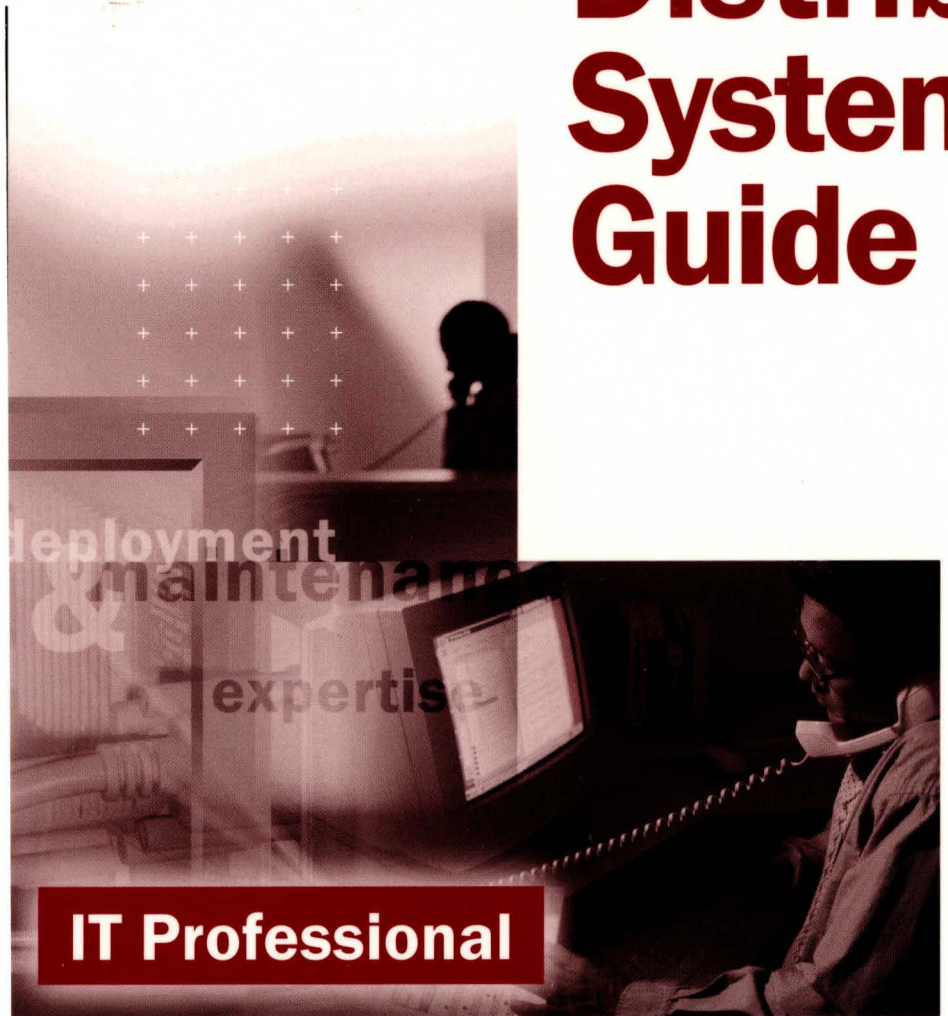
Microsoft Windows

Microsoft®

Microsoft® Windows® 2000 Server

Distributed Systems Guide



IT Professional

Microsoft®

Microsoft® Windows® 2000 Server Distributed Systems Guide

PUBLISHED BY
Microsoft Press
A Division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Library of Congress Cataloging-in-Publication Data
Microsoft Windows 2000 Server Resource Kit / Microsoft Corporation.
p. cm.
ISBN 1-57231-805-8
1. Microsoft Windows 2000 Server. 2. Operating systems (Computers). I. Microsoft Corporation.
QA76.76.O63 M5241328 2000
005.4'4769--dc21 99-045616

Printed and bound in the United States of America.

1 2 3 4 5 6 7 8 9 WCWC 5 4 3 2 1 0

Distributed in Canada by Penguin Books Canada Limited.

A CIP catalogue record for this book is available from the British Library.

Microsoft Press books are available through booksellers and distributors worldwide. For further information about international editions, contact your local Microsoft Corporation office or contact Microsoft Press International directly at fax (425) 936-7329. Visit our Web site at mspress.microsoft.com.

Acquisitions Editor: Juliana Aldous
Project Editor: Maureen Williams Zimmerman Part No. 097-0001948

Thank you to those who contributed to this book:

Department Managers: Paul Goode, Ken Western
Documentation Managers: Laura Burris, Martin DelRe, Peggy Etchevers
Resource Kit Program Managers: Chris Hallum, Martin Holladay, Louis Kahn, Ryan Marshall, Paul Sutton

Distributed Systems Guide

Technical Writing Leads: Don Lundman, David Stern
Writers: Laura Burris, Eric Carmel, Jim Glynn, Randy Grandle, Bill Gruber Mary Hillman, Heidi Johnson, James Klima, Don Lundman, Kathleen Norton Shira Paul, Brian Roberts, Jason Rush, Alan Spring, David Stern, Jeff Thomas

Editing Leads: Deborah Annan, Jennifer Hendrix, Kate O'Leary
Book Editing Lead: Sigrid Strom
Developmental Editors: Vincent Abella, Kevin Bentley, Diana Boyle Christopher Clements, Sandra Faucett, Chris McKitterick, Gary Moore Kate O'Leary, Sigrid Strom, Todd Young
Copy Editors: Kate McLaughlin, Mary Rose Sliwoski, Scott Somohano, Debbie Uyeshiro
Glossary: Daniel Bell

Resource Kit Tools Software Developers: Dan Grube, Michael Hawkins, Darryl Wood, Zeyong Xu
Documentation Tools Software Developers: Amy Buck, Tom Carey, Ryan Farber, Mark Pengra, Fred Taub

Production Leads: Sandy Dean, Jane Dow, Keri Grassl, Jason Hershey
Production Specialists: Michael Faber, Dani McIntyre, Lori Robinson

Indexing Leads: Jane Dow, Veronica Maier
Indexers: Lee Ross, Tony Ross

Lead Graphic Designer: Flora Goldthwaite
Designers: Chris Blanton, Siamack Sahafi
Art Production: Blaine Dollard, Jenna Kiter, Gabriel Varela

Test Lead: Jonathan Fricke
Testers: Brian Klauber, Jeremy Sullivan

Windows 2000 Lab Manager: Edward Lafferty
Administrators: Deborah Jay, Grant Mericle, Dave Meyer, Dean Prince, Robert Thingwold, Luke Walker, Joel Wingert, Frank Zamarron
Lab Partners: Cisco Systems, Inc., Compaq, Inc., Hewlett-Packard Corporation, Intel Corporation

A special thanks to the following technical experts who contributed to and supported this effort:
Steven Adler, Mohammad Shabbir Alam, Hilal Al-Hilali, Anoop Anantha, Johnson Apacible, Elena Apreutesei, Michael Armijo, Gigel Avram, Rudolph Balaz, Greg Baribault, Zach Beers, Colin Brace, John Brezak, Eric Brown, Mark Brown, Peter Brundrett, Marius Bulau, Steven Burns, Charlie Chase, Michael Cherry, Behrooz Chitsaz, Miguel Claudio, John Claugherty, Arren Conner, Paul Darcy, Michael Dennis, Chuck Dermody, Tony Donno, Bo Downey, Michael Emanuel, Levon Esibov, Cameron Etezadi, Scott Field, Carl Fischer, Eric Fitzgerald, Eric Flo, Michele Freed, Trevor Freeman, Billy Fuller, Praerit Garg, Andrew Goodsell, Jay Graham, Robert Gu, Arobinda Gupta, Don Hacherl, Andy Harjanto, Jim Harper, Richard Harrington, Vic Heller, Anne Hopkins, Michael Howard, Khushru Irani, Billy Jack, Marc Jacobs, Rick Johnson, Nikhil Joshi, Steven Judd, Stuart Kwan, Brian LaMacchia, Paul Leach, William Lees, Cheen Liao, David S. Loudon, Martin Lucas, Matt Lyons, Jim Masson, Rui Maximo, Drew McDaniel, Denise Y. Miller, Mark Myers, Arun Nanda, Jon Newman, Kumar Pandit, Jee Fung Pang, Jeff Parham, Geoff Pease, Michael Primeaux, Bohdan Raciborski, Ajay Ramachandran, Balan Sethu Raman, Rich Randall, Robert Reichel, Jason Robarts, Cynda Rochester, Jaret Russell, Mohammed Samji, Susan Saranovich, Murli Satagopan, Eyal Schwartz, Joseph Seifert, Ullattil Shaji, Ron Sherrell, Brett Shirley, Dan Simon, Pete Skelly, Heinrich Smit, Alan Smith, Eric Stadter, Todd Stecher, Jonathan Stephens, Dave Straube, Chandana Surlu, Russ Surridge, Mike Swift, Kahren Tevosyan, Varadarajan Thiruvillamalai, Matt Thomlinson, Darol Timberlake, Cezar Ungureanasu, Cliff Van Dyke, BJ Whalen, Charlie Wickham, Tim Williams, Shaohua Yin, Kelvin Yiu, Gideon Yuval, Marios Zikos.

Contents

Part 4 Desktop Configuration Management

Introduction

Welcome to the *Microsoft® Windows® 2000 Server Resource Kit Distributed Systems Guide.*

The *Microsoft® Windows® 2000 Server Resource Kit* consists of seven volumes and a single compact disc (CD) containing tools, additional reference materials, and an online version of the books. Supplements to the *Windows 2000 Server Resource Kit* will be released as new information becomes available, and updates and information will be available on the Web on an ongoing basis.

The *Distributed Systems Guide* provides a conceptual, theoretical, functional, and practical view of the various technologies that make up the Microsoft® Windows® 2000 distributed systems. This guide provides in-depth technical information that encompasses four major areas: Active Directory™, distributed security, enterprise technologies, and desktop configuration management.

Document Conventions

The following style conventions and terminology are used throughout this guide.

Element	Meaning
bold font	Characters that you type exactly as shown, including commands and switches. User interface elements are also bold.
Italic font	Variables for which you supply a specific value. For example, *Filename.ext* could refer to any valid file name for the case in question.
`Monospace font`	Code samples.
`%SystemRoot%`	The folder in which Windows 2000 is installed.

Reader Alert	Meaning
Tip	Alerts you to supplementary information that is not essential to the completion of the task at hand.
Note	Alerts you to supplementary information.
Important	Alerts you to supplementary information that is essential to the completion of a task.
Caution	Alerts you to possible data loss, breaches of security, or other more serious problems.
Warning	Alerts you that failure to take or avoid a specific action might result in physical harm to you or to the hardware.

Resource Kit Compact Disc

The *Windows 2000 Server Resource Kit* companion CD includes a wide variety of tools and resources to help you work more efficiently with Windows 2000.

Note The tools on the CD are designed and tested for the U.S. version of Windows 2000. Use of these programs on other versions of Windows 2000 or on versions of Microsoft® Windows NT® can cause unpredictable results.

The *Resource Kit* companion CD contains the following:

Windows 2000 Server Resource Kit Online Books An HTML Help version of the print books. Use these books to find the same detailed information about Windows 2000 as is found in the print versions. Search across all of the books to find the most pertinent information to complete the task at hand.

Windows 2000 Server Resource Kit Tools and Tools Help Over 200 software tools, tools documentation, and other resources that harness the power of Windows 2000. Use these tools to manage Active Directory™, administer security features, work with the registry, automate recurring jobs, and many other important tasks. Use Tools Help documentation to discover and learn how to use these administrative tools.

Windows 2000 Resource Kit References A set of HTML Help references:

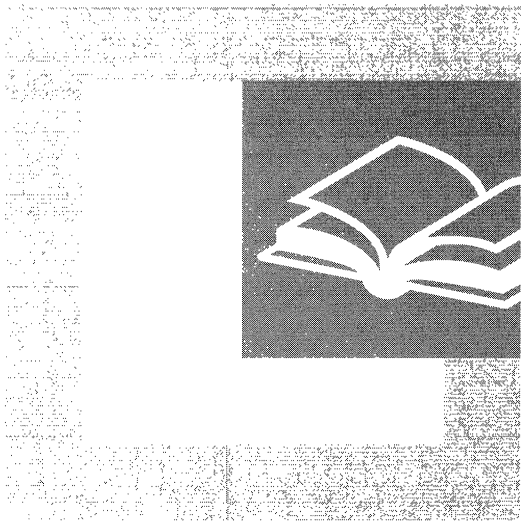
- **Error and Event Messages Help** contains most of the error and event messages generated by Windows 2000. With each message comes a detailed explanation and a suggested user action.
- **Technical Reference to the Registry** provides detailed descriptions of Windows 2000 registry content, such as the subtrees, keys, subkeys, and entries that advanced users want to know about, including many entries that cannot be changed by using Windows 2000 tools or programming interfaces.
- **Performance Counter Reference** describes all performance objects and counters provided for use with tools in the Performance snap-in of Windows 2000. Use this reference to learn how monitoring counter values can assist you in diagnosing problems or detecting bottlenecks in your system.
- **Group Policy Reference** provides detailed descriptions of the Group Policy settings in Windows 2000. These descriptions explain the effect of enabling, disabling, or not configuring each policy, as well as explanations of how related policies interact.

Resource Kit Support Policy

The software supplied in the *Windows 2000 Server Resource Kit* is not supported. Microsoft does not guarantee the performance of the *Windows 2000 Server Resource Kit* tools, response times for answering questions, or bug fixes to the tools. However, we do provide a way for customers who purchase the *Windows 2000 Server Resource Kit* to report bugs and receive possible fixes for their issues. You can do this by sending e-mail to rkinput@microsoft.com. This e-mail address is only for *Windows 2000 Server Resource Kit* related issues. For issues relating to the Windows 2000 operating system, please refer to the support information included with your product.

PART 1

Active Directory

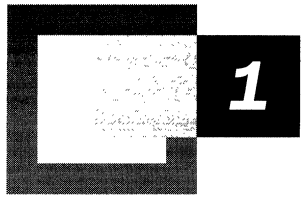


Active Directory™ is the core feature of distributed systems in Microsoft® Windows® 2000. Part 1 provides detailed technical information about Active Directory architecture and operation. This information is useful for network administrators when implementing or troubleshooting directory services.

In This Part

CHAPTER 1

Active Directory Logical Structure



Active Directory™, the directory service that is included with Microsoft® Windows® 2000, stores information about network objects and also implements the services that make this information available and usable to users, computers, and applications. The Domain Name System (DNS) hierarchical naming system and Windows 2000 trust relationships provide a consistent, logical structure that facilitates the organization of domains and domain resources in a predictable and useful way.

In This Chapter

Related Information in the Resource Kit

- For more information about Active Directory architecture and physical storage, see "Active Directory Data Storage" in this book.
- For more information about planning the DNS namespace, domain hierarchy, and organizational unit structure, see "Designing the Active Directory Structure" in the *Microsoft® Windows® 2000 Server Resource Kit Deployment Planning Guide*.
- For more information about standard DNS concepts and using Windows 2000 DNS server, see "Introduction to DNS" and "Windows 2000 DNS" in the *Microsoft® Windows® 2000 Server Resource Kit TCP/IP Core Networking Guide*.

Active Directory 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. A domain is an administrative boundary because administrative privileges do not extend to other domains. It is a security boundary because each domain has a security policy that extends to all security accounts within the domain. Active Directory stores information about objects in one or more domains.

Domains can be organized into parent-child relationships to form a hierarchy. A parent domain is the domain directly superior in the hierarchy to one or more subordinate, or child, domains. A child domain also can be the parent of one or more child domains, as shown in Figure 1.1.

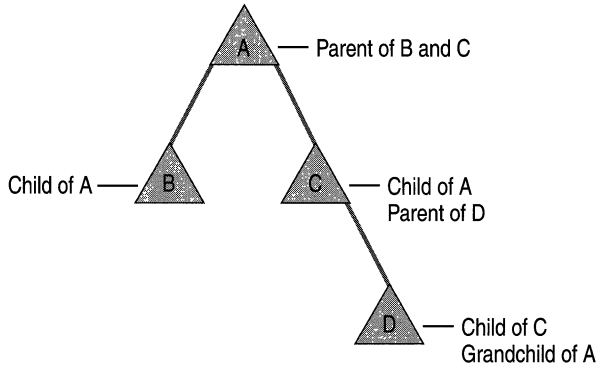


Figure 1.1 Example of a Domain Hierarchy

This hierarchical structure is a change from the flat domain structure of Microsoft® Windows NT® version 4.0 and Microsoft® Windows NT® version 3.51. The domain hierarchy of Windows 2000 allows you to search multiple domains in one query because each level of the hierarchy has information about the levels that are immediately above it and below it. This hierarchy information eliminates the need for you to know the location of a particular object in order for you to find it. In Windows NT 4.0 and earlier, you must know both the domain and the server where the object is located in order to find it.

For more information about Active Directory searches, see "Name Resolution in Active Directory" in this book.

Active Directory Domain Names

Windows 2000 uses DNS naming standards for hierarchical naming of Active Directory domains and computers. For this reason, domain and computer objects are part of both the DNS domain hierarchy and the Active Directory domain hierarchy. Although these domain hierarchies have identical names, they represent separate namespaces.

Note The domain hierarchy defines a namespace. A namespace is any bounded area in which standardized names can be used to symbolically represent some type of information (such as an object in a directory or an Internet Protocol [IP] address) and that can be resolved to the object itself. In each namespace, specific rules determine how names can be created and used. Some namespaces, such as the DNS namespace and the Active Directory namespace, are hierarchically structured and provide rules that allow the namespace to be partitioned. Other namespaces, such as the Network Basic Input/Output System (NetBIOS) namespace, are flat (unstructured) and cannot be partitioned.

The main function of DNS is to map user-readable computer names to computer-readable IP addresses. Thus, DNS defines a namespace for computer names that can be resolved to IP addresses, or vice versa. In Windows NT 4.0 and earlier, DNS names were not required; domains and computers used NetBIOS names, which were mapped to IP addresses by using the Windows Internet Name Service (WINS). Although DNS names are required for Windows 2000 domains and Windows 2000–based computers, NetBIOS names also are supported in Windows 2000 for interoperability with Windows NT 4.0 domains and with clients that are running Windows NT 4.0 or earlier, Microsoft® Windows® for Workgroups, Microsoft® Windows® 98, or Microsoft® Windows® 95.

Note WINS and NetBIOS are not required in an environment where computers run only Windows 2000, but WINS is required for interoperability between Windows 2000–based domain controllers, computers that are running earlier versions of Windows, and applications that depend on the NetBIOS namespace—for example, applications that call NetServerEnum and other "Net*" application programming interfaces (APIs) that depend on NetBIOS.

DNS Naming Conventions

DNS naming standards are used within Active Directory to provide support for industry-standardized mapping of DNS domain names to the IP addresses of computers. When a DNS server is given a computer name, it resolves the name to an IP address so that Transmission Control Protocol/Internet Protocol (TCP/IP)–based applications can communicate with the computer by its IP address.

In addition to being identified by computer name in DNS, Active Directory domain controllers are identified by specific services that they provide, such as Lightweight Directory Access Protocol (LDAP) servers, domain controllers, and Global Catalog servers. Therefore, when given the name of a domain and a service specification, a DNS server can locate a domain controller of that type within that domain. (For more information about how domain controllers are located, see "Name Resolution in Active Directory" in this book.)

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.

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

In the DNS naming structure, every portion of a DNS name that is separated by a period (.) represents a node in the DNS hierarchical tree structure and an Active Directory domain name in the Windows 2000 domain hierarchical tree structure.

Note In DNS, a node in the hierarchy can be a domain or a computer.

When you view a DNS name, a domain is any single portion of the DNS name that is separated from other parts of the DNS name by a period. For example, in the DNS domain name calif.noam.reskit.com, "calif," "noam," "reskit," and "com" each corresponds to a DNS domain. As illustrated in Figure 1.2, in Active Directory, the domain name calif.noam.reskit.com represents a hierarchy in which reskit.com is the root (topmost) domain, noam is a child domain of reskit.com (noam.reskit.com), and calif is a child domain of noam.reskit.com.

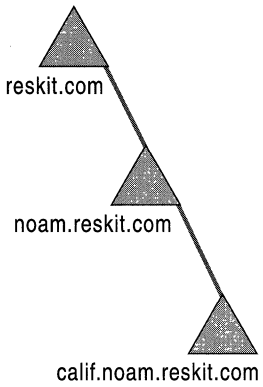


Figure 1.2 Example of an Active Directory Hierarchy with DNS Names

Note In reality, the .com domain is outside Active Directory, although it appears as part of the domain name. Domains such as .com, .org, and .edu, to name a few, are called top-level domains and are used on the Internet to classify organizations by type. (For more information about top-level DNS domains, see Windows 2000 Server Help.)

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

In DNS, the name that precedes a period is called a label. Reading the domain name from right to left, applications can correctly interpret the order of hierarchy for each label. And because two children of the same parent cannot have the same name, a domain name is always unique in the hierarchy. Each domain in the hierarchy defines its own portion of the overall namespace.

Whereas the Active Directory tree contains domains, the DNS tree contains both domains and computers. Therefore, the leftmost portion of a DNS name can also be a computer name.

Note For more information about Internet Standard host names, see the Request for Comments (RFC) link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. Search for RFC 1123. This host naming standard is recommended, but not required, for Active Directory domain names. (In DNS, computers are historically referred to as "hosts.")

For more information about DNS naming and functionality, see Windows 2000 Server Help, and see "Introduction to DNS" in the *TCP/IP Core Networking Guide*.

NetBIOS Domain Names

Windows 2000 provides support for applications that use the NetBIOS networking API and the flat NetBIOS names used by these applications. This support for non-DNS domain and computer names allows computers that are running Windows NT 4.0 and earlier, or those that are running Windows 95 or Windows 98, to identify Windows 2000 domains. For example, in a mixed-mode environment, Windows NT 4.0–based backup domain controllers (also known as "BDCs") recognize a specified Windows 2000–based domain controller as the primary domain controller (also known as a "PDC"). They use NetBIOS names to locate the primary domain controller; therefore, each Windows 2000–based domain controller must have a NetBIOS name to allow computers that are not running Windows 2000 to log on. Likewise, other server and workstation computers are recognized by NetBIOS names.

When you create a new domain during the Active Directory installation procedure, the system provides a default NetBIOS domain name that matches the leftmost label in the DNS domain name up to the first 15 bytes (NetBIOS names have a limit of 15 bytes). You can change this name during the procedure, but you cannot change it thereafter. When you name a stand-alone server or workstation computer, you provide a computer name that is used as the NetBIOS name and is concatenated with the domain name to form the full computer name.

Note An ASCII character is the equivalent of 1 byte. However, DNS host names are encoded in UTF-8 format and do not necessarily have only 1 byte per character.

For more information about DNS domain names, computer names, and host names, see "Windows 2000 DNS" in the *TCP/IP Core Networking Guide*. For more information about the Active Directory installation procedure, see "Name Resolution in Active Directory" in this book.

Active Directory and DNS

DNS is the de facto naming system for IP-based networks and the naming service that is used to locate computers on the Internet. Windows 2000 uses DNS to locate computers and domain controllers (that is, to locate Active Directory). A workstation or member server finds a domain controller by querying DNS. For this reason, installing or upgrading to Microsoft® Windows® 2000 Server requires that a DNS infrastructure is in place or is installed simultaneously.

Windows 2000 DNS server is included with Windows 2000 Server and Windows 2000 Advanced Server, and can be used to integrate DNS and Active Directory for ease of DNS management. Windows 2000 DNS server can be installed at the time you install Windows 2000 Server, at the time you install Active Directory, or manually after you have installed either of them.

For more information about installing and using Windows 2000 DNS server, see "Windows 2000 DNS" in the *TCP/IP Core Networking Guide*. For more information about installing Active Directory, see "Active Directory Data Storage" in this book.

DNS Hierarchy and Active Directory

Every Windows 2000 domain has a DNS name (for example, reskit.com), and every Windows 2000–based computer has a DNS name (for example, win2kserver.reskit.com). Thus, domains and computers are represented both as objects in Active Directory and as nodes in DNS.

Because DNS domains and Active Directory domains share identical domain names, it is easy to confuse their roles. The difference is that the two namespaces, although sharing an identical domain structure, store different data and, therefore, manage different objects: DNS stores zones and resource records, and Active Directory stores domains and domain objects. Both systems use a database to resolve names.

- DNS resolves domain names and computer names to resource records through requests received by DNS servers as DNS queries to the DNS database.
- Active Directory resolves domain object names to object records through requests that are received by domain controllers as LDAP search requests or as modify requests to the Active Directory database.

Thus, 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. For more information about using Windows 2000 DNS server, see "Windows 2000 DNS" in the *TCP/IP Core Networking Guide*.

DNS and the Internet

The Internet is a TCP/IP network. Every computer on the Internet or on any other TCP/IP network has an IP address. DNS locates TCP/IP hosts by resolving the computer names that end users understand to the IP addresses that computers understand. For example, the computer name DC1.reskit.com is resolved to the IP address 172.16.44.1. The addresses on the Internet are managed by using the globally distributed DNS database, but DNS can also be implemented locally to manage addresses within private TCP/IP networks. To create a presence on the Internet, it is recommended that an organization register its domain name with an Internet DNS registration agency so that other computers on the Internet can locate its servers, and vice versa. Registration is important to ensure that another organization does not use the same domain name.

Because DNS is the organizational structure of the Internet, the entire Internet is a single global namespace that is subdivided into a set of top-level domains that are then further subdivided into second-level domains. The root of the Internet domain namespace is managed by an Internet authority that is responsible for delegating administrative responsibility for the top-level domains of the DNS namespace and for registering second-level domain names. The top-level domains are the basic domain categories, such as commercial (.com), educational (.edu), and governmental (.gov). Second-level domains represent namespaces that are formally registered to individuals and to institutions and, thus, provide an Internet presence for these individuals and institutions. Their presence is supported by pointers in the relevant top-level domains to DNS servers that are authoritative for an individual's or organization's root domain; for example, name servers that are authoritative for the .com DNS database contain pointers to DNS name servers in the root domain of the private domain (reskit.com). These DNS pointers enable other domains to use the Internet to find the reskit.com domain. Similarly, DNS servers that are authoritative for the root domain for an individual or organization provide pointers to all DNS servers in child domains of the root domain, and so on down the hierarchy. DNS name servers on a private network likewise can contain pointers to Internet name servers if you want to be able to locate other domains on the Internet.

For more information about DNS and the Internet, see "Introduction to DNS" in the *TCP/IP Core Networking Guide*. For more information about top-level and second-level domains, see Windows 2000 Server Help.

Active Directory and the Internet

Active Directory can exist within the scope of the global Internet DNS namespace. When an Internet presence is required by an individual or organization, the Active Directory namespace is maintained as one or more hierarchical Windows 2000 domains beneath a root domain that is registered as a DNS namespace. Registration of individual and organizational root domain DNS names ensures the global uniqueness of all DNS names and provides for the assignment of network addresses that are recorded in the global DNS database. Registration of the DNS name for the root domain of the individual or organization also grants that individual or organization the authority to manage its own hierarchy of child domains, zones, and hosts within the root domain.

Note An organization might or might not choose to be part of the global Internet DNS namespace. However, even if the organization's root domain is not registered as an Internet DNS namespace, the DNS service is required to locate Windows 2000–based computers in general and Windows 2000–based domain controllers in particular.

For more information about DNS and DNS name servers, see Windows 2000 Server Help, and see "Introduction to DNS" in the *TCP/IP Core Networking Guide*.

DNS Host Names and Windows 2000 Computer Names

In Windows 2000, a computer name is a human-friendly name that maps to the IP network address by which the computer is physically located. In Windows NT 4.0 and earlier, a computer is identified primarily by a NetBIOS name—a name that is recognized by WINS, which maps the name to a static IP address or to an address configured dynamically by the Dynamic Host Configuration Protocol (DHCP). In earlier versions of Windows NT, if DNS was employed, NetBIOS applications queried the DNS namespace by appending a DNS domain name to the NetBIOS name.

To maintain compatibility with NetBIOS names that are used on computers that are not running Windows 2000, Windows 2000 incorporates the NetBIOS computer name as the DNS host name and the DNS domain name as the Primary DNS suffix. These two names are employed as distinct pieces of the full computer name (equivalent to the FQDN), thereby accommodating both DNS and, where needed, NetBIOS.

In Windows 2000, therefore, the full computer name has two parts:

- DNS host name

  The leftmost label in the fully qualified DNS name. The DNS host name identifies the computer's account that is stored in Active Directory. It is also the name of the Security Accounts Manager (SAM) local computer account on a workstation or member server (a computer that runs Windows NT Server or Windows 2000 Server but is not a domain controller). By default, the DNS host name is also used as the NetBIOS version of the computer name for compatibility with Windows NT 3.5*x* and Windows NT 4.0 domains and with computers that are running Windows 95 or Windows 98.

  Note The NetBIOS name and the DNS host name are not necessarily identical because NetBIOS names are limited to 15 bytes in length. If the length of the DNS host name is 15 bytes or less, by default the two names are identical. (For more information about DNS host names, see "Windows 2000 DNS" in the *TCP/IP Core Networking Guide*.)

- Primary DNS suffix domain name

  By default, the Windows domain to which the computer is joined. The default can be changed.

Figure 1.3 illustrates the form of a full computer name.

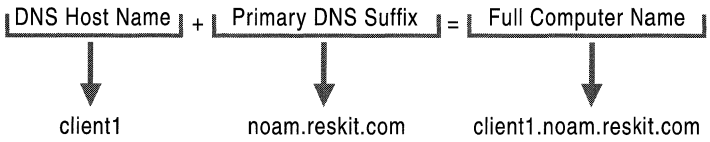


Figure 1.3 Components of a Full Computer Name

For more information about DNS, see "Introduction to DNS" and "Windows 2000 DNS" in the *TCP/IP Core Networking Guide*.

DNS Name Servers and Zones

DNS is a distributed database. DNS employs a client/server mechanism wherein the server portion, the name server, maintains the database, responds to queries, and processes updates. Windows 2000 DNS server can be installed on any computer that is running Windows 2000 Server or Windows 2000 Advanced Server. When installed on a domain controller, the DNS server is optimized for use with Active Directory. DNS clients, called "resolvers," are built-in components of modern TCP/IP protocol implementations and, therefore, are readily available to communicate with DNS servers.

DNS servers store the DNS database in zones. A zone is a contiguous partition of the DNS namespace that contains the resource records for the DNS domains that belong to the zone. When you configure DNS, you determine what these partitions are. Because Active Directory domain names have a 1:1 correspondence with DNS domains, it follows that DNS zones can include data about the computers in one or more Active Directory domains—that is, zones and domains do not have to have a 1:1 correspondence. One zone can encompass more than one domain. A DNS namespace contains domains, subdomains, and computers, which are also called "nodes."

DNS zones store records that represent computers, which also have objects in Active Directory. Figure 1.4 shows the relationship between the Active Directory object for a computer and the DNS node and host resource record for this same computer. The computer object and the host record are stored in different namespaces, but they represent the same physical computer.

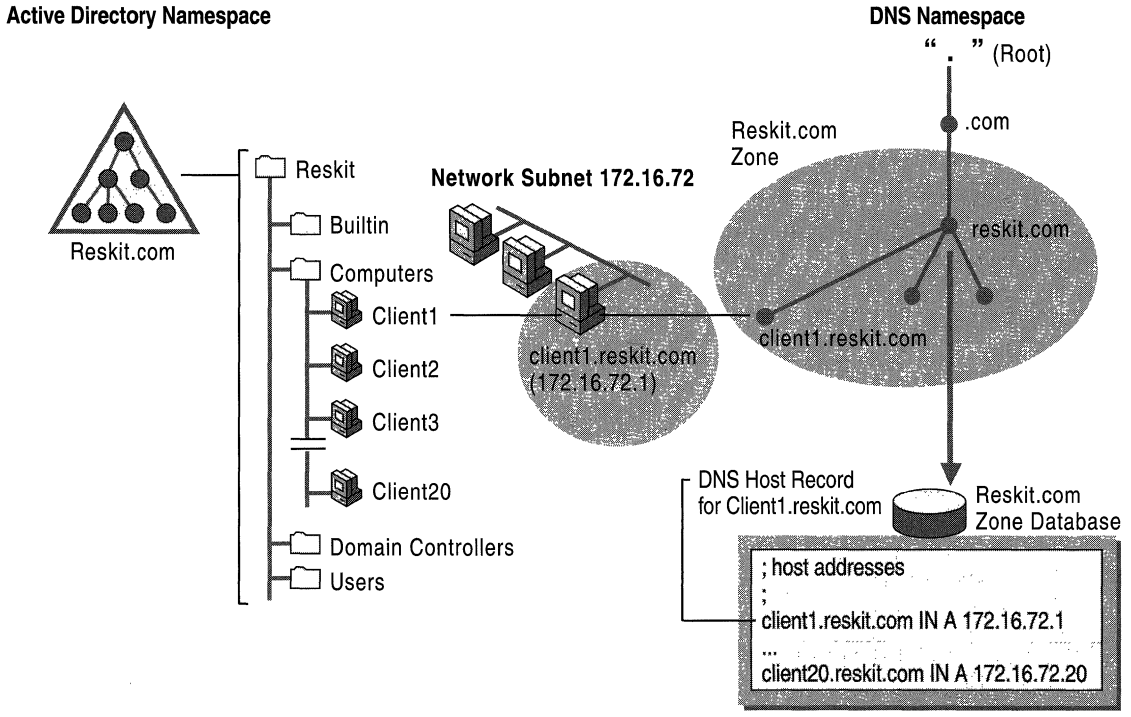


Figure 1.4 The Client1.reskit.com Computer Object in Active Directory and Its Host Record in DNS

Zone data identifies each host by DNS name and IP address; the data identifies computers specifically as domain controllers by linking the service they run (LDAP) to a computer name and IP address. Zone files also contain site information that makes it possible to locate domain controllers in the same site as the client and to locate domain controllers that have specific roles in the domain, such as a Global Catalog server or a Kerberos v5 server. Zone data can be stored in text files or in Active Directory. When zone data is stored in Active Directory, you configure the zone as an Active Directory–integrated zone.

Note It is not necessary to use Windows 2000 DNS as your DNS server in order to use Active Directory. However, the DNS server that you use must support service resource records (SRV records) in accordance with the Internet Engineering Task Force Internet Draft, "A DNS RR for Specifying the Location of Services (DNS SRV)," which updates RFC 2052. To perform automatic zone updates, the DNS server also should support the dynamic update protocol that is described in RFC 2136. For more information about IETF Internet Drafts, see the Internet Engineering Task Force (IETF) link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. Follow the links to Internet Drafts.

For more information about DNS, DNS zones, DNS resolvers, and DNS name servers, see "Introduction to DNS" in the *TCP/IP Core Networking Guide*. For more information about deciding which DNS server to use, see "Designing the Active Directory Structure" in the *Deployment Planning Guide*. For more information about Active Directory–integrated zones, see "Introduction to DNS" and "Windows 2000 DNS" in the *TCP/IP Core Networking Guide*. For more information about DNS service resource records, see "Name Resolution in Active Directory" in this book.

Active Directory–Integrated DNS

Active Directory–integrated DNS enables Active Directory storage and replication of DNS zone databases. Windows 2000 DNS server, the DNS server that is included with Windows 2000 Server, accommodates storing zone data in Active Directory. When you configure a computer as a DNS server, zones are usually stored as text files on name servers—that is, all of the zones required by DNS are stored in a text file on the server computer. These text files must be synchronized among DNS name servers by using a system that requires a separate replication topology and schedule called a zone transfer. However, if you use Active Directory–integrated DNS when you configure a domain controller as a DNS name server, zone data is stored as an Active Directory object and is replicated as part of domain replication.

Note Only DNS servers that run on domain controllers can load Active Directory–integrated zones.

To use DNS integration within Active Directory, assign the zone type **Active Directory-integrated** when you create the zone. (For more information about how to create zones, see Windows 2000 Server Help.) Objects that represent zone database records are created in the Microsoft DNS container within the System container (visible in the **Advanced Features** view in Active Directory Users and Computers), and the contents are replicated to all domain controllers in the domain. When you have Active Directory–integrated DNS zones, all Active Directory domain controllers that run Windows 2000 DNS server and are appropriately configured function as primary name servers.

When DNS data is stored in Active Directory, each DNS zone is an Active Directory container object (class *dnsZone*). The dnsZone object contains a DNS node object (class *dnsNode*) for every unique name within that zone. These unique names include the variations assigned to a specific host computer when it functions, for example, as a primary domain controller or as a Global Catalog server. The dnsNode object has a *dnsRecord* multivalue attribute that contains a value for every resource record that is associated with an object's name.

Figure 1.5 shows the relationship between the DNS nodes (dnsNode objects) in the Active Directory–integrated zone and the computer objects that were illustrated in Figure 1.4. When DNS is integrated with Active Directory, the DNS node for a computer corresponds to a dnsNode object in the directory. The resource records that are registered by the computer in DNS are represented as attribute values on the dnsNode object.

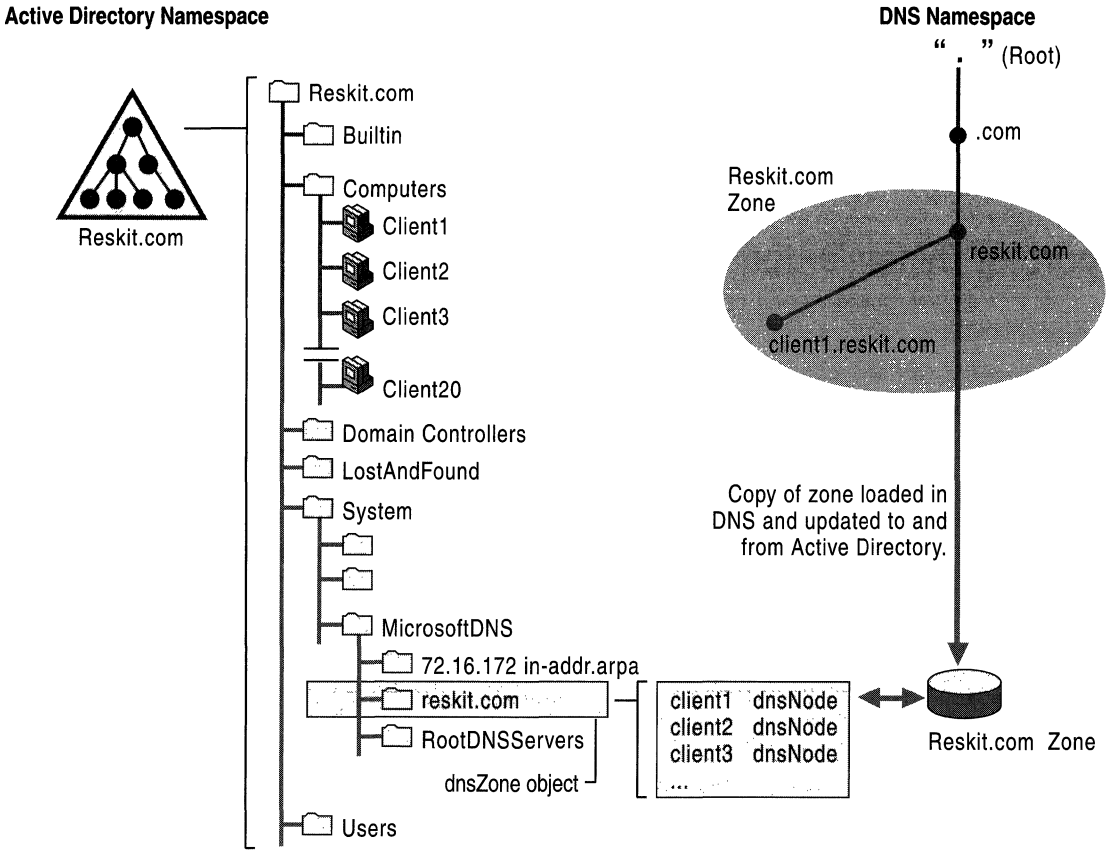


Figure 1.5 DNS Zones and DNS Nodes Stored in Active Directory

When other non-Windows 2000 DNS servers are already in place and Active Directory domains represent only part of the overall DNS namespace, standard zone transfer still can be used to synchronize zone data between Active Directory and other DNS servers (that is, using Active Directory–integrated primary DNS zones does not preclude other DNS implementations). For more information about using different DNS servers, see "Windows 2000 DNS" in the *TCP/IP Core Networking Guide*.

When Windows 2000 DNS server is installed on at least one domain controller and has Active Directory–integrated zones, the zone data is always replicated to every domain controller in the domain.

For more information about where to place DNS servers, see Windows 2000 Server Help and also "Windows 2000 DNS" in the *TCP/IP Core Networking Guide*. For more information about zone transfer and the location of zone data in Active Directory, see "Introduction to DNS" and "Windows 2000 DNS," respectively, in the *TCP/IP Core Networking Guide*. For more information about DNS objects in Active Directory, see Windows 2000 Server Help.

Support for Dynamic Updates

Windows 2000 DNS server supports the dynamic update protocol. This protocol is a standard that allows hosts to dynamically register their names in the DNS database, thus reducing administrative costs. When DNS zones are stored in Active Directory, DNS is configured by default to accept dynamic updates. The specification for dynamic update protocol is RFC 2136.

Secure dynamic update is a Windows 2000 feature that provides the additional benefit of making it possible to authenticate clients that dynamically register their host names in DNS. The server does not perform a dynamic update on behalf of a client unless it has authenticated the client in Active Directory and determined that the client has appropriate permissions to perform the dynamic update.

Note Secure dynamic update is available only with Active Directory–integrated zones.

For more information about dynamic updates and secure dynamic updates, see Windows 2000 Server Help and "Windows 2000 DNS" in the *TCP/IP Core Networking Guide*.

Tree and Forest Structure

In accordance with DNS naming standards, Active Directory domains are created in an inverted tree structure, with the root at the top. In addition, this Windows 2000 domain hierarchy is based on trust relationships—that is, the domains are linked by interdomain trust relationships.

Note The default interdomain trust relationships are created by the system during domain controller creation. The number of trust relationships that are required to connect n domains is n–1, whether the domains are linked in a single, contiguous parent-child hierarchy or they constitute two or more separate contiguous parent-child hierarchies.

When it is necessary for domains in the same organization to have different namespaces, create a separate tree for each namespace. In Windows 2000, the roots of trees are linked automatically by two-way, transitive trust relationships. Trees linked by trust relationships form a forest. A single tree that is related to no other trees constitutes a forest of one tree.

The tree structures for the entire Windows 2000 forest are stored in Active Directory in the form of parent-child and tree-root relationships. These relationships are stored as trust account objects (class *trustedDomain*) in the System container within a specific domain directory partition. For each domain in a forest, information about its connection to a parent domain (or, in the case of a tree root, to another tree root domain) is added to the configuration data that is replicated to every domain in the forest. Therefore, every domain controller in the forest has knowledge of the tree structure for the entire forest, including knowledge of the links between trees. You can view the tree structure in Active Directory Domain Tree Manager.

For more information about configuration data, see "Active Directory Data Storage" in this book.

Tree: Implementation of a Domain Hierarchy and DNS Namespace

A Windows 2000 tree is a DNS namespace: it has a single root domain and is built as a strict hierarchy; each domain below the root domain has exactly one superior, or parent, domain. The namespace created by this hierarchy, therefore, is contiguous—each level of the hierarchy is directly related to the level above it and to the level below it, if any, as illustrated in Figure 1.6.

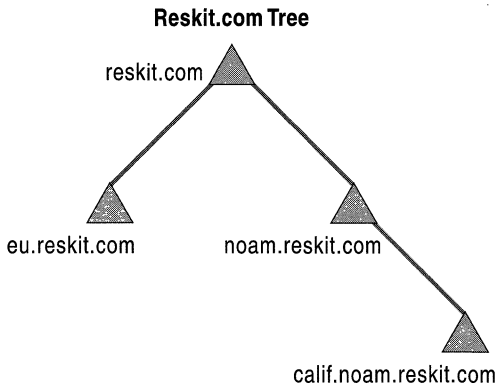


Figure 1.6 Example of a Contiguous Tree Hierarchy

In Windows 2000, the following rules determine the way that trees function in the namespace:

- A tree has exactly one name. The name of the tree is the DNS name of the domain at the root of the tree.
- The names of domains created beneath the root domain (child domains) are always contiguous with the name of the tree root domain.
- The DNS names of the child domains of a tree's root domain reflect this organization; therefore, the children of the root domain called "somedomain" are always children of that domain in the DNS namespace (for example, child1.somedomain, child2.somedomain, and so forth).

Child domains can represent geographical entities (for example, the United States and Europe), administrative entities within the organization (for example, sales and marketing departments), or other organization-specific boundaries, according to the needs of the organization. Domains are created below the root domain to minimize Active Directory replication and to provide a means for creating domain names that do not change. Changes in the overall domain architecture, such as domain collapses and domain re-creation, create difficult and potentially IT-intensive support requirements. A good namespace design should be capable of withstanding company reorganizations without the need to restructure the existing domain hierarchy.

Note Administrative privileges do not extend from parent domains to child domains. Privileges must be granted explicitly for each domain.

For more information about namespace design and the rationale for naming the root domain and creating child domains, see "Designing the Active Directory Structure" and "Determining Domain Migration Strategies" in the *Deployment Planning Guide*. For more information about administrative privileges, see "Authentication" and "Access Control" in this book.

Forest: Implementation of All Trees

A forest is a collection of one or more Windows 2000 Active Directory trees, organized as peers and connected by two-way, transitive trust relationships. A single domain constitutes a tree of one domain, and a single tree constitutes a forest of one tree. Thus, a forest is synonymous with Active Directory—that is, the set of all directory partitions in a particular directory service instance (which includes all domains and all configuration and schema information) makes up a forest.

Trees in the same forest do not form a contiguous namespace. They form a noncontiguous namespace that is based on different DNS root domain names. However, trees in a forest share a common directory schema, configuration, and Global Catalog. This sharing of common schema and configuration data, in addition to trust relationships between their roots, distinguishes a forest from a set of unrelated trees. Although the roots of the separate trees have names that are not contiguous with each other, the trees share a single overall namespace because names of objects can still be resolved by the same Active Directory. A forest exists as a set of cross-reference objects and trust relationships that are known to the member trees. Transitive trusts at the root domain of each namespace provide mutual access to resources. (For more information about cross-reference objects, see "Name Resolution in Active Directory" in this book.)

Important Tree and forest hierarchies are specific to Windows 2000 domains. A Windows NT 4.0 domain that is configured to trust or to be trusted by a Windows 2000 domain is not part of the Windows 2000 forest to which the Windows 2000 domain belongs.

The forest structure provides companies with the option of constructing their enterprise from separate, distinct, noncontiguous namespaces. Having a separate namespace is desirable under some conditions where, for example, an acquired company's namespace should remain intact. If you have business units with distinct DNS names, you can create additional trees to accommodate the names. An example of this type of organization is shown in Figure 1.7.

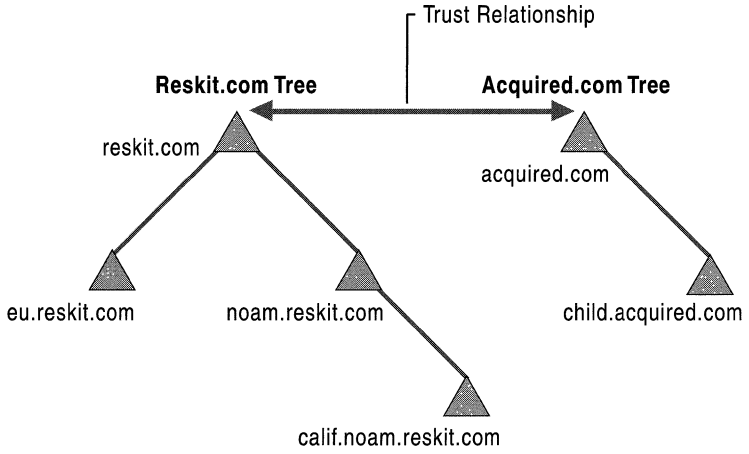


Figure 1.7 Example of a Forest That Has Two Trees

Domains within an Active Directory forest share a common directory schema, configuration information, and Global Catalog. They also have transitive trust relationships that allow users in each domain access to available resources in all other domains in the tree.

Note The directory schema and configuration data are shared because they are stored in separate logical directory partitions that are replicated to domain controllers in every domain in the forest. (For more information about directory partitions, see "Active Directory Data Storage" in this book.) The data relative to a particular domain is replicated only to domain controllers in the same domain. (For more information about replication, see "Active Directory Replication" in this book.) The Global Catalog is a domain controller that stores all objects of all domains in an Active Directory forest, which makes it possible to search for objects at the forest level rather than at the tree level.

For more information about the contents of Active Directory configuration, directory schema, and Global Catalog, see "Active Directory Data Storage" in this book. For more information about searching in Active Directory, see "Name Resolution in Active Directory" in this book.

Forest Root Domain

The first domain created in the forest is called the forest root domain. The forest root domain cannot be deleted, changed, or renamed. When you create a new tree, you specify the root domain of the initial tree, and a trust relationship is established between the root domain of the second tree and the forest root domain. If you create a third tree, a trust relationship is established between the root domain of the third tree and the forest root domain. Because a trust relationship is transitive and bidirectional, the root domain of the third tree also has a two-way trust relationship with the root domain of the second tree.

The distinguished name of the forest root domain is used to locate the configuration and schema directory partitions in the namespace. The distinguished names for the Configuration and Schema containers in Active Directory always show these containers as child objects in the forest root domain. For example, in the child domain noam.reskit.com, the distinguished name of the Configuration container is cn=configuration,dc=reskit,dc=com. The distinguished name of the Schema container is cn=schema,cn=configuration,dc=reskit,dc=com. However, this naming convention provides only a logical location for these containers. The containers do not exist as child objects of the forest root domain, nor is the schema directory partition actually a part of the configuration directory partition. They are separate directory partitions. Every domain controller in a forest stores a copy of the configuration and schema directory partitions, and every copy of these partitions has the same distinguished name on every domain controller.

When Active Directory is installed on a Windows 2000 Server–based computer, configuration and directory schema information is copied from the parent domain to the new server. Updates to configuration and directory schema information are replicated to all domain controllers throughout the forest. The distribution of this configuration and directory schema information ensures that each domain controller is aware of all other trust-related domains and of the replication topology, which makes finding and using resources in other domains possible. (For more information about finding information in Active Directory, see "Name Resolution in Active Directory" in this book.)

Note The Active Directory rootDSE is a figurative object that has no LDAP distinguished name; it is not an "entry" in the directory but is represented as a null distinguished name (" "). It does, however, have attributes and is known to LDAP as rootDSE. RootDSE is required by LDAP as an entry point to the directory. The distinction must be clear between this root—the set of attributes that LDAP uses to connect to a particular portion of the directory on a particular domain controller—and the root domain of the forest. In addition, both of these "roots" are distinct from the root of the DNS hierarchy, which is the empty space at the top of the namespace that is represented as a period (".") and that is required as an entry point to the DNS hierarchy.

For more information about rootDSE attributes and the directory tree, see "Active Directory Data Storage" in this book. For more information about the DNS root, see "Introduction to DNS" in the *TCP/IP Core Networking Guide*.

Trust Relationships

Active Directory provides security across multiple domains through interdomain trust relationships. When there are trust relationships between domains, the authentication mechanism for each domain trusts the authentication mechanism for all other trusted domains. If a user or application is authenticated by one domain, its authentication is accepted by all other domains that trust the authenticating domain. Users in a trusted domain have access to resources in the trusting domain, subject to the access controls that are applied in the trusting domain.

Note "Access to resources" in any discussion of trust relationships always assumes the limitations of access control. Trust relationships allow users and computers to be authenticated (to have their identity verified) by an authentication authority. Access control allows authenticated users to use the resources (files, folders, and virtual containers) that they are authorized to use and prohibits them from using (or even seeing) resources that they are not authorized to use. For more information about resource authorization, see "Access Control" in this book.

Transitive and Nontransitive Trust

In Windows NT 3.51 and Windows NT 4.0, trust relationships must be created explicitly in one direction. A two-way trust relationship is established by creating two one-way trust relationships. Domains can be connected by explicit one-way or two-way trust relationships for the purpose of enabling access to resources, but they are not necessarily related in any other way.

In Windows 2000, domains can be joined to a domain tree or forest, and each child domain has an automatic two-way trust relationship with the parent domain. This trust relationship is also transitive. Transitive trust means that the trust relationship extended to one domain is extended automatically to any other domain that is trusted by that domain. Transitive trust is applied automatically for all domains that are members of the domain tree or forest. Therefore, when a grandchild domain is created, the trust relationship between the parent and child domains is accepted by the grandchild domain, and vice versa. For example, if a user account is authenticated by the parent domain, the user has access to resources in the grandchild domain. Similarly, if the user is authenticated by a child domain, the user has access to resources in the parent domain, as well as in the grandparent domain.

The effect of transitive trust in Windows 2000 domains is that there is complete trust between all domains in an Active Directory forest—every domain has a transitive trust relationship with its parent domain, and every tree root domain has a transitive trust relationship with the forest root domain.

Note In Windows 2000, transitive trust relationships are always two-way trust relationships.

A nontransitive trust relationship can be created between Windows 2000 domains when a transitive trust relationship is not appropriate, but this trust relationship must be created explicitly. It can be created, for example, between two Windows 2000 domains that are not in the same forest.

A trust relationship between a Windows 2000 domain and a Windows NT 4.0 domain is always a nontransitive trust relationship. If one of these domains is an account domain and the other is a resource domain, the trust relationship is usually created as a one-way trust relationship. If there are user accounts in both domains, two one-way trust relationships can be created between them.

The trust relationship between two domains—whether one-way or two-way, transitive or nontransitive—is stored as an interdomain trust account object in Active Directory.

For more information about the nature and management of interdomain trust objects, see "Authentication" in this book. For more information about mixed-mode trust relationships, see "Determining Domain Migration Strategies" in the *Deployment Planning Guide*.

Direction of Trust

In describing trust relationships, arrows illustrate the direction of trust between domains as follows:

- If B is the trusting domain and A is the trusted domain, B-->A indicates that domain B trusts domain A. (The same trust relationship can be illustrated as A<--B, that is, A is trusted by B.)
- When domain B trusts domain A (B-->A), users with accounts in domain A can be authenticated for access to resources in domain B. However, users with accounts in domain B are not trusted to be authenticated for access to resources in domain A.

A hierarchy of Windows 2000 domains is implemented by trust relationships between domains. The direction of the trust relationship between a parent domain and its child domain in Active Directory is two-way (A<---->B), but it has the following restrictions:

- The parent-child relationship between two domains in a domain tree is defined by a subordinate name relationship. For example, noam.reskit.com is a child of reskit.com, but noam.com is not a child of reskit.com. A parent-child trust relationship requires both a parent-child relationship and a direction of trust, as follows: Domain A can be specified as the parent of domain B only if B-->A and B is a subordinate name of A.
- When a new domain joins a domain tree as a child, a parent-child trust relationship is defined automatically that establishes a two-way, transitive trust relationship.

Note Automatic configuration of replication topology requires that all parent-child trust relationships within the forest are bidirectional and transitive.

The use of two-way, transitive trust relationships reduces management time because it decreases by more than half the number of trust relationships that must be managed, as the diagram in Figure 1.8 illustrates.

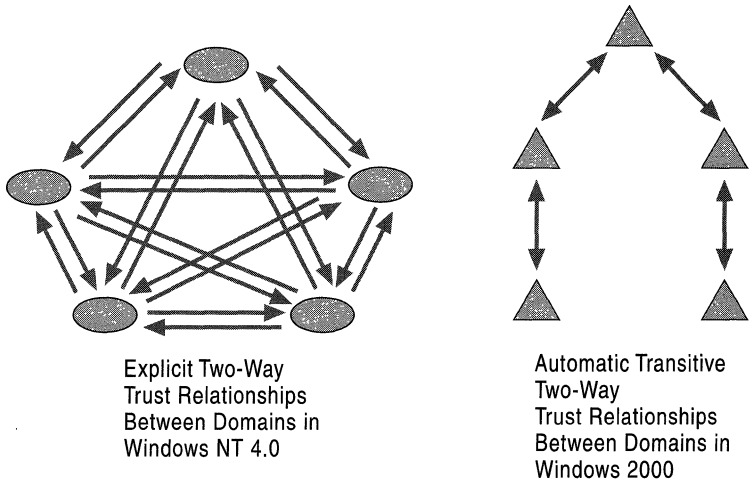


Figure 1.8 Comparison of Two-way Trust Relationships in Windows NT 4.0 and Windows 2000

Authentication Protocols

Windows 2000 authenticates users and applications by using one of two protocols: the Kerberos v5 authentication protocol or the NTLM authentication protocol. The protocol to be used is determined by the capabilities of the client and the server. If the client does not recognize the Kerberos protocol (for example, a computer that is running Windows NT 3.51 or Windows NT 4.0), authentication occurs by using the NTLM challenge-response protocol. Conversely, if the resource server does not support Kerberos authentication, the client uses NTLM to authenticate to the server.

The Kerberos v5 protocol is the default protocol for network authentication on computers that are running Windows 2000. The NTLM protocol is the default for network authentication in Windows NT 4.0 and for Windows 95–based and Windows 98–based computers that are running Distributed Systems Client. It is retained in Windows 2000 for compatibility with previous versions of Windows-based clients and servers. But the protocol of choice in Windows 2000, when there is a choice, is the Kerberos protocol.

In Windows 2000 domains, the Kerberos v5 authentication protocol is used to authenticate logons when all of the following conditions are true:

- The user who is logging on uses a security account in a Windows 2000 domain.
- The computer that is being logged on to is a Windows 2000–based computer.
- The computer that is being logged on to is joined to a Windows 2000 domain.
- The computer account and the user account are in the same forest.

For any other combination of conditions, such as a computer that is running Windows NT 3.51 or Windows NT 4.0, a user who has an account in a Windows NT 3.51 or Windows NT 4.0 domain, or a domain that is a Windows NT 3.51 or Windows NT 4.0 domain, the NTLM protocol is used to authenticate logons.

The essential differences between the two protocols are these:

- When the NTLM protocol is used, the server must contact a domain authentication service on a domain controller to verify the client credentials. A server authenticates a client by forwarding the client credentials to a domain controller in the client account domain.
- When the Kerberos protocol is used, the server does not have to contact the domain controller. A client gets a ticket for a server by requesting one from a domain controller in the server account domain; the server validates the ticket without consulting any other authority.

For more information about the Kerberos v5 and NTLM authentication protocols, see "Authentication" in this book.

Trust Path

A trust path is defined by a series of trust links from one domain to another domain for passing authentication requests. For example, when a user makes a request for information from a server in a domain other than the domain in which the user is currently logged on, the server must be able to authenticate the user. Before authentication can occur, Windows security must determine whether the domain that is requested (the domain in which the contacted server is located) has a trust relationship with the logon domain of the user account. To make this determination, the Windows 2000 security system computes a trust path between the domain controller for the server that receives the request and a domain controller in the requesting user's account domain.

In the Windows 2000 distributed security model, every workstation and server has a direct trust path to a domain controller in the domain in which it is located. The trust path is implemented by the Net Logon service through an authenticated remote procedure call (RPC) connection to the trusted domain authority—namely, the domain controller. In addition, a secure channel extends to other Windows 2000 domains through interdomain trust relationships. The secure channel is used to obtain and verify security information, including security identifiers (SIDs) for users and groups.

Every Windows 2000 domain has knowledge of all other domains in the forest, as well as of all external domains that it directly trusts or that trust it. By using this information, a domain controller builds the shortest path for authentication. When building the trust path, each domain is first checked to see whether it is the requested domain and then checked for any shortcut trust relationships to the requested domain. If none of these conditions exists, the request is passed ("referred") to the parent domain (because by definition, the child domain trusts the parent domain). However, if there is no transitive trust relationship, the request is denied. If the request is passed all the way to the root domain, it can be referred to a different domain tree root in the forest or, if an external trust relationship exists, to a domain in a different forest.

Note A shortcut trust relationship is a trust relationship that is created explicitly to shorten the trust path between domains that are in the same forest.

If the authentication request is referred, a path is computed for either NTLM pass-through authentication or for a Kerberos referral by using the information about the tree and current shortcut trust relationships to find the path to the destination domain. In this computation, shortcut trust relationships play the role of circumventing the higher domains in the hierarchy. At each level of the tree, a check is made of the shortcut trust relationships that might exist. If one is found to the destination domain, the next domain in the tree does not have to be checked.

Processing Authentication Referrals

When a request for authentication is referred, trust relationships must be taken into account with respect to their direction and whether they are transitive or nontransitive. The two Windows authentication protocols process referrals differently.

Kerberos v5 Authentication Protocol

If the client uses the Kerberos v5 protocol, the client requests a ticket to the server in the target domain from a domain controller in its account domain. The Kerberos Key Distribution Center (KDC) is a service that acts as a trusted intermediary between a client and server; it provides a session key that enables the two parties to authenticate each other. If the target domain is different from the current domain, the KDC uses the following logic to determine whether an authentication request can be referred:

- Is the current domain trusted directly by the domain of the server that is being requested?
  - If yes, send the client a referral to the requested domain.
  - If no, go to the next step.
- Is there a transitive trust relationship between the current domain and the next domain on the trust path?
  - If yes, send the client a referral to the next domain on the trust path.
  - If no, send the client a logon-denied message.

NTLM Authentication Protocol

If the client uses the NTLM authentication protocol, the initial request for authentication goes directly from the client to the resource server in the target domain. This server sends the user's security credentials to a domain controller in its computer account domain. This domain controller checks the user account against its security accounts database. If the account does not exist, the domain controller uses the following logic to perform pass-through authentication, forward the request, or deny the request:

- Does the current domain have a direct trust relationship with the user's domain?
  - If yes, the domain controller sends the credentials of the client to a domain controller in the user's domain for pass-through authentication.
  - If no, go to the next step.
- Does the current domain have a transitive trust relationship with the user's domain?
  - If yes, pass the authentication request on to the next domain in the trust path. This domain controller begins the process again by checking the user's credentials against its security accounts database.
  - If no, send the client a logon-denied message.

For more information about NTLM authentication and Kerberos v5 authentication mechanisms, see "Authentication" in this book. For more information about cross-reference objects in the Configuration container, see "Name Resolution in Active Directory" in this book.

Types of Trust Relationships

The following types of trust relationships can be established with Windows 2000 domains:

Tree-Root Trust Relationship. A tree-root trust relationship is the trust relationship that is established when you add a new tree to a forest. The Active Directory installation process automatically creates a trust relationship between the domain you are creating (the new tree root) and the forest root domain. A tree-root trust relationship has the following restrictions:

- It can be set up only between the roots of two trees in the same forest.
- It must be transitive and two-way.

Parent-Child Trust Relationship. A parent-child trust relationship is the trust relationship that is established when you create a new domain in a tree. The Active Directory installation process automatically creates a trust relationship between the new domain and the domain that immediately precedes it in the namespace hierarchy (for example, noam.reskit.com is created as the child of reskit.com). The parent-child trust relationship has the following characteristics:

- It can exist only between two domains in the same tree and namespace.
- The parent domain is always trusted by the child domain.
- It must be transitive and two-way in Windows 2000. The bidirectional nature of transitive trust relationships allows the global directory information in Windows 2000 to replicate throughout the hierarchy.

Shortcut Trust Relationship. A shortcut trust relationship (also called a cross-link trust relationship) is a manually created trust relationship that improves the efficiency of remote logons by shortening the trust path. If users in domain A often need to gain access to resources in domain C, you might want to create a direct link through a shortcut trust relationship so that domain B can be bypassed in the trust path. A shortcut trust relationship has the following characteristics:

- It can be established between any two domains in the same forest.
- It must be set up manually in each direction.
- It must be transitive.

External Trust Relationship. An external trust relationship is a manually created trust relationship between Windows 2000 domains that are in different forests or between a Windows 2000 domain and a domain whose domain controller is running Windows NT 4.0 or earlier. An external trust relationship has the following characteristics:

- It is one-way.
- It must be set up manually in each direction to establish a two-way external trust relationship.
- It is nontransitive.

Non-Windows Kerberos Realm Trust Relationship. A trust relationship that can be established between a non-Windows Kerberos realm and a Windows 2000 domain. This trust relationship allows cross-platform interoperability with security services based on other Kerberos v5 implementations. (For more information about non-Windows Kerberos interoperability and setting up trust relationships between Windows 2000 domains and non-Windows Kerberos realms, see the Microsoft Windows 2000 Server link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. Follow the links to Deployment and then Security Services.)

The non-Windows Kerberos realm trust relationship has the following characteristics:

- It is used only by the Kerberos v5 authentication protocol, not by NTLM or other authentication protocols.
- It is one-way by default. To establish a two-way trust relationship, a one-way trust relationship in each direction must be set up manually.
- It is nontransitive by default.
- When the direction of trust is from a non-Windows Kerberos realm to a Windows 2000 domain, the non-Windows Kerberos realm trusts all security principals in the Windows 2000 domain.

- When the direction of trust is from a Windows 2000 domain to a non-Windows Kerberos realm, account mappings in Active Directory are used to map a foreign Kerberos identity in a trusted non-Windows Kerberos realm to a local account identity in a Windows 2000 domain. The Windows 2000 domain uses only the account to which the non-Windows principal is mapped to evaluate access to domain objects that have security descriptors. This identity is required because non-Windows Kerberos tickets do not contain all of the authorization data that is needed for Windows 2000. All such Windows 2000 proxy accounts can be used in groups and on access control lists (ACLs) to control access on behalf of the non-Windows security principal.

  MIT account mappings are managed by using Active Directory Users and Computers. (For more information about MIT Kerberos interoperability and managing foreign Kerberos identities, see the Microsoft Windows 2000 Server link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. Follow the links to Deployment and then Security Services.)

Note If you create a non-Windows Kerberos realm trust relationship by using Active Directory Domains and Trusts, the trust is one-way and nontransitive. You can use the Netdom tool (Netdom.exe) to establish two-way, transitive, non-Windows Kerberos realm trust relationships. You also can use Netdom to modify a non-Windows Kerberos realm trust relationship that you created in Active Directory Domains and Trusts; you can change the trust relationship from non-transitive to transitive by using the /Transitive:yes option in Netdom. (To use Netdom, install the Support Tools that are located in the Support\Tools folder on the Windows 2000 Server operating system CD. To install the tools, double-click the **Setup** icon in that folder.) For more information about using Netdom to create non-Windows Kerberos realm trust relationships, see Windows® 2000 Support Tools Help.

Use Active Directory Domains and Trusts to manage trust relationships by using the properties of a domain object. The **Properties** page shows two lists; one shows the trusted domains (**Domains trusted by this domain**), and the other shows the trusting domains (**Domains that trust this domain**) for the current domain.

For more information about establishing trust relationships by using Active Directory Domains and Trusts, see Windows 2000 Server Help. For more information about planning trust relationships, see "Determining Domain Migration Strategies" in the *Deployment Planning Guide*.

Trust Relationships Between Windows 2000 and Windows NT 4.0 Domains

Windows 2000 and Windows NT 4.0 domains can trust each other so that users from either domain can authenticate in the other domain to gain access to resources, but users can do so only if explicit, one-way trust relationships have been created between the domains.

The following examples illustrate the effect of the direction of trust between a Windows 2000 domain and a Windows NT 4.0 domain.

- A (Windows 2000 domain) --> B (Windows NT 4.0 domain). This trust relationship indicates that users in domain B have access to resources in domain A but do not have access to resources in any other domain within the tree.
- B (Windows NT 4.0 domain) --> A (Windows 2000 domain). This trust indicates that only users in domain A (not users in other domains within the tree) have access to resources in domain B.

When a client views Windows 2000 trust relationships from a Windows NT 4.0–based computer, the list of trust relationships that is displayed depends on the type of domain to which the computer belongs:

- In a native-mode domain, the client sees a complete list of the domains in the forest.
- In a mixed-mode domain, the client sees only those domains that are trusted directly by the domain to which the client belongs. In a mixed-mode domain, the client can be using a Windows NT 4.0–based backup domain controller. To ensure consistent results, whether it uses a Windows NT 4.0–based domain controller or a Windows 2000–based domain controller, the same limited list of domains is presented to the client in both cases.

Mixed-Environment Scenario

Figure 1.9 illustrates a mixed environment of two Windows 2000 forests and a Windows NT 4.0 domain. In all, four separate namespaces are implemented: A.com, D.com, G.com, and F.

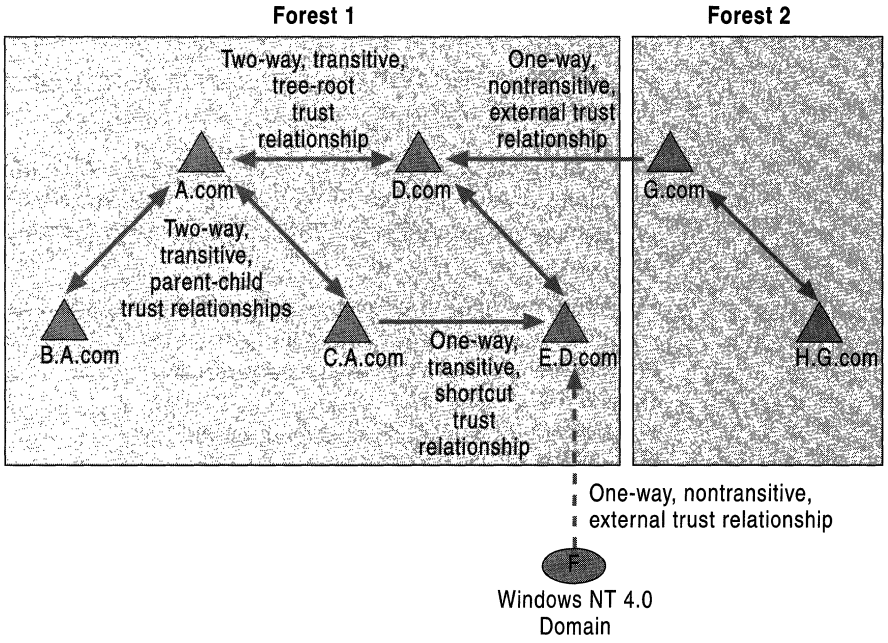


Figure 1.9 Mixed Environment of Two Forests and a Windows NT 4.0 Domain

The following conditions are represented in Figure 1.9:

- A.com and D.com are the roots of separate trees in forest 1. The two-way, transitive, tree-root trust between them provides complete trust between all domains in the two trees of forest 1.
- E.D.com uses resources in C.A.com for everyday business operations. To shorten the trust path between the two domains, C.A.com trusts E.D.com directly. This trust relationship serves only the purpose of shortening the trust path for authenticating E.D.com users to use resources in C.A.com. The path is shortened by cutting the number of hops required for authentication from three (E.D.com to D.com, D.com to A.com, and A.com to C.A.com) to one (E.D.com to C.A.com), which increases the speed of authentication.
- G.com is the root of a single tree that makes up forest 2. The two-way, transitive trust relationship between G.com and H.G.com allows both domains to use each others' resources.

- Domain G.com in forest 2 implements an explicit one-way external trust relationship with domain D.com in forest 1; users in domain D.com are trusted to use resources in domain G.com. Because the trust relationship is nontransitive, no other domains in forest 1 have access to resources in G.com, and D.com does not have access to resources in H.G.com.
- Domain F is a Windows NT 4.0 domain that provides support services to the users in E.D.com. This one-way nontransitive trust relationship does not extend to any other domains in forest 1.

Active Directory Objects

Active Directory objects represent the physical entities that make up a network. An object is an instance of storage of a class. A class is defined in the Active Directory schema as a specific set of mandatory and optional attributes—that is, an attribute can be present in an object in Active Directory only when that attribute is permitted by the object's class. Classes also contain rules that determine which classes of objects can be superior to (parents of) a particular object of the class. Each attribute is also defined in the directory schema. The attribute definitions determine the syntax for the values the attribute can have.

When you create an object in Active Directory, you provide values for the attributes of the object in its particular class, and you do so according to the rules of the directory schema. For example, when you create a user object, you provide alphanumeric values for the user's first and last names, the logon identifier, and perhaps other values, such as telephone number and address. You cannot create the user object successfully without providing acceptable values for the user name and logon name because these attributes are mandatory, according to the directory schema.

Applications that create or modify objects in Active Directory use the directory schema to determine what attributes the object must and might have, and what those attributes can look like in terms of data structures and syntax constraints. For this reason, the directory schema is maintained forest-wide so that all objects created in the directory conform to the same rules.

Objects are either container objects or leaf objects. A container object stores other objects, and, as such, it occupies a specific level in a subtree hierarchy. An object class is a container if at least one other class specifies it as a possible superior; thus, any object class defined in the schema can become a container. A leaf object does not store other objects, and, as such, it occupies the endpoint of a subtree.

For more information about how Active Directory objects are stored, see "Active Directory Data Storage" in this book. For more information about the directory schema, see "Active Directory Schema" in this book.

Object Naming

Active Directory is an LDAP-compliant directory service, which means that all access to directory objects occurs through LDAP. LDAP requires that names of directory objects be formed according to RFC 1779 and RFC 2247, which define the standard for object names in an LDAP directory service.

Distinguished Name

Objects are located within Active Directory domains according to a hierarchical path, which includes the labels of the Active Directory domain name and each level of container objects. The full path to the object is defined by the distinguished name (also known as a "DN"). The name of the object itself, separate from the path to the object, is defined by the relative distinguished name.

The distinguished name is unambiguous (identifies one object only) and unique (no other object in the directory has this name). By using the full path to an object, including the object name and all parent objects to the root of the domain, the distinguished name uniquely and unambiguously identifies an object within a domain hierarchy. It contains sufficient information for an LDAP client to retrieve the object's information from the directory.

For example, a user named James Smith works in the marketing department of a company as a promotions coordinator. Therefore, his user account is created in an organizational unit that stores the accounts for marketing department employees who are engaged in promotional activities. James Smith's user identifier is JSmith, and he works in the North American branch of the company. The root domain of the company is reskit.com, and the local domain is noam.reskit.com. The diagram in Figure 1.10 illustrates the components that make up the distinguished name of the user object JSmith in the noam.reskit.com domain.

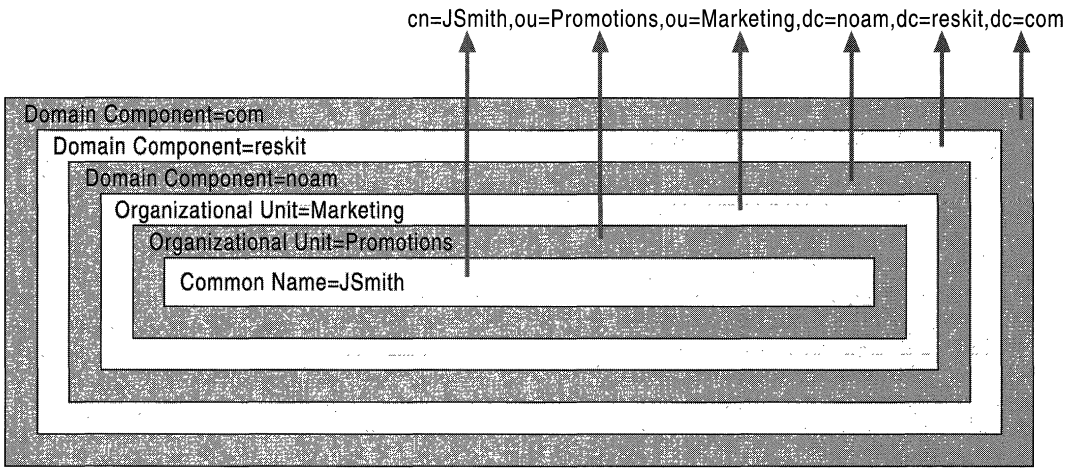


Figure 1.10 Distinguished Name for the User Object JSmith

Note Active Directory snap-in tools do not display the LDAP abbreviations for the naming attributes domain component (dc=), organizational unit (ou=), common name (cn=), and so forth. 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, as described later in this chapter. Because distinguished names are difficult to remember, it is useful to have other means for retrieving objects. Active Directory supports querying by attribute (for example, the building number where you have to find a printer), so an object can be found without having to know the distinguished name. (For more information about searching Active Directory, see "Name Resolution in Active Directory" in this book.)

Relative Distinguished Name

The relative distinguished name (also known as the "RDN") of an object is the part of the name that is an attribute of the object itself—the part of the object name that identifies this object as unique from its siblings at its current level in the naming hierarchy. In Figure 1.10, in the preceding section, the relative distinguished name of the object is JSmith. The relative distinguished name of the parent object is Users. The maximum length allowed for a relative distinguished name is 255 characters, but attributes have specific limits imposed by the directory schema. For example, in the case of the common name, which is the attribute type often used for naming the relative distinguished name (cn), the maximum number of characters allowed is 64.

Active Directory relative distinguished names are unique within a specific parent —that is, Active Directory does not permit two objects with the same relative distinguished name under the same parent container. However, 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. (For example, the object cn=JSmith,dc=noam,dc=reskit,dc=com is recognized by LDAP as being different from cn=JSmith,dc=reskit,dc=com.)

The relative distinguished name for each object is stored in the Active Directory database. Each record contains a reference to the parent of the object. By following the references to the root, the entire distinguished name is constructed during an LDAP operation. (For more information about LDAP operations, see "Name Resolution in Active Directory" in this book.)

Naming Attributes

As illustrated earlier in this section, an object name consists of a series of relative distinguished names that represent the object itself and also every object in the hierarchy above it, up to the root object. Each portion of the distinguished name is expressed as *attribute_type=value* (for example, cn=JSmith). The attribute type that is used to describe the object's relative distinguished name (in this case, cn=) is called the naming attribute. If you were to create a new class in the Active Directory schema (that is, a new *classSchema* object), the optional *RdnAttID* attribute could be used to specify the naming attribute for the class. In Active Directory, instances of default objects that you create have a default mandatory naming attribute. For example, part of the definition of the class User is the attribute cn (Common-Name) as the naming attribute. For this reason, the relative distinguished name for user JSmith is expressed as cn=JSmith.

The naming attributes shown in Table 1.1 are used in Active Directory, as described in RFC 2253.

Table 1.1 Default Active Directory Naming Attributes

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

Other naming attributes described in RFC 2253, such as o= for organization name and c= for country/region name, are not used in Active Directory, although they are recognized by LDAP.

The use of distinguished names, relative distinguished names, and naming attributes is required only when you are programming for LDAP and using Active Directory Service Interfaces (ADSI) or other scripting or programming languages. The Windows 2000 user interface does not require you to enter such values.

For more information about creating new *classSchema* objects, see "Active Directory Schema" in this book. For more information about using ADSI, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Object Identity and Uniqueness

In addition to its distinguished name, every object in Active Directory has a unique identity. Active Directory is identity based—that is, objects are known internally by their identity, not by their current name. Objects might be moved or renamed, but their identity never changes. The identity of an object is defined by a globally unique identifier (GUID), a 128-bit number that is assigned by the directory system agent when the object is created. The GUID is stored in an attribute, *objectGUID*, that is present on every object. The *objectGUID* attribute is protected so that it cannot be altered or removed. When you store a reference to an Active Directory object in an external store (for example, a database such as Microsoft® SQL Server™), the *objectGUID* value should be used. Unlike a distinguished name or a relative distinguished name, which can be changed, the GUID never changes.

Active Directory Name Formats

Several formats for providing object names are supported by Active Directory. These formats accommodate the different forms a name can take, depending on its application of origin. Active Directory administrative tools display name strings in a default format, which is the canonical name. 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 RFC 1779 and RFC 2247 naming conventions, which take the form cn=common name, ou=organizational unit, o=organization, c=country/region. Active Directory uses the domain component (dc) instead of o=organization and does not support c=country/region. In the LDAP distinguished name, the relative distinguished names appear in order beginning at the left with the name of the leaf and ending at the right with the name of the root, as shown here:

```
cn=jsmith,ou=promotions,ou=marketing,dc=noam,dc=reskit,dc=com
```

LDAP Uniform Resource Locator (URL). Active Directory supports access through the LDAP protocol from any LDAP-enabled client. LDAP URLs are used in scripting. An LDAP URL names the server holding Active Directory services and the attributed name of the object (the distinguished name). For example:

```
LDAP://server1.noam.reskit.com/cn=jsmith,ou=promotions,
ou=marketing,dc=noam,dc=reskit,dc=com
```

Active Directory Canonical Name. By default, the Windows 2000 user interface displays object names that use the canonical name, which lists the relative distinguished names from the root downward and without the RFC 1779 naming attribute descriptors; it uses the DNS domain name (the form of the name where the domain labels are separated by periods). For the LDAP distinguished name in the previous example, the respective canonical name would appear as follows:

```
noam.reskit.com/marketing/promotions/jsmith
```

Note If the name of an organizational unit contains a forward slash character (/), the system requires an escape character in the form of a backslash (\) to distinguish between forward slashes that separate elements 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 Promotions/Northeast and the name of the domain is Reskit.com, the canonical name is displayed as Reskit.com/Promotions\/Northeast.

DNS-to-LDAP Distinguished Name Mapping

Although DNS domain names match Active Directory domain names, they are not the same thing. Active Directory names have a different format, which is required by LDAP to identify directory objects. DNS domain names are therefore mapped to Active Directory domain names, and vice versa, as described in RFC 2247.

All access to Active Directory is carried out through LDAP. LDAP uses distinguished names to provide unique names to directory objects; every object in Active Directory has an LDAP distinguished name. A distinguished name is a naming structure that consists of a string of the hierarchical components that make up the complete object. Each distinguished name component is the relative distinguished name of an object in the hierarchy, beginning with the object itself and ending with the root object in the domain tree. An algorithm automatically provides 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. Each DNS label corresponds to the relative distinguished name of an Active Directory domain. For example, the DNS domain noam.reskit.com is translated to the LDAP distinguished name that has the form dc=noam,dc=reskit,dc=com.

Logon Names

A unique logon name is required by user security principals for gaining access to a domain and its resources. Security principals are objects to which Windows security is applied in the form of authentication and authorization. Users are security principals, and they are authenticated (their identity is verified) at the time they log on to the domain or local computer. They are authorized (allowed or denied access) when they use resources.

User security principals have two types of logon names:

SAM Account Name. A SAM account name is a name that is required for compatibility with Windows NT 4.0 and Windows NT 3.*x* domains. SAM account names are sometimes referred to as flat names (because there is no hierarchy in the naming, so every name must be unique in the domain). These terms serve to differentiate these names from DNS hierarchical names.

User Principal Name. A user principal name (also known as a "UPN") is a "friendly" name that is shorter than the distinguished name and easier to remember. The user principal name consists of a shorthand name that represents the user and usually the DNS name of the domain where the user object resides, or any other designated name.

The user principal name format consists of the user name, the "at" sign (@), and a user principal name suffix. For example, the user James Smith, who has a user account in the reskit.com domain, might have the user principal name JSmith@reskit.com. The user principal name is independent of the distinguished name of the user object, so a user object can be moved or renamed without affecting the user logon name.

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

If you create no other user principal name, the user principal name suffix for a security principal is the domain in which the account is created (for example, @reskit.com). You can create additional user principal name suffixes and assign them to security principal accounts if you don't want to use the default domain name (for example, if the DNS domain name is extremely long and hard to remember). The e-mail name can also be used as the user principal name suffix. For example, in a large organization that has many domains, a user's e-mail address might be *<userName>@<companyName>*.com.

You can manage user principal name suffixes for a domain in the Active Directory Domains and Trusts console in MMC. To add or remove a user principal name suffix, open the properties for the Active Directory Domains and Trusts node. User principal names are assigned at the time a user or group is created. If you have created additional suffixes for the domain, you can select from the list of available suffixes when you create the user or group account.

The suffixes appear in the list in the following order:

- Alternate suffixes. If you have created additional suffixes, the last one that you created appears first.
- Root domain.
- The current domain.

For more information about creating user principal names, see Windows 2000 Server Help.

Domain Controllers

A domain controller is a computer that is running Windows 2000 Server and hosts Active Directory. Domain controllers run the KDC service, which is responsible for authenticating domain user logons. A domain controller stores *directory* partitions. Directory partitions (also known as "naming contexts") correspond to the logically distributed segments of Active Directory that are replicated as discrete units. These segments correspond to the following directory partitions:

- A domain, of which there can be many in a particular forest (directory).
- The directory schema, of which there is one in a particular forest (directory).
- The Configuration container, of which there is one in a particular forest (directory).

In addition to the domain directory partition that it stores, every domain controller stores a replica of the schema directory partition and the configuration directory partition. (For more information about directory partitions, see "Active Directory Data Storage" in this book.)

Multimaster Operations

A domain can deploy many domain controllers, and all domain controllers can accept Active Directory changes. Earlier versions of Windows NT used multiple domain controllers, only one of which was allowed to update the directory database. This single-master scheme required all changes to be replicated from the primary domain controller to the backup domain controllers.

In Windows 2000, every domain controller can receive changes, and the changes are replicated to all other domain controllers. The day-to-day operations that are associated with managing users, groups, and computers are typically multimaster operations—that is, changes to these objects can be made on any domain controller. There are some operations, however, that are not performed as multimaster operations because they must occur at only one place and time. For these operations, there are specially designated domain controllers that manage the operations singly.

Single-Master Operations

Most operations can be made at any domain controller and the effects of these operations (for example, deleting a user object) are replicated to all other domain controllers that store a replica of the same directory partition in which the change occurred. However, there are certain operations that must occur at only one domain controller.

The domain controllers that are assigned to manage single-master operations are called role owners for the operations. (For more information about managing single-master operations, see "Managing Flexible Single-Master Operations" in this book.) The single-master operations include the following:

Relative ID Pool Allocation One domain controller per domain is responsible for assigning "pools" of relative identifiers to other domain controllers in that domain. Relative identifiers (also known as "RIDs") are identifiers that are used in association with a domain identifier to make up the security identifier (also known as a "SID") for each security principal created in Active Directory. To ensure uniqueness in a domain, a single domain controller has the relative ID master role. The relative ID master assigns relative identifiers from a single pool of these identifiers for the domain.

Schema Modification Changes to the same schema objects on different domain controllers can result in an inconsistent directory schema and corrupt data. For this reason, a single domain controller in a forest has the schema master role. The schema master is responsible for all changes to the schema directory partition.

Primary Domain Controller Emulation For compatibility with Windows NT 3.51–based and Windows NT 4.0–based servers, which can operate as backup domain controllers in a mixed-mode Windows 2000 domain but still require a primary domain controller (also known as the "PDC"), a specific Windows 2000–based domain controller, the PDC emulator, is assigned to emulate the role of the primary domain controller. This domain controller is perceived by the Windows NT 3.51–based and Windows NT 4.0–based servers as a primary domain controller. In a Windows 2000 domain, one domain controller is assigned to be the PDC emulator and performs the role of the primary domain controller.

For information about upgrading Windows NT 3.51 and Windows NT 4.0 domains to Windows 2000 domains, see "Determining Domain Migration Strategies" in the *Deployment Planning Guide*.

Certain Infrastructure Changes When objects are moved or deleted, a single domain controller per domain, the infrastructure master, is responsible for updating the security identifiers and distinguished names in cross-domain object references in that domain.

Domain Naming A single domain controller per forest, the domain naming master, is assigned the responsibility of ensuring that domain names are unique in the forest and that cross-reference objects to external directories are maintained.

For more information about managing single-master roles, see "Managing Flexible Single-Master Operations" in this book.

Global Catalog Servers

Every domain controller 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 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. The additional directory partitions are "partial" because, although they collectively contain every object in the directory, only a limited set of specific attributes are included for each object. The Global Catalog is built automatically by the Active Directory replication system.

All of the directory partitions on a Global Catalog server, whether full or partial partitions, are stored in a single directory database (Ntds.dit) on that server. There is no separate storage area for Global Catalog attributes; they are treated as additional information in the domain controller directory database.

When a new domain is added to the forest, the information about the new domain is stored in the configuration directory partition, which reaches the Global Catalog server (and all domain controllers) through replication of forest-wide information. When a new Global Catalog server is designated, this information is also stored in the configuration directory partition and replicated to all domain controllers in the forest.

Global Catalog Attributes

In its role as a domain controller, a Global Catalog server stores one domain directory partition that has writable objects with a full complement of writable attributes. The objects in all other domain directory partitions in the forest are stored on a Global Catalog server as read-only objects with a partial set of attributes. An attribute is marked as being replicated to the Global Catalog as part of its schema definition. In the Active Directory Schema console in MMC, you can use the **Replicate this attribute to the Global Catalog** check box to designate an *attributeSchema* object as a member of the attribute set that is replicated to the Global Catalog servers. If this check box is selected, the value in the attribute *isMemberOfPartialAttributeSet* on the *attributeSchema* object is set to TRUE, and the attribute is replicated to the Global Catalog as part of normal Active Directory replication. The replication topology for the Global Catalog is generated automatically by the Knowledge Consistency Checker (also known as the "KCC"), a built-in process that implements a replication topology that is guaranteed to deliver the contents of every directory partition to every Global Catalog server. The attributes replicated into the Global Catalog include a base set defined by Microsoft. Administrators can use the Active Directory Schema console to specify additional attributes to meet the needs of their installation.

For information about adding an attribute to the Global Catalog attribute set, see Windows 2000 Server Help and "Active Directory Schema" in this book. For information about the Knowledge Consistency Checker and replication, see "Active Directory Replication" in this book.

Designating a Global Catalog

The first domain controller in a forest is automatically designated as a Global Catalog. Thereafter, a domain controller can be designated as a Global Catalog in the **NTDS Settings Properties** dialog box in Active Directory Sites and Services. The NTDS Settings object is a child of the server object, which is a child of the site object in the Sites container. When you select the **Global Catalog Server** check box, the domain controller is added to the Global Catalog replication topology and populated by means of the normal replication process. When you change an attribute that is flagged as belonging in the Global Catalog in *any* domain, it is replicated to all Global Catalog servers.

The NTDS Settings object has the multivalue attribute *hasMasterNCs*, which identifies the directory partitions that the domain controller stores. ("NC" stands for "naming context," which is a synonym for "directory partition.") For every domain controller, there are exactly three "master" (full and writable) directory partitions: the domain directory partition, the schema directory partition, and the configuration directory partition. The NTDS Settings object also has the multivalue attribute *hasPartialReplicationNCs*. If the domain controller is a Global Catalog server, this attribute has a value for each domain directory partition in the forest, and it receives attribute changes through replication with each respective domain directory partition in the forest.

Because the NTDS Settings object is stored in the configuration directory partition, which is replicated to all domain controllers in the forest, all domain controllers have the information about which servers are Global Catalog servers.

For more information about designating a Global Catalog server, see Windows 2000 Server Help.

Global Catalog and Domain Logon Support

In a native-mode domain, a Global Catalog server is a requirement for logging on to the domain. For this reason, it is advisable to have at least one Global Catalog server in a site. If a Global Catalog is not available in a site and there is another Global Catalog server in a remote site, the server in the remote site can be used for the logon process. If no Global Catalog is available in any site, the logon process proceeds with cached logon information.

Note A member of the Domain Admins group can complete the logon process (not cached) even when a Global Catalog server is not available.

Universal Group Membership

The reason that a Global Catalog must be available for the domain logon process is that the membership for universal groups is not stored on all domain controllers. Because the membership of all universal groups is replicated to Global Catalog servers, the complete universal group membership of a user can be determined by querying a Global Catalog server.

Note Universal groups are available only when a domain is in native mode.

During the logon process, a security token that contains the groups to which the user belongs is associated with the user. Because universal group membership is stored only on Global Catalog servers, only these servers can identify a user as having membership in a specific universal group. If a universal group is present as an access control entry in an access control list on a specific directory object, the access token associated with the user during the logon session must contain that group in order for the Allow or Deny access permission to be applied to the user. Otherwise, a user could be granted access (on the basis of another group membership) to an object that is specifically denied that user as a member of the universal group. Similarly, this user would not be able to gain access to resources to which he or she has legitimate access as a member of the universal group.

Note Deny access permission is processed before Allow access permission. Therefore, if you are denied access to an object by virtue of membership in one group and allowed access by virtue of membership in another group, the Deny access takes precedence over the Allow access.

User Principal Name and Global Catalog Logon Support

User principal names are user names that can be used when a user is logging on to a Windows 2000 domain. A user also can provide a SAM account name (*<DomainName\UserName>*). In the Windows 2000 logon screen, you can type your user name and select the domain name from the list, or you can use the user principal name. If you use the user principal name, when you type the "at" sign (@), the domain list is unavailable; Windows 2000 takes the domain name from the user principal name suffix.

The user principal name format (*<UserName>@<DNSDomainName>*) is resolved by the Global Catalog server. If a company has more than one forest and uses trust relationships between the domains in the different forests, a user principal name cannot be used to log on to a domain that is outside the forest because the user principal name is resolved in the Global Catalog of the forest. For information about Global Catalog placement to facilitate logging on to domains, see "Designing the Active Directory Structure" in the *Deployment Planning Guide*.

Search Requests and the Global Catalog

Because the Global Catalog stores every object in the forest, it can be used to locate objects in any domain without a referral to a different server. When a search request is sent to port 389 (the default LDAP port), the search is conducted on a single directory partition. If the object is not found in that directory partition (and is not in the schema or configuration directory partitions), the request is referred to a domain controller in a different domain that is assumed to contain the requested object, on the basis of the distinguished name that is presented in the search request.

When a search request is sent to port 3268 (the default Global Catalog port), the search includes all directory partitions in the forest—that is, the search is processed by a Global Catalog server. If the request specifies attributes that are part of the Global Catalog attribute set, the Global Catalog can return results for objects in any domain without generating a referral to a domain controller in a different domain.

For more information about LDAP search referrals and Global Catalog searches, see "Name Resolution in Active Directory" in this book.

Organizational Units

Active Directory allows administrators to create a hierarchy within a domain that meets the needs of their organization. The object class of choice for building these hierarchies is the class *organizationalUnit*, a general-purpose container that can be used to group most other object classes together for administrative purposes. An organizational unit in Active Directory is analogous to a directory in the file system; it is a container that can hold other objects.

Administrative Hierarchy

Organizational units can be nested to create a hierarchy within a domain and form logical administrative units for users, groups, and resource objects, such as printers, computers, applications, and file shares. The organizational unit hierarchy within a domain is independent of the structure of other domains; each domain can implement its own hierarchy. Likewise, domains that are managed by a central authority can implement similar organizational unit hierarchies. The structure is completely flexible, which allows organizations to create an environment that mirrors the administrative model, whether it is centralized or decentralized.

For information about planning and implementing an organizational unit hierarchy, see "Designing the Active Directory Structure" in the *Deployment Planning Guide*.

Group Policy

Group Policy can be applied to organizational units to define the abilities of groups of computers and users that are contained within the organizational units. Levels of control range from complete desktop lockdown to a relatively autonomous user experience. Group Policy can affect functionality, such as what applications are available to a group of users, what features within an application are accessible on a particular computer, where documents are saved, and access and user permissions. Group Policy also affects where, when, and how application and operating system updates or special scripts are applied.

Group Policy settings are stored as Group Policy objects in Active Directory. A Group Policy object can be associated with one or more Active Directory containers, such as a site, domain, or organizational unit.

For more information about Group Policy, see "Introduction to Desktop Management," "Software Installation and Maintenance," and "Group Policy" in this book.

Delegation of Control

The Windows 2000 object-based security model implements default access control that is propagated down a particular subtree of container objects. You use this technology to determine the security for an entire group of objects according to the security that you set on the organizational unit that contains the objects, which effectively delegates administrative control to individuals in the organization. The best way to take full advantage of delegation and inherited control on directory objects is to organize the hierarchy to match the way that the directory is administered.

Note Because Active Directory is indexed, there is no need to organize the tree for ease of browsing, which is likely to run counter to administrative objectives.

Administrative control over directory objects can be applied—or delegated—to organizational units through access control. (For more information about administrative control, see "Delegation of Administration" later in this chapter.)

Object Security

Authentication of user accounts determines that a user who logs on to a Windows 2000 domain is who the user claims to be and that the user does indeed have an account either in the domain or in a domain that is trusted. After the user is authenticated, however, Active Directory must provide security (authorization) to determine what objects the authenticated user can view or change and what kinds of changes are allowed. This type of security is achieved through access control.

Note The information presented here is provided as a security overview in the context of understanding basic Active Directory functionality. For more information about Active Directory security, see the chapters under "Distributed Security" in this book.

Access Control

All Active Directory objects are protected by an ACL. ACLs determine who can see the object and what actions each user can perform on the object. The existence of an object is never revealed to a user who is not allowed to read it.

An ACL is a list of access control entries (ACEs) that are stored with the object that the ACL protects. In Windows 2000, an ACL is stored as a binary value within a security descriptor. Each ACE contains a security identifier that identifies the security principal (the user or group) to whom the access control entry applies and also information about what type of access the access control entry grants or denies.

ACLs on Active Directory objects contain ACEs that apply to the object as a whole and ACEs that apply to the individual properties of the object. This structure allows an administrator to control not only which users can see an object but also what properties the users can see. For example, all users might be granted read access to the e-mail and telephone number properties for all other users, but the 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 properties such as the telephone and mailing addresses on their own user objects. Use the **Delegate Control** command in the context menu of an organizational unit to set the access limits for appropriate groups.

For more information about access control, see "Access Control" in this book. For more information about built-in object security, see "Active Directory Data Storage" in this book. For information about anonymous read access, see "Name Resolution in Active Directory" in this book.

Delegation of Administration

Delegation is one of the most important security features of Active Directory. Delegation allows a higher administrative authority to grant specific administrative user rights for containers and subtrees to individuals and groups. Delegation eliminates the need for domain administrators to have sweeping authority over large segments of the user population.

ACEs can grant specific administrative rights on the objects in a container to a user or group. Rights are granted for specific operations on specific object classes through ACEs in the container's ACL. For example, to allow the user James Smith to be an administrator of the Corporate Accounting organizational unit, you add ACEs to the ACL on Corporate Accounting as shown in Table 1.2.

Table 1.2 Example of ACL Contents on an Organizational Unit

ACE	Security Principal	Right	Applied to These Objects
Allow	James Smith	Create, Delete User objects	This object only
Allow	James Smith	Full control	User objects
Allow	James Smith	Create, Delete Group objects	This object only
Allow	James Smith	Full control	Group objects
Allow	James Smith	Set Password	User objects

Now James Smith can create new users and groups in Corporate Accounting and set the passwords on existing users, but he can neither create any other object classes nor affect users in any other containers (unless, of course, he is granted that access by ACEs in the other containers).

For more information about delegation of administration, see "Access Control" in this book. For information about how to apply delegation, see Windows 2000 Server Help.

Inheritance

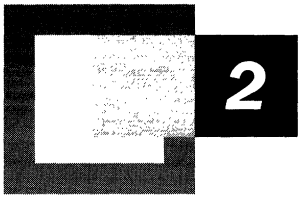
You use inheritance to propagate a particular ACE from the container where it was applied to all objects within the container. Inheritance can be combined with delegation to grant administrative rights to a whole subtree of the directory in a single operation. For more information about inheritance, see "Access Control" in this book.

Additional Resources

- For more information about DNS, see *DNS and BIND*, 3d ed., by Paul Albitz and Cricket Liu, 1998, Sebastopol, CA: O'Reilly & Associates.
- For more information about Requests for Comments (RFCs) and Internet Drafts, see the Internet Engineering Task Force (IETF) link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

CHAPTER 2

Active Directory Data Storage



A directory service consists of both a directory storage system (called the *directory store*) and a mechanism that is used to locate and retrieve information from the system. Active Directory™, the directory service that is included with Microsoft® Windows® 2000, stores objects that provide information about the real things that exist in an organization's network and that are associated with one or more domains, such as users, specific groups of users, computers, applications, services, files, and distribution lists. It then makes this information available to users and applications throughout the organization.

In This Chapter

Related Information in the Resource Kit

- For more information about the Active Directory hierarchy and Domain Name System (DNS) naming, domain controller location, and tree and forest structure, see "Active Directory Logical Structure" in this book.
- For more information about replication between sites and within a site, see "Active Directory Replication" in this book.
- For more information about understanding and modifying the Active Directory schema, see "Active Directory Schema" in this book.
- For more information about DNS, see "Introduction to DNS" and "Windows 2000 DNS" in the *Microsoft® Windows® 2000 Server Resource Kit TCP/IP Core Networking Guide*.

Active Directory Architecture

Understanding the interactions of Active Directory architectural components provides the basis for understanding how Active Directory stores and retrieves data. The first step is understanding the relationship of Active Directory to the rest of the Microsoft® Windows® 2000 Server operating system.

Active Directory and Windows 2000 Architecture

Windows 2000 uses modules and modes that combine to provide operating system services to applications. Two processor access modes, *kernel* and *user*, divide the low-level, platform-specific processes from the upper-level processes, respectively, to shield applications from platform differences and to prevent direct access to system code and data by applications. Each application, including service applications, runs in a separate *module* in user mode, from which it requests system services through an application programming interface (API) that gains limited access to system data. An application process begins in user mode and is transferred to kernel mode, where the actual service is provided in a protected environment. The process is then transferred back to user mode. The *security subsystem* in user mode is the module in which Active Directory runs. The *security reference monitor*, which runs in kernel mode, is the primary authority for enforcing the security rules of the security subsystem. Figure 2.1 shows the location of Active Directory within Windows 2000.

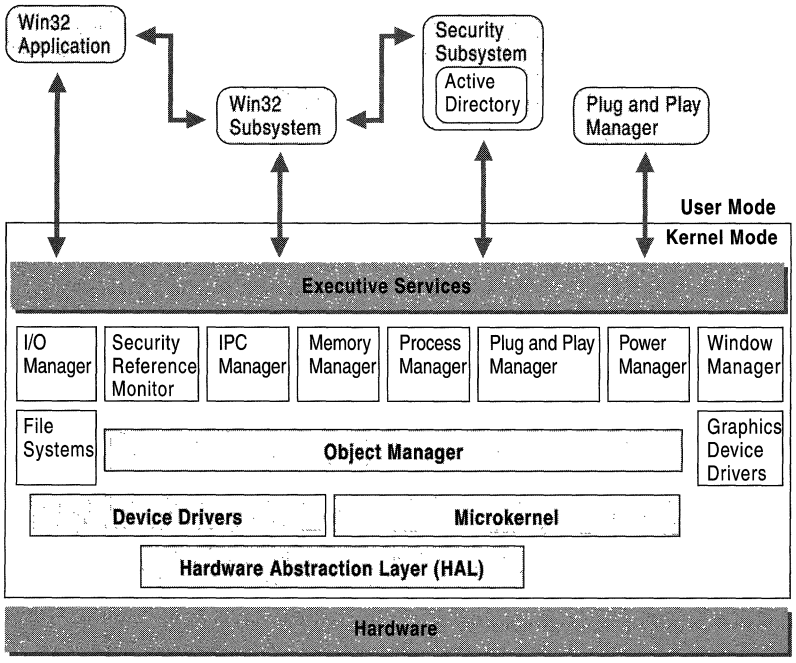


Figure 2.1 Active Directory Within the Windows 2000 Operating System

The tight integration of the directory service and security subsystem services is key to the implementation of Windows 2000 distributed systems. Access to all directory objects first requires proof of identity (authentication), which is performed by components of the security subsystem, and then validation of access permissions (authorization), which is performed by the security subsystem in conjunction with the security reference monitor. The security reference monitor enforces the access control applied to Active Directory objects.

For more information about the Windows 2000 operating system, see "Overview of Networking in Windows 2000 Professional" in the *Microsoft® Windows® 2000 Professional Resource Kit*, which contains information about the core technologies for both Microsoft® Windows® 2000 Professional and Windows 2000 Server. For more information about authentication, see "Authentication" in this book. For more information about access permissions, see "Access Control" in this book.

Security Subsystem Architecture

Windows 2000 includes a set of security components that make up the Windows security model. These components ensure that applications cannot gain access to resources without authentication and authorization. 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

The security subsystem keeps track of the security policies and the accounts that are in effect on the computer system. In the case of a *domain controller*, which is a computer that has Active Directory installed, these policies and accounts are the ones that are in effect for the domain in which the domain controller is located. They are stored in Active Directory.

The *Local Security Authority (LSA)* is a protected subsystem that maintains the information about all aspects of local security on a system (collectively known as the *local security policy*) and provides various services for translation between names and identifiers.

In general, the LSA performs the following functions:

- Manages local security policy.
- Provides interactive user authentication services.
- Generates tokens, which contain user and group information as well as information about the security privileges for that user. After the initial logon process is complete, all users are identified by their security identifier (SID) and the associated access tokens.
- Manages the Audit policy and settings. When an audit alert is generated by the Security Reference Monitor, the LSA is charged with writing that alert to the appropriate system log.

The local security policy identifies the following:

- The domains that are trusted to authenticate logon attempts.
- Who can have access to the system and in what way (for example, interactively, over the network, or as a service).
- Who is assigned privileges.
- What security auditing is to be performed.
- Default memory quotas (paged and nonpaged memory pool usage).

Figure 2.2 shows a local perspective of Active Directory within the LSA security subsystem (Lsass.exe). The LSA security subsystem provides services to both the kernel mode and the user mode for validating access to objects, checking user privileges, and generating audit messages.

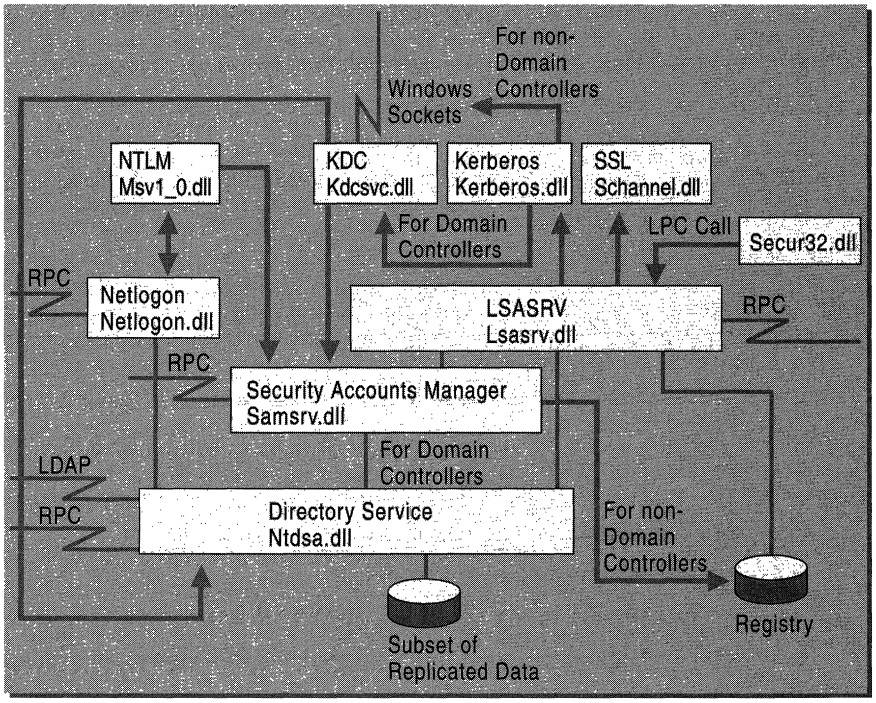


Figure 2.2 Active Directory Within the Local Security Authority (Lsass.exe)

The LSA has the following components:

Netlogon.dll. The Net Logon service. Net Logon maintains the computer's secure channel to a domain controller. It passes the user's credentials through a secure channel to the domain controller and returns the domain security identifiers and user rights for the user. In Windows 2000, the Net Logon service uses DNS to resolve names to the Internet Protocol (IP) addresses of domain controllers. Net Logon is the replication protocol for Microsoft® Windows NT® version 4.0 primary domain controllers and backup domain controllers.

Msv1_0.dll. The NTLM authentication protocol. This protocol authenticates clients that do not use Kerberos authentication.

Schannel.dll. The Secure Sockets Layer (SSL) authentication protocol. This protocol provides authentication over an encrypted channel instead of a less-secure clear channel.

Kerberos.dll. The Kerberos v5 authentication protocol.

Kdcsvc.dll. The Kerberos Key Distribution Center (KDC) service, which is responsible for granting ticket-granting tickets to clients.

Lsasrv.dll. The LSA server service, which enforces security policies.

Samsrv.dll. The Security Accounts Manager (SAM), which stores local security accounts, enforces locally stored policies, and supports APIs.

Ntdsa.dll. The directory service module, which supports the Windows 2000 replication protocol and Lightweight Directory Access Protocol (LDAP), and manages partitions of data.

Secur32.dll. The multiple authentication provider that holds all of the components together.

For more information about the LSA and its components, see "Authentication" in this book. For more information about access control, see "Access Control" in this book.

Directory Service Architecture

Active Directory functionality can be described as a layered architecture in which the layers represent the 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 accommodate the different types of information that are 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, in the case of replication, between two directory services.

Figure 2.3 shows the Active Directory service layers and their respective interfaces and protocols. The direction of the arrows indicates the manner in which the 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 separate dynamic-link library (DLL) and can call only entry points exported by the directory system agent DLL, Ntdsa.dll. 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. Thus, a three-way interaction is required between the three DLLs.

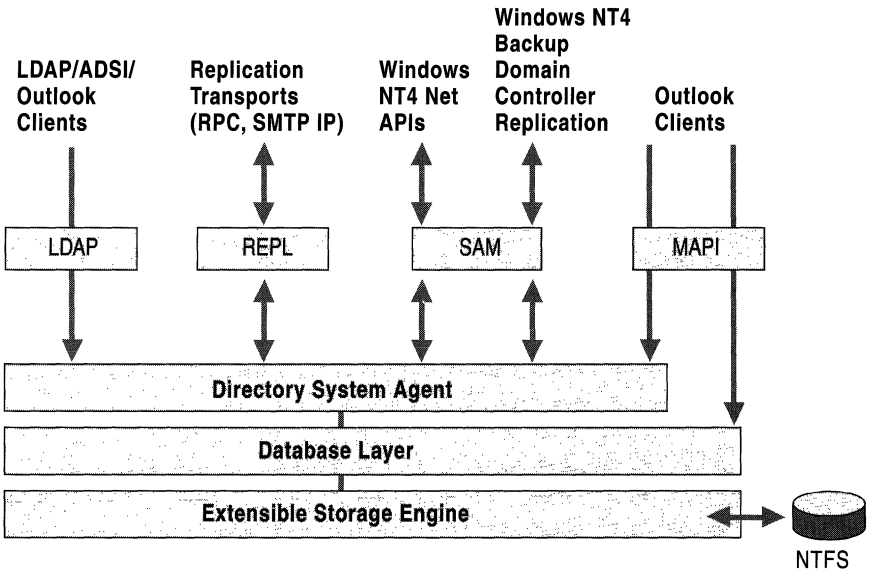


Figure 2.3 Active Directory Service Layers and Interface Agents

The key service components include the following:

- Directory system agent. Builds a hierarchy from the parent-child relationships stored in the directory. Provides APIs for directory access calls.
- Database layer. Provides an abstraction layer between applications and the database. Calls from applications are never made directly to the database; they go through the database layer.
- Extensible storage engine. Communicates directly with individual records in the directory data store on the basis of the object's relative distinguished name attribute.

- Data store (the database file Ntds.dit). This file is manipulated only by the extensible storage engine database engine. You can administer the file by using the Ntdsutil command-line tool. (To use Ntdsutil, install the Support Tools that are located in the Support\Tools folder on the Windows 2000 Server operating system CD. To install the tools, double-click the **Setup** icon in that folder. For information about installing and using the Windows 2000 Support Tools and Support Tools Help, see the file Sreadme.doc in the Support\Tools folder of the Windows 2000 operating system CD.)

For more information about using Ntdsutil, see "Active Directory Diagnostic Utility (Ntdsutil.exe)" and "Active Directory Diagnostics, Troubleshooting, and Recovery" in this book, and see *Microsoft Windows 2000 Support Tools* Help.

Clients obtain access to Active Directory by using one of the following mechanisms that are supported by Active Directory:

- LDAP/ADSI. Clients that support LDAP use it to connect to the directory system agent. The extensible storage engine (ESE) that is used by Microsoft® Exchange Server version 5.5 (and earlier) client/server messaging and groupware also uses LDAP. Active Directory supports LDAPv3 (defined by RFC 2251) and LDAPv2 (defined by RFC 1777). Windows 2000 clients, as well as Microsoft® Windows® 98 and Microsoft® Windows® 95 clients that have the Active Directory client components installed, use LDAPv3 to connect to the directory system agent. Active Directory Service Interfaces (ADSI) is a means of abstracting the LDAP API by providing component object model (COM) interfaces to Active Directory; however, Active Directory uses only LDAP. The LDAP API is part of Wldap32.dll.
- MAPI. Microsoft® Outlook® messaging and collaboration clients connect to the directory system agent by using the MAPI remote procedure call (RPC) Address Book provider interface.
- SAM. Windows clients that use Windows NT 4.0 or earlier use the SAM interface to connect to the directory system agent. Replication from backup domain controllers in a mixed-mode domain goes through the SAM interface as well.
- REPL. During directory replication, Active Directory directory system agents connect to each other by using a proprietary RPC interface.

Directory System Agent

The *directory system agent (DSA)* is the process that provides access to the *store*. The store is the physical store of directory information located on a hard disk. The DSA is the server-side process that creates an instance of a directory service. Clients use one of the supported interfaces to connect (bind) to the DSA and then search for, read, and write Active Directory objects and their attributes.

The Active Directory namespace is partitioned so that individual domain controllers do not store the entire directory. Every DSA holds at least a single Windows 2000 directory partition that stores domain data for a domain (such as users, groups, and organizational units) plus two non-domain directory partitions that store forestwide data, which includes the schema and configuration data.

The DSA layer provides the following functionality:

Object Identification Every object in Active Directory has a permanent globally unique identifier (GUID) that is associated with several string forms of the object name (SAMAccountName, user principal name, distinguished name) as well as a security identifier. These object names and the security identifier are not permanent—that is, they can be changed. All permanent references to the object are kept in terms of the GUID; the object name is used for hierarchy navigation and display purposes, and the security identifier is used for access control. The DSA maintains the GUID association with an object when the object's string name or security identifier changes.

Schema Enforcement of Updates In a multimaster system, a change to a schema object in one replica might conflict with existing objects in that replica and also with objects in other replicas. In Windows 2000, a schema change is a single-master operation, so if an update does not produce a conflict at the originating replica, the update is considered acceptable at all replicas. Thus, replicated updates do not perform any schema checks, and you do not have to wait until the schema replicates before creating instances of a new object or attribute.

Access Control Enforcement The DSA enforces security limitations in the directory. The DSA layer reads security identifiers (SIDs) on the access token.

Support for Replication The DSA contains the hooks for replication notifications. All object updates ultimately must go through the appropriate function for the directory service to work properly.

Referrals DSA manages the directory hierarchy information (referred to as "knowledge"), which it receives from the database layer. DSA is responsible for cross-references of Active Directory domain objects up and down the hierarchy and also out to other domain hierarchies.

Database Layer

The *database layer* provides an object view of database information by applying schema semantics to database records, thereby isolating the upper layers of the directory service from the underlying database system. The database layer is an internal interface. No database access calls are made directly to the extensible storage engine; instead, all database access is routed through the database layer.

Active Directory provides a hierarchical namespace. Each object is uniquely identified in the database by its *distinguished name*. The individual naming attribute, called the *relative distinguished name*, is unique within the object's parent container; the relative distinguished name and the chain of successive parent object names make up the object's distinguished name. The database stores the relative distinguished name for each object, as well as a reference to the parent object. The database layer follows these parent references and concatenates the successive relative distinguished names to form distinguished names.

All data that describes an object is held as a set of attributes, which are stored as columns in the database. The database layer is responsible for the creation, retrieval, and deletion of individual records, attributes within records, and values within attributes. To carry out these functions, the database layer uses the schema cache (an in-memory structure in the DSA) to get the information about the attributes that it needs. For more information about the schema cache, see "Active Directory Schema" in this book. For more information about distinguished names and relative distinguished names, see "Active Directory Logical Structure" in this book.

Extensible Storage Engine

Active Directory is implemented on top of an indexed sequential access method (ISAM) table manager. This database is a version of the ESE database that is used by Microsoft® Exchange Server version 5.5 client/server messaging and groupware. The Windows 2000 version of this database is Esent.dll.

ESE stores all Active Directory objects. It can support a database up to 16 terabytes in size, which can theoretically hold many millions of objects per domain.

Note Testing of the database has been carried out to 40 million objects per domain.

The following ESE characteristics make it well suited to the storage needs of Active Directory:

- Is used by the directory service and information store in Exchange Server version 5.5.
- Supports indexing.
- Supports multivalue attributes.
- Supports update operations that are transacted for stability and integrity across system failures.
- Can be backed up while the domain controller is online.
- Handles sparse rows well — that is, rows in which many of the properties do not have values.

Active Directory comes with a predefined schema that defines all of the attributes that are required and allowed for a given object. ESE reserves storage only for the space that is used—that is, only for the attributes that have values, not for all possible attributes. For example, if a user object already has 50 attributes defined in the schema and you create a user with values for only 4 attributes, storage space is allocated only for those 4 attributes. If more attributes are added later, more storage is allocated for them.

Esent.dll implements the search and retrieval functionality of the underlying database. Also, ESE is able to store attributes that can have multiple values. For example, the database can store multiple phone numbers for a single user without requiring a different phone number attribute for each phone number.

Protocols and Interfaces to Active Directory

The diagram of the Active Directory architecture (Figure 2.3) illustrates four avenues of entry to Active Directory: LDAP/ADSI, REPL (replication), SAM, and MAPI. Each of these access routes uses a different set of protocols and APIs that enable communication with the directory service. Table 2.1 shows the APIs that Active Directory supports and that can be used by developers to integrate with Active Directory or use resources in Active Directory.

Table 2.1 Active Directory APIs

API Name	Description
LDAP C API	As described in RFC 1823 for LDAPv3, LDAP API is a C language API to the LDAP network protocol.
ADSI	COM interface to Active Directory that abstracts the details of LDAP communications. ADSI provides services and Active Directory information to directory-aware applications. ADSI supports multiple programming languages, including Microsoft® Visual Basic®, C, and Microsoft® Visual C++®. ADSI also can be accessed by using Windows Script Host (WSH).
MAPI	Messaging API that is supported for compatibility with Microsoft® Exchange Client and Outlook Address Book client applications.
Windows NT 4.0	Windows NT 4.0 networking APIs (Net APIs) that are used by Windows NT 4.0 clients to gain access to Active Directory through SAM.
SAM	APIs that communicate with the DSA APIs.

These APIs communicate with Active Directory by using various access methods, as described in Table 2.2.

Table 2.2 Active Directory Access Methods

Access Method	Description
LDAP	Core protocol that is supported by Active Directory, as described in RFC 2251 (LDAPv3) and RFC 1777 (LDAPv2).
MAPI RPC	RPC interfaces used by MAPI Address Book provider
Replication RPC	RPC interfaces used by Active Directory replication over IP transport for replication within sites and between sites.
Replication Simple Mail Transfer Protocol (SMTP)	Replication protocol used by Active Directory replication over IP transport for message-based replication between sites only.

For more information about RPC, see "Windows 2000 Network Architecture" in the *TCP/IP Core Networking Guide*.

LDAP

LDAP is both a protocol and an API. It is also associated with both a directory service model that defines client/server mechanisms and an information model that defines the nature of objects stored in an LDAP directory service.

The LDAP protocol is the Active Directory core protocol, which means that LDAP is the only wire protocol that is supported by Active Directory. LDAP is the preferred and most common way of interacting with Active Directory. The LDAP API provides access to the LDAP protocol, and ADSI is the COM interface to Active Directory that uses LDAP as the protocol.

Note LDAP is a *wire protocol*, which means that it manages the encapsulation and sending of requests between a client and server.

LDAP Protocol

LDAP is a directory service protocol that specifies directory communications. It runs directly over Transmission Control Protocol/Internet Protocol (TCP/IP) and can also run over user datagram protocol (UDP) connectionless transports. LDAP enables clients to query, create, update, and delete information stored in a directory service over a TCP connection through the TCP default port 389. LDAP was used initially as a front end to X.500 directories. LDAPv3 is an industry standard that can be used with any directory service, such as Active Directory, that implements the LDAP protocol. Active Directory supports LDAPv2 (RFC 1777) and LDAPv3 (RFC 2251).

Note Windows 2000 Active Directory 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.

For more information about TCP/IP, see the *TCP/IP Core Networking Guide*.

LDAP Directory Service Model

The LDAP directory service is based on a client/server model. One or more LDAP servers contain the data making up the directory tree. An LDAP client connects to an LDAP server and requests information or performs an operation. The server performs the operation or provides the information, or it refers the client to another LDAP server that might be able to do so. When connecting to a specific LDAP directory tree, it does not matter what LDAP server a client connects to; a name presented to one LDAP server references the same object (referred to as an *entry* in LDAP) that it would reference at another LDAP server. This is an important feature of a global directory service.

LDAP Information Model

The LDAP information model is based on the entry, which contains information about some object (for example, a person or computer). In Active Directory, an LDAP entry is referred to as an object. Entries are composed of attributes, which have a type and one or more values. Each attribute has a syntax that determines what kind of values are allowed in the attribute. Examples of attribute syntaxes are Unicode string, binary, and integer.

The following key aspects characterize the LDAP protocol:

- The protocol is carried directly over TCP for connection-oriented transport (receipt of data is acknowledged) and UDP for connectionless transport (no acknowledgment on sending or receiving data).
- Most protocol data elements can be encoded as ordinary strings (for example, distinguished names).
- Referrals to other servers can be returned to the client.
- Simple Authentication and Security Layer (SASL) mechanisms can be used with LDAP to provide associated security services.
- Attribute values and distinguished names can be internationalized through the use of the International Standards Organization (ISO) 10646 character set.
- The protocol can be extended to support new operations, and controls can be used to extend existing operations.
- The schema is published through an attribute on the root object (rootDSE) for use by clients. (For more information about the schema, see "Active Directory Schema" in this book. For more information about rootDSE, see "RootDSE" later in this chapter.)

For more information about the LDAPv3 protocol, see the Request for Comments (RFC) link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. Follow the links to RFC 2251 ("Lightweight Directory Access Protocol (v3)," the original LDAPv3 description), RFC 2252 ("Attribute Syntax Definitions"), RFC 2253 ("UTF-8 String Representation of Distinguished Names"), RFC 2254 ("The String Representation of LDAP Search Filters"), RFC 2255 ("The LDAP URL Format"), RFC 2256 ("A Summary of the X.500[96] User Schema for Use with LDAPv3"), and RFC 2247 ("Using Domains in LDAP/X.500 Distinguished Names"). For more information about the LDAP RFCs, see "LDAP Requests for Comments" in this book.

Differences Between LDAPv2 and LDAPv3

LDAPv3 supports the following implementations that were not supported in LDAPv2:

- The use of UTF-8 for all text string attributes to support extended character sets. Active Directory sends all responses in UTF-8 encoded form.
- Operational attributes that the directory maintains for its own use (for example, to log the date and time when another attribute is modified).
- Referrals, which allow a server to direct a client to another server that might have the information the client is requesting. An LDAP server can return a referral to an LDAP client when the operation presented by the client cannot be serviced locally and the LDAP server has knowledge of other LDAP servers that can handle the operation.
- Schema publishing with the directory, which allows a client to discover what object classes and attributes a server supports.
- Extended searching operations that allow paging and sorting of results and client-defined searching and sorting controls.
- Stronger security through an SASL-based authentication mechanism.
- Extended operations, which provide additional functionality without changing the protocol version.

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

LDAP API

Unlike most other Internet protocols, the LDAP protocol has an associated API that simplifies writing Internet directory service applications. LDAP API is a C-language API to the LDAP protocol. RFC 1823 specifies the LDAP APIs that are required for a client to gain access to a directory service that supports the LDAP protocol. This API set is relatively simple and supports both synchronous and asynchronous calls to the server.

Microsoft implements the LDAP API in Wldap32.dll—also referred to as "LDAP C" or "C-binding LDAP." Applications that are written in LDAP are compatible only with LDAP directory services. ADSI, which provides a COM interface to Active Directory, is layered on top of LDAP and provides the easiest access to Active Directory through LDAP. However, Active Directory also fully supports the LDAP APIs for directory access.

For more information about the LDAP API and about programming in LDAP, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. For more information about LDAP referrals, see "Name Resolution in Active Directory" in this book. For more information about the schema, see "Active Directory Schema" in this book.

ADSI

The primary and recommended API for Active Directory is ADSI. ADSI enables access to Active Directory by exposing objects stored in the directory as COM objects. A directory object is manipulated using the methods on one or more COM interfaces. *ADSI providers* contain the implementation of ADSI objects for a particular namespace. By implementing the required interfaces, ADSI providers translate these interfaces to the API calls of a particular directory service.

ADSI LDAP Provider

The ADSI LDAP provider operates on the ADSI client to provide access to Active Directory or to other LDAP directory services. The ADSI LDAP provider works with any LDAP server that supports at least LDAPv2. In addition to Windows 2000 Active Directory, directory services that are accessible through the LDAP provider include the following:

- Netscape Directory Server.
- Exchange Server 5.*x*.
- Microsoft Commercial Internet System (MCIS) Address Book Server.
- University of Michigan Stand-alone LDAP Directory (SLAPD) Server.
- Other Internet directory servers (for example, Ldap.yahoo.com).

Note The WinNT ADSI provider enables access to Microsoft® Windows NT® version 3.*x* and Windows NT 4.0 directories, providing for communication with Windows NT 4.0 primary domain controllers and backup domain controllers. Other providers include NDS for access to Novell Directory Services directories, NWCOMPAT for access to Novell NetWare 3.*x* and Novell NetWare 4.*x* directories, and IIS for access to HTTP data directories used by Internet Information Services (IIS).

For more information about ADSI, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Active Directory Replication

Active Directory replication is performed over replication transport protocols, which are represented in the Active Directory architecture diagram (Figure 2.3) as REPL. For replication within a site, Active Directory replication uses RPC-over-IP transport protocols. For replication between sites, Active Directory replication uses two replication transport protocols: IP (RPC over IP) and Simple Mail Transfer Protocol (SMTP over IP).

Note The user interface that is associated with connection properties in Active Directory Sites and Services displays **RPC** for all connections within a site, and displays either **IP** (for RPC over IP) or **SMTP** (for SMTP over IP) for connections between sites. This convention is used to distinguish between RPC over IP for connections that are between sites and those that are within a site.

RPC replication between sites can be scheduled and is compressed. For replication within a site, RPC is always used. RPC replication within a site is not compressed. Thus, Windows 2000 directory replication recognizes three degrees of connectivity:

- Uniform, high-speed connectivity (RPC over IP for replication of all directory partitions within a site).
- Point-to-point, synchronous, low-speed connectivity (RPC over IP for replication of all directory partitions between sites).
- Mail-only, asynchronous connectivity (SMTP over IP for replication of only non-domain directory partitions between sites).

On each DSA, replication uses a single thread to receive changes from other servers and applies them locally by using either RPC synchronous transport or asynchronous transport for messaging between sites. The choice of transport is determined by the corresponding connection object (class *nTDSConnection*). Connection objects are created automatically by the Knowledge Consistency Checker (KCC). You can also create connection objects manually by using Active Directory Sites and Services. Both synchronous and asynchronous transports operate on a request-response basis.

For more information about Active Directory replication, see "Active Directory Replication" in this book.

MAPI

Messaging clients gain access to the Exchange Server directory service by using MAPI address book providers. For compatibility with existing messaging clients, Active Directory supports the MAPI-RPC address book provider, which allows access to Active Directory (for example, to find the telephone number of a user).

Note In Windows 2000, the MAPI address book provider is provided solely for backward compatibility with MAPI clients, such as Outlook.

SAM

SAM is a protected subsystem that manages user and group account information. In Windows 2000, workstation security accounts are stored by SAM in the local computer registry, and domain controller security accounts are stored in Active Directory. In Windows NT 4.0, both local and domain security accounts are stored in the registry.

Using SAM in Mixed Mode and Native Mode

Windows 2000 supports Win32 security APIs in both mixed mode and native mode.

In mixed-mode domains, where Windows NT 4.0–based backup domain controllers are still in use, SAM clients that run Microsoft® Windows NT® version 3.51 or Windows NT 4.0 communicate with the SAM server through SAM APIs, which are required for replication and for authentication against the SAM database.

In native-mode domains, there are no Windows NT 4.0 domain controllers, but there can be clients that run Windows 95, Windows 98, Windows NT 3.*x*, or Windows NT 4.0. These clients continue to authenticate by using the same SAM APIs.

SAM Client and Server Operations

Most SAM operations are structured as reads and writes of properties. For workstation accounts, operations are reads from and writes to the registry. Domain-account operations are performed on Active Directory objects and their corresponding properties, which are stored as column values in the directory database. The SAM client calls public SAM routines, which in turn call internal routines that encapsulate the RPC. On the server side, the internal SAM routines do the bulk of the work.

In Windows NT 4.0, all access to account information is accomplished through internal SAM routine calls to the accounts database that is stored in the registry. In Windows 2000, the SAM server effectively splits off the domain controller account information from the workstation account information and places it in Active Directory instead of in the registry. The logic in Samsrv.dll manages the security principal database differently, depending on the role of the computer. On a domain controller, Samsrv.dll uses Active Directory for security principal storage. On all other Windows 2000–based computers, Samsrv.dll uses the SAM database in the registry for storage.

Gaining access to Windows 2000 domain controller account information is accomplished by routines that are implemented as part of the DSA process on the server. These routines are called in-process on the server and offer the ability to search for, read, and write directory service objects.

Figure 2.4 illustrates the interactions between the SAM client and server processes and the storage of domain and local accounts. The SAM server depicts the logic applied by Samsrv.dll in the domain controller case (Directory API), where the accounts are domain accounts, and in all other cases (Registry API), where the accounts are local to the computer.

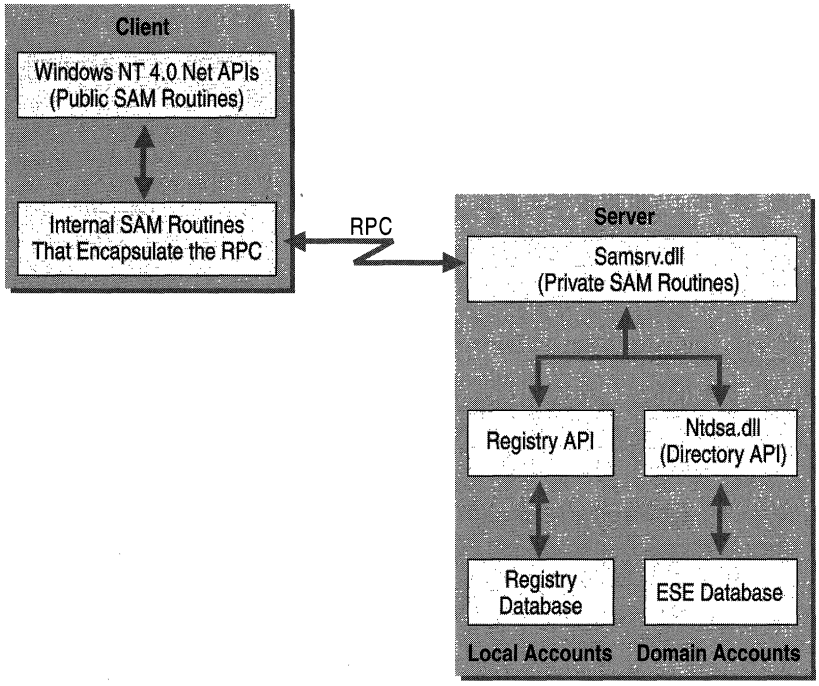


Figure 2.4 SAM Client and Server Interactions and Account Storage

Data Storage

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. Computers that store Active Directory are called domain controllers.

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 for 100s of thousands of clients and to provide availability, each logical partition replicates its changes separately among those domain controllers in the forest that store copies (replicas) of the same directory partitions.

Some directory partitions store forestwide configuration information and schema information; other directory partitions store information that is 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. In this way, Active Directory provides a data repository that is logically centralized but physically distributed. Because all domain controllers store forestwide configuration and schema information, a domain controller in one domain can reference a domain controller in any other domain if the information that it is requesting is not stored locally. In addition, domain controllers that are Global Catalog servers store a full replica of one domain directory partition plus a partial replica of every other domain in the forest. Thus, a domain controller that is a Global Catalog server 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:

Domainwide Data

- 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.

Forestwide Data

- Forestwide 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 that are 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.

Note When Active Directory is first installed on a computer that is 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.

Data Characteristics

The key characteristics of the data that is stored by a directory service correspond to size and latency. Active Directory should store objects that are not so large that they hamper replication and not so unstable that they change before an update replicates to all replicas in the forest. Therefore, large, unstructured data sets and data values that change frequently are not appropriate for storage in Active Directory.

In general, Active Directory is appropriate for the storage of data that has the following characteristics:

- The data is globally useful information in the domain that needs to be replicated to each Active Directory domain controller.
- The data has well-defined object attributes and semantics.
- The data has a useful life that is at least two times the maximum replication latency for the forest (to include replication of data that is marked to replicate to the global catalog). In general, if data can become outdated before the completion of a replication cycle or shortly thereafter, it should not be stored in Active Directory. Clients should be able to tolerate the inability to update data for at least as long as it takes for the data to be replicated throughout the domain.

- The data-per-attribute value is not so large that it affects performance. An attribute value is replicated as a single block of data; therefore, an attribute that is *x* megabytes in size requires an equivalent amount of buffer space in the sending and in the receiving domain controllers. If the amount of required buffer space is large, the performance of the domain controller can be adversely affected.

Storage Limits

There are no practical limits to the number of objects stored in Active Directory. The Active Directory database has been tested for up to 40 million objects. Performance tests show logon performance for a single LDAP client to be the same with 10,000 objects, 100,000 objects, and 1 million objects—that is, the directory service does not slow measurably when the size of the database increases.

Note In a mixed-mode environment in which backup domain controllers are running Windows NT 4.0, the recommended limit for the number of security principal objects per domain is 40,000 (the sum of users, groups, and computers). This limit is based on the Windows NT 4.0 SAM database storage capacity. (For more information about SAM database capacity, see "Determining Domain Migration Strategies" in the *Microsoft Windows 2000 Server Resource Kit Deployment Planning Guide*.)

Object Size vs. Maximum Database Record Size

Each object in the directory is represented as one record, or row, in the database, and each attribute is represented as one column in the row. The only exceptions are certain attributes whose values are stored separately as links. The limit for record size in the database is 800 non-linked values across all attributes. Attributes that represent links do not count in this value. (For more information about linked attributes, see "Linked Attributes" later in this chapter.) The size of objects is not a problem if you use the recommended guidelines described in "Data Characteristics" earlier in this chapter.

Note To enhance performance on domain controllers, install the Windows 2000 operating system on one drive, the Active Directory database file (Ntds.dit) on a second drive, and the log files on a third drive. (For more information about disk management, see "Data Storage and Management" in the *Microsoft Windows 2000 Server Resource Kit Server Operations Guide*. For more information about database page sizes, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.)

Garbage Collection

Instead of deleting objects physically from the database, the directory service removes most of the attributes and then tags the object as being in the *tombstone* state, which means it has been logically deleted from the directory but has not yet been completely removed. The tombstone tag alerts replication partners that the object was deleted. Objects that are tagged as tombstones are moved to the Deleted Objects container, where they remain until garbage collection removes them. Thus, tombstones are used to replicate object deletions.

Garbage collection is a housekeeping process that runs on every domain controller. At regular intervals (by default, 12 hours), garbage collection deletes objects that are no longer needed by the directory service.

Garbage collection performs the following tasks:

- Deletes tombstones.
- Defragments the database file.

There are two values that control how garbage collection runs and what it removes. These values are attributes of the cn=Directory Service,cn=Windows NT,cn=Services,cn=Configuration,dc=*ForestRootDomain* object:

- *Tombstone lifetime* determines the number of hours that a deleted object lives as a tombstone in the directory before being collected as garbage, and it is set in the *tombstoneLifetime* attribute. The default setting is 60 days, and the minimum setting is 2 days.
- *Garbage collection interval* determines how often a domain controller examines its database for expired tombstones that can be collected, and it is set in the *garbageCollPeriod* attribute. The default setting is 12 hours, and the minimum setting is 1 hour. This period is to ensure proper replication of deleted objects.

Note The default value for these two attributes applies if the attribute is not set (the initial state of the system). The minimum value applies if the attribute is set to a value below the minimum (that is, the minimum is not declared in the schema).

It is important that the tombstone lifetime be substantially longer than the expected replication latency. The interval between cycles of deleting tombstones must be at least as long as the maximum replication propagation delay across the forest. Because the expiration of a tombstone lifetime is based on the time when an object was deleted logically—rather than on the time when a particular server received that tombstone through replication—an object's tombstone is collected as garbage on all servers at approximately the same time. If the tombstone has not yet replicated to a particular server, that server never records the deletion. Likewise, if you restore a domain controller from a backup that is older than the tombstone lifetime, the domain controller does not have a record of some deletions, which leads to inconsistencies between domain controllers.

The maximum garbage collection interval is one-third of the tombstone lifetime (in hours). So if you set *tombstoneLifetime* to 30 days and *garbageCollPeriod* to 300 hours, your actual garbage collection period is only 10 days, or 240 hours.

You can use ADSI Edit to view or change the default settings for these attributes. To change the values, use the procedure that follows.

Note To use ADSI Edit, install the Support Tools that are located in the Support\Tools folder on the Windows 2000 Server operating system CD. To install the tools, double-click the **Setup** icon in that folder. For more information about using ADSI Edit, see *Microsoft Windows 2000 Support Tools* Help. For information about installing and using the Windows 2000 Support Tools and Support Tools Help, see the file Sreadme.doc in the Support\Tools folder of the Windows 2000 operating system CD.

▶ **To view or change attribute values by using ADSI Edit**

1. On the **Start** menu, point to **Programs**, **Windows 2000 Support Tools**, **Tools**, and then click **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 type 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. Navigate to 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 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**.

When you view properties on cn=Directory Service,cn=Windows NT, cn=Services,cn=Configuration,dc=*forestRootDomain*, if no value is set (which means that the default is in effect), the value that you type in the **Edit Attribute** box replaces the default value when you click **Set**.

For more information about backing up and restoring Active Directory, see "Active Directory Backup and Restore" in this book. For more information about replication, see "Active Directory Replication" in this book.

Database Defragmentation

To update the directory database file, the database system uses the quickest way to fill database pages. Although this system is efficient in updating the database quickly, it does not make the most efficient use of space in the database. *Defragmentation* rearranges how the data is written in the database in order to compress the data. You can defragment the database file online or offline by using the Ntdsutil command-line tool. Defragmentation can take place online (while the computer is running as a domain controller) or offline (while the computer is running as a stand-alone server).

Online Defragmentation

ESE supports online defragmentation, which effectively rearranges pages within the data file but does not release space back to the file system. ESE invokes online defragmentation automatically at regular intervals after garbage collection. Online defragmentation makes space available, but it does not reduce the size of the database file. Only offline defragmentation provides you with a clear picture of the amount of space consumed by the database file.

Offline Defragmentation

To release space back to the file system, you can perform offline defragmentation. Offline defragmentation must be performed in Directory Services Restore Mode, which restarts the computer as a stand-alone server—that is, the computer runs offline and is not acting as a domain controller. In Directory Services Restore Mode, you can use the Ntdsutil command-line tool to defragment the Ntds.dit file. Offline defragmentation produces the defragmented version of the database file in a separate directory. You can archive the original Ntds.dit file and move the defragmented file into the current directory. (For more information about using Ntdsutil to perform offline defragmentation, see "Active Directory Diagnostics, Troubleshooting, and Recovery" and "Active Directory Diagnostic Utility (Ntdsutil.exe)" in this book, and see *Microsoft Windows 2000 Support Tools* Help.)

You can use offline defragmentation to test database growth by comparing the defragmented version of the file with the fragmented version. For example, on a newly installed domain controller, if you perform a bulk load of objects and then defragment the database file offline, the difference between the two files is the space occupied by the new objects.

You can set the DSA to log, during garbage collection, a message in the Directory Service event log that states how much disk space might be freed up by offline defragmentation. To activate logging of this message in the Directory Service event log, edit the value of the **Garbage Collection** registry entry in HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS\Diagnostics.

▶ **To activate logging of disk space that is freed by defragmentation**

1. On the **Start** menu, click **Run**.
2. Type the following:

   regedt32.exe

   –Or–

   regedit.exe
3. Click **OK**.
4. In the registry editor, navigate to HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS\Diagnostics.
5. Double-click the **Garbage Collection** entry.

6. In Regedt32.exe, type **1** in the **Data** box, type:

   1

   –Or–

   In Regedit.exe, type **1** in the **Data value** box, type:

   1

7. Click **OK**, and then close the registry editor.

Caution Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. There are programs available in Control Panel or Microsoft Management Console (MMC) for performing most administrative tasks. These programs provide safeguards that prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your system. Registry editors bypass the standard safeguards that are provided by these administrative tools. Modifying the registry is recommended only when no administrative tool is available. Before you make changes to the registry, it is recommended that you back up any valuable data on the computer. For instructions about how to edit registry entries, see Help for the registry editor that you are using. For more information about the registry, see the *Microsoft Windows 2000 Resource Kit* Technical Reference to the Windows 2000 Registry (Regentry.chm).

To defragment the database file offline, start the domain controller in Directory Services Restore Mode.

▶ **To start a domain controller in Directory Services Restore Mode**

1. During the phase of startup when you would usually select the operating system, press F8 to display advanced startup options.
2. On the **Windows 2000 Advanced Options** menu, use the arrow keys to select **Directory Services Restore Mode**, and then press ENTER.

Follow these recommended defragmentation procedures:

- Use offline defragmentation only when you know that database contents have decreased considerably (for example, when a Global Catalog server becomes a normal domain controller) and you need to reclaim space for other uses.
- Retain the original Ntds.dit file until the domain controller has restarted with the defragmented file. When you have no doubt that the directory database is in a consistent state, you can delete the fragmented (original) database file.

For more information about performing offline defragmentation, see "Active Directory Diagnostics, Troubleshooting, and Recovery" in this book.

When you test the effects of loading a specific set of objects on database growth, keep the following in mind:

- When the directory contains objects that you have added recently, the database is in a fragmented state, which makes it impossible to tell how much space the objects actually occupy in the database file.
- Online defragmentation does not reduce the size of the database file. Only offline defragmentation accomplishes reduction of file size.
- If you have garbage collection event logging set to **1**, you can use the Directory Service log in Event Viewer to view messages about the amount of space that would be made available by offline defragmentation.

Growth Estimates for Active Directory Users and Organizational Units

A series of tests has been performed to arrive at average object sizes for objects that are typically stored in Active Directory. These tests begin with the size of the default Ntds.dit file immediately following the promotion of a domain controller.

Important Ntds.dit grows by increments and uses fractional pages in the Btrees to avoid page splits. Therefore, if you add objects, the database size seems larger than the actual data.

The database file size is not updated until the file has been closed. Because Active Directory always keeps its database open and the database file never shrinks while online, Windows 2000 always reports the larger file size when the domain controller is restarted. However, the free space that remains on the drive is reported correctly, so through subtraction you can reconstruct the actual file size.

To find the approximate size of data plus indices, set the registry to log disk space that is freed by defragmentation (see the procedure "To activate logging of disk space that is freed by defragmentation" earlier in this chapter) and use the Ntdsutil command-line tool to compact (defragment) the database offline. (For more information about using Ntdsutil to perform offline defragmentation, see "Active Directory Diagnostics, Troubleshooting, and Recovery" and "Active Directory Diagnostic Utility (Ntdsutil.exe)" in this book.)

Table 2.3 shows the sizes for users, organizational units, and attributes. You can use these numbers to perform additional calculations.

Table 2.3 Disk Space That Is Required for Storing Active Directory Objects

Object	Estimated size in database
User	3.7 KB
Organizational unit	1.1 KB
Attribute (10 bytes)	100 bytes

Directory Database Sizing Tests

In general, two kinds of directory database sizing tests were performed. In the first series of tests, the database was loaded with large numbers of identical objects to show how the database grows when objects are loaded and how much space is consumed for the object types in each test. In the second series of tests, a blueprint for a company with user objects, group objects, and file shares was created. Then objects were added in successive increments of 100,000 to show how large the database can become for small, medium, and large companies.

In most of the tests, the object load operation consisted of several steps. The empty database size was recorded, and the first set of objects was loaded. The database then was defragmented offline and the fragmented and the defragmented versions of the database measured. The fragmented version of the file was used when the next set of objects was loaded.

The objective of the single-object load tests was to determine how the size of the database increases with respect to an increase in attribute value size. Because the database engine consumes space only for attributes for which values are set, the number of object attributes that have values makes a significant difference in the size of an object. In tests that were conducted only for user and organizational unit objects, only mandatory attributes were set. Mandatory attributes must contain at least one value in order for the directory service to create the object. Subsequent tests show how adding attributes to an object affects object size.

Because user objects play a significant role in directory deployment, you have to know how big the user objects are to determine how big the database will grow when you add users. Up to 500,000 users were loaded in the database in increments of 100,000 users. The results of the test are shown in Table 2.4.

Table 2.4 User Object Growth Test Results

	Fragmented database			Defragmented database		
Number of users	**KB per database**	**Growth (in KB)**	**Bytes per user**	**KB per database**	**Growth (in KB)**	**Bytes per user**
0	10,256	--	--	10,256	--	--
100,000	516,064	505,808	5,179	364,560	354,304	3,628
200,000	899,088	383,024	4,551	720,912	356,352	3,639
300,000	1,294,352	395,264	4,383	1,079,312	358,400	3,649
400,000	1,675,280	380,928	4,262	1,435,664	356,352	3,649
500,000	2,060,328	385,048	4,199	1,792,016	356,352	3,649

The database file growth pattern shown in Table 2.4 is linear. In fact, the growth between two load operations is always almost identical—approximately 385,000 KB in the fragmented version of the file and approximately 356,000 KB in the defragmented version of the file. The only exception to this linear growth pattern is the load operation in the first step when the first 100,000 users are loaded. Here the increase in file size is much greater (about 516,000 KB).

This test illustrates how online defragmentation affects database size. Between the time the first set of objects (100,000 users) was added on the first day and the time the size of the file was checked on the next day, online defragmentation had occurred and rearranged the objects in the database. The online defragmentation made space available for new objects, but it did not shrink the database file size. Therefore, the size of the fragmented file is much greater than the size of the file when the next set of 100,000 user objects is loaded and the space in the database file can be used. The increase in file size for the fragmented version of the file is not much different from the increase in file size for the defragmented version of the file.

Loading 500,000 users (with values for only the mandatory attributes set) into Active Directory requires about 1.8 GB. To compute the size of one user object, subtract the size of an empty store and then divide the store size by the number of users. In this case, one user object is 3,649 bytes in size. User objects in Active Directory are larger than most other objects because they contain many mandatory attributes.

Organizational Units

Because information in Active Directory is organized in a hierarchy of organizational units, a series of tests was conducted to measure the effect of adding organizational unit objects on the database size. Table 2.5 shows the results of these tests.

Table 2.5 Organizational Unit Object Growth Test Results

	Fragmented database			Defragmented database		
Number of users	**KB per database**	**Growth (in KB)**	**Bytes per user**	**KB per database**	**Growth (in KB)**	**Bytes per user**
0	10,256	--	--	10,256	--	--
2,000	12,304	2,048	1,049	10,256	0	--
4,000	16,440	4,136	1,583	12,304	2,048	524
6,000	18,448	2,008	1,398	14,352	2,048	699
8,000	20,496	2,048	1,311	18,448	4,096	1,049
10,000	24,592	4,096	1,468	20,496	2,048	1,049
12,000	26,640	2,048	1,398	22,544	2,048	1,049
14,000	28,688	2,048	1,348	24,592	2,048	1,049
16,000	32,784	4,096	1,442	26,640	2,048	1,049

The results in Table 2.5 show that the growth pattern for organizational units also is linear. The size of one organizational unit is 1,049 bytes.

Adding Attributes

The next series of tests enlarged the user objects with extension attributes 1 through 10. The attributes are defined in the schema as string-valued attributes. Each string was filled with 10 characters.

The test began with a store that contained 100,000 user objects with only their mandatory attribute set. The server was demoted to a non-domain controller, promoted back to a domain controller, and then loaded with 100,000 user objects again, this time with one extension attribute. The process was repeated with two attributes, then three attributes, and so forth. The results are shown in Table 2.6.

Table 2.6 Test Results for Extension Attributes with a User Object

	Fragmented database		Defragmented database		
Number of extension attributes	KB per database	Bytes per user	Database/ defrag.	Bytes per user	Bytes per attribute
0	522,256	5,242	364,560	3,627	--
1	413,712	4,130	364,560	3,627	0
2	413,712	4,130	364,560	3,627	0
3	485,392	4,864	382,992	3,816	63
4	663,568	6,689	405,520	4,046	105
5	698,384	7,045	405,520	4,046	84
6	706,576	7,129	407,568	4,067	73
7	704,528	7,108	444,432	4,445	117
8	702,480	7,087	444,432	4,445	102
9	497,680	4,990	444,432	4,445	91
10	497,680	4,990	444,432	4,445	82
11	497,680	4,990	444,432	4,445	74

In the fragmented version of the database, the size of the database increases by significantly larger increments at some steps than at others. This variation occurs when the database engine allocates space for new pages. Such large increases do not occur for the defragmented version of the database.

The growth of the defragmented database again is very linear. One additional attribute with a string size of 10 characters adds approximately 100 bytes to an object's size.

For more information about Active Directory growth management and capacity planning, see "Designing the Active Directory Structure" in the *Deployment Planning Guide*, and see the Microsoft Windows 2000 Server link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Windows 2000 SAM Storage

In Windows NT 4.0, both domain controllers and workstations store security principal accounts in the SAM database, which uses the registry as its underlying persistent storage. In Windows 2000, domain security principal accounts are stored in Active Directory instead of the registry. Although security accounts are stored in Active Directory, SAM is retained on Windows 2000 domain controllers for compatibility with those domains and applications that depend on it. SAM also is used by Windows 2000–based computers that are not domain controllers for local account storage. Thus, SAM manages security principal accounts. It uses Active Directory for storage of these accounts on a domain controller, and it uses the SAM database in the registry on workstations, stand-alone servers, and member servers. SAM (Samsrv.dll) provides a simple form of name resolution, minimal transactions, replication, and secure storage for the security database.

In Windows 2000, there are two types of accounts: *workstation accounts* and *domain accounts*. Workstation accounts, which include user and group accounts on workstations, member servers, and stand-alone servers, are limited in scope to the physical computer where the accounts reside. A domain account has a broader scope than a workstation account; it extends to all physical computers within the domain. A workstation administrator, for example, has administrative privileges on the local computer (a workstation or member server), but a domain administrator has administrative privileges on all computers within the domain.

In Windows NT 3.51 and Windows NT 4.0, both categories of accounts are stored in the SAM database (in the registry). In Windows 2000, domain controllers store domain user, group, and computer accounts only in Active Directory; workstations and member servers continue to store local accounts in the SAM database. On Windows 2000 domain controllers, the existing SAM database is deleted and replaced by a new registry key that stores a small SAM database, which is used principally for Directory Services Restore Mode. When you start a domain controller in Directory Services Restore Mode, the SAM registry database is used for the security principal database instead of Active Directory.

In addition, Windows 2000 SAM supports the following:

- Multimaster account replication among peer domain controllers
- Creation and deletion of user properties
- Read, write, and query third-party properties as defined by supplemental security packages in the LSA. (For more information about the LSA, see "Access Control" in this book.)

Domain controllers that are running Windows 2000 Server are completely compatible with domain controllers that are running Windows NT 4.0—that is, a Windows NT 4.0–based client can be authenticated by a Windows 2000–based domain controller, and a Windows NT 4.0–based backup domain controller can continue to replicate with Windows 2000–based domain controllers. In a Windows 2000 domain, a Windows 2000–based domain controller can be configured to assume, or "emulate," the role of a primary domain controller (the PDC emulator flexible single-master operation role).

For more information about the PDC emulator role of a Windows 2000 domain controller, see "Managing Flexible Single-Master Operations" in this book, see "Determining Domain Migration Strategies" in the *Deployment Planning Guide*, and see Windows 2000 Server Help.

Mixed-Mode Storage Considerations

In mixed mode, account storage capacity is limited by the SAM database, which is still used for domain accounts on the backup domain controllers. A Windows NT 4.0–based backup domain controller is able to store approximately 40,000 security principal accounts (users, groups, and computers). The SAM database size does not decrease when you delete objects, but the database becomes fragmented and contains "empty" space. This empty space is reclaimed as new objects are added, which can result in less available storage than the number of accounts might indicate. For example, changing group membership leaves an unoccupied storage space for the membership that was removed.

Note Running Regback against the SAM database can remove the spaces, but only if the physical RAM of the computer is at least twice as large as the current SAM (because of the way Regback works). For information about techniques for compressing the SAM database, see the Knowledge Base link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. Search the Knowledge Base using the keywords "database" and "shrink."

SAM Structure

The Windows NT 4.0 and Windows 2000 SAM both contain collections of domain security accounts. A "domain" in the SAM sense can refer either to all of the accounts on a single computer or all of the accounts in a Windows domain. The Builtin container contains default local group accounts (such as Administrators and Users) that are installed whenever a new workstation, server, or domain controller is set up. It provides some basic account types, such as Administrator and Guest, that give the operator sufficient capability to add further accounts to the computer or domain. The Builtin container account SIDs are the same on every Windows 2000 or earlier system. These fixed SIDs allow the predefined groups to be placed in access control lists without regard to the domain of the system. For this reason, the objects in the Builtin container cannot be changed.

In Windows 2000, domains continue to contain the same objects as in Windows NT 4.0, as well as several additional properties on certain objects.

SAM Accounts on a Windows 2000 Server That Becomes a Domain Controller

When you install Active Directory on a computer that is running Windows 2000 Server to create a domain controller, you can either create a new domain or configure the domain controller to contain a copy of an existing domain. In both cases, the existing registry key that contains the SAM database is deleted and is replaced by a new, smaller SAM database. The security principals in this database are used only when the server is started in Directory Services Restore Mode.

The disposition of the security principals in the SAM database on the server is different in each case, as follows:

- If you create an additional domain controller in an existing domain, the security accounts in the existing SAM database on the server are deleted. The accounts from the existing domain are replicated to Active Directory on the new domain controller.
- If you create a new domain, the security accounts in the existing SAM database are preserved as follows:
  - User accounts become user objects in Active Directory.
  - Local groups in the account domain become group objects in Active Directory. The group type indicates a local group.
  - Built-in local groups become group objects in Active Directory. The group type indicates a built-in local group. These groups retain their constant SIDs and are stored in the Builtin container.

Migration of Windows NT 4.0 SAM Accounts to Active Directory Objects

When a Windows NT 4.0 domain controller is upgraded to Windows 2000, SAM security accounts are migrated to Active Directory objects. The relationship between Windows NT 4.0 accounts and Windows 2000 Active Directory objects is as follows:

- "Normal" user accounts, which represent people, are stored as objects of the class *user* in Active Directory.
- Computer user accounts (called "machine accounts" in Windows NT 4.0), which represent devices, are stored as objects of the class *computer*, which is a derived class of *user* and is exposed as the base class *user* to clients and domain controllers that are running earlier versions of Windows. (For more information about derived classes and base classes, see "Active Directory Schema" in this book.) By default, these accounts are placed in the Computers container after an upgrade, although there is no restriction that requires computer accounts to be confined to the Computers container. A control flag on the user account identifies the account type as a server or workstation, domain controller, or normal user account. Windows 2000 maintains the Windows NT 4.0 semantics (the flags that determine the nature of objects—for example, a computer versus a user object) for workstation accounts.

Note In Active Directory Users and Computers, the **Role** property ("attribute") on computer accounts indicates the account type. This property represents the userAccountControl flag value on the *machineRole* property of 4096 for a server or workstation or 8192 for a domain controller.

- Global group accounts are stored as group objects in Active Directory.
- Local group accounts from the SAM account domain are stored as group objects in Active Directory.
- Built-in local group accounts from the SAM Builtin domain (for example, the Administrators group) are stored as domain local group objects in Active Directory in the Builtin container. Groups from the SAM Builtin domain have constant SIDs.
- Backup domain controller computer accounts are represented identically to workstation computer accounts, except that a different flag is set to distinguish them.

- LSA account objects grant privileges on the workstation computer to a particular account. They are maintained in the registry and synchronized between the domain controllers by being replicated to the workstation policy. By default, each domain controller in the domain has the same workstation policy. Therefore, a change to an LSA account object updates the corresponding workstation policy for the PDC emulator. The workstation policy change replicates to every other Windows 2000 domain controller in the domain.

The upgrade from a given Windows NT 4.0 SAM account to the corresponding Windows 2000 Active Directory object is summarized in Table 2.7.

Table 2.7 Upgrade of Windows NT 4.0 Accounts to Windows 2000 Active Directory Objects

Windows NT 4.0 SAM	Windows 2000 Active Directory
Normal user account	User object.
Computer user account	Computer object, where the user account control flag indicates a workstation trust account.
Domain controller account	Computer object, where the user account control flag indicates a server trust account.
Global group in an account domain	Group object, where the group type indicates a global group.
Local group in an account domain	Group object, where the group type indicates a local group.
Local group in the Builtin domain	Group object, where the group type indicates a local group as well as Builtin group (for example, Administrators, Backup Operators, and so forth).
Domain trust account	Trusted domain object. (Assumes the role of both inbound and outbound halves of the trust relationship; there is also a domain trust account of class *user* for backward compatibility.)
Trusted domain object	Trusted domain object, upgraded.

For more information about upgrade issues, see "Determining Domain Migration Strategies" in the *Deployment Planning Guide*.

Data Model

The Active Directory data model is derived from the X.500 model of objects and attributes. An *object* is a distinct, named set of attributes that represents something concrete, such as a user, a printer, or an application. Thus, Active Directory holds objects that represent entities of various sorts, which are described by *attributes* (also called "properties"). For example, attributes of a user object might include the user's given name, surname, and e-mail address.

The universe of objects that can be stored in Active Directory is defined in the schema. For each object class, the schema defines what mandatory attributes an instance of the class is required to have, what optional additional attributes it can have, and what object class can be a parent of the current object class. LDAP defines the protocol that is used for accessing and modifying directory information.

Note Active Directory is not an X.500 directory; as such, it does not support X.500 protocols.

Container Objects and Leaf Objects

A leaf object is an object that has no child objects. The term "container" refers to one of two things:

- An object of the *container* structural class.
- An object that has child objects.

Container is a *structural class* of object, which means that container objects can be created in Active Directory. In the schema, structural classes define objects that can be created as instances of the class in Active Directory. Other objects can be "container" objects in the general sense of the word (that is, they can have child objects), but they do not belong to the *container* class. For example, an organizational unit is a container object, although its class is *organizationalUnit*, not *container*. An organizational unit object has many attributes that provide functionality that an ordinary container does not have.

For more information about structural classes, see "Active Directory Schema" in this book.

Directory Tree

The *directory tree* represents the hierarchy of Active Directory objects for a given forest. The hierarchy provides the basis both for using names for navigation and for defining the scope of search requests.

For every object in Active Directory, information is stored in the directory database that identifies (references) the parent object; each object has exactly one parent. By virtue of these parent references, the hierarchy of objects managed by Active Directory forms a tree structure in which the vertices are the directory entries (class instances, or objects) and the connecting lines are the parent-child relationships between the entries. The objects that populate the directory create this tree structure according to the rules of the schema, which define what classes of objects are allowed to be created in which positions relative to other objects. For example, the schema might dictate that a given class of object can be the child of one class but not the child of another class.

The following are several architectural restrictions and requirements within the directory tree:

- Domain objects, which are containers, can be children only of other domain objects. For example, a domain cannot be the child of an organizational unit.
- The root of the directory tree is called *rootDSE*, or *directory root*. RootDSE is an "imaginary" object that has no hierarchical name or schema class, but it does have a set of attributes that identify the contents of a given domain controller. Thus, rootDSE constitutes the root of the directory tree from the perspective of the domain controller to which you are connected.
- Below the root of the tree, every directory has a *root domain*, which is the first domain created in a forest. This domain always has a child container called Configuration, which contains configuration data for the forest. The configuration data includes information about all services, sites, and other domains (partitions) in the forest. The Configuration container has a child container called Schema. The domain and the Configuration container, with its child Schema container, represent the three default Active Directory directory partitions.

For more information about parent-child relationships, see "Active Directory Schema" and "Active Directory Logical Structure" in this book.

RootDSE

The rootDSE (DSA-specific Entry) represents the top of the logical namespace for one domain controller, and, therefore, it represents the top of the LDAP search tree. There is only one root for a given directory, but the information stored in the root is specific to the domain controller to which you connect. The attributes of rootDSE identify both the directory partitions (the domain, schema, and configuration directory partitions) that are specific to one domain controller and the forest root domain directory partition. Thus, the rootDSE provides a "table of contents" for a given domain controller.

The rootDSE publishes information about the LDAP server, including what LDAP versions it supports, supported Simple Authentication and Security Layer (SASL) mechanisms, and supported controls, as well as the distinguished name for its *subschemaSubentry*.

The following are the operational attributes on the rootDSE object. All LDAP servers recognize these attribute names, but when the attribute corresponds to a feature that the server does not implement, the attribute is absent.

subschemaSubentry The name of a subschema entry, which is used to administer information about the schema; in particular, the object classes and attribute types that are supported. (For more information about *subschemaSubentry*, see "Active Directory Schema" in this book.)

namingContexts Naming contexts (directory partitions) that this server masters (stores as a writable replica) or shadows (stores as a read-only replica). This attribute allows a client to choose suitable base objects for searching when the client has contacted a server.

supportedControl Object identifiers that identify the LDAP controls that the server supports. If the server does not support any controls, this attribute is absent.

supportedSASLMechanisms The names of the SASL mechanisms that the server supports. SASL is a standard for negotiating an authentication mechanism and (optionally) an encryption mechanism. If the server does not support either type of mechanism, this attribute is absent.

supportedLDAPVersion The versions of LDAP that the server implements.

supportedExtension Object identifiers (known as "OIDs") that identify the supported extended operations that the server supports. If the server does not support any extensions, this attribute is absent. This attribute is absent by default for Active Directory servers.

altServer The values of this attribute are URLs of other servers that can be contacted when this server becomes unavailable. If the server does not know of any other servers, this attribute is absent. This attribute is absent by default for Active Directory servers.

In addition to the operational attributes described in the preceding paragraphs, Active Directory also supports the following informational attributes:

currentTime. The current time in the generalized time format.

dsServiceName. NTDS settings.

defaultNamingContext. The default naming context (directory partition) for a particular server. This value is the distinguished name of the domain directory partition for which this domain controller is authoritative.

schemaNamingContext. The naming context (directory partition) for the forest schema.

configurationNamingContext. The naming context (directory partition) for the forest Configuration container.

rootDomainNamingContext. The distinguished name for the domain naming context (directory partition) that is the first domain that was created in this forest. This domain functions as the forest root domain.

supportedLDAPPolicies. Supported LDAP management policies.

highestCommittedUsn. Highest update sequence number (USN) committed to the database on this domain controller. (For information about update sequence numbers, see "Active Directory Replication" in this book.)

dnsHostName. The DNS name of this domain controller.

serverName. The fully qualified distinguished name for this domain controller.

supportedCapabilities. The object identifier value (1.2.840.113556.1.4.800) that indicates the additional capabilities of an Active Directory server, such as dynamic update, integrated DNS zones, and LDAP policies.

LdapServiceName. The service principal name for the LDAP server, which is used for mutual authentication.

isSynchronized. Boolean indicator for whether the domain controller has completed its initial sync with replica partners.

isGlobalCatalogReady. Boolean indicator for whether the domain controller is prepared to advertise itself as a Global Catalog.

For more information about rootDSE and rootDSE attributes, see the Request for Comments (RFC) link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. Follow the links to RFC 2251 and RFC 2252.

You can use ADSI Edit or Ldp to see the contents of rootDSE for a given domain controller.

Note To use ADSI Edit and Ldp, install the Support Tools that are located in the Support\Tools folder on the Windows 2000 Server operating system CD. To install the tools, double-click the **Setup** icon in that folder. For more information about using ADSI Edit and Ldp, see *Microsoft Windows 2000 Support Tools* Help. For information about installing and using the Windows 2000 Support Tools and Support Tools Help, see the file Sreadme.doc in the Support\Tools folder of the Windows 2000 operating system CD.

▶ **To view rootDSE properties by using ADSI Edit**

1. In ADSI Edit, right-click the ADSI Edit icon, and then click **Connect to**.
2. To connect to a different domain controller from the default domain controller (the domain controller for the domain to which you are logged on), click **Select or type a domain or server**, and then type a domain name or server name.
3. Under **Connection Point**, click **Naming Context**.
4. In the **Naming Context** list, click **RootDSE** and then click **OK**.
5. Expand the **RootDSE [*ServerName*]** node.
6. Right-click the **RootDSE** folder, and then click **Properties**.
7. In the **RootDSE Properties** dialog box, view a property value by selecting the property in the **Select properties to view** box.

You can use ADSI Edit to view one rootDSE property value at a time. To view the entire list of properties and their values, use Ldp.

Ldp.exe is a graphical tool that you can use to perform LDAP operations, such as connect, bind, search, modify, add, and delete, against any LDAP-compatible directory, such as Active Directory. When you use Ldp to connect to a domain controller, the tool displays a list of the rootDSE attribute values that are stored on the domain controller to which you connect.

Note You can open Ldp in any of the following ways: from the **Windows 2000 Support Tools** menu by selecting **Active Directory Administration Tool**; from the **Run** dialog box by typing **ldp**; or from a command prompt by typing **ldp**.

▶ To connect to a domain controller and view rootDSE attributes by using Ldp

1. In Ldp, on the **Connection** menu, click **Connect**.
2. In the **Server** box, either use the current domain controller name or type the name of the domain controller to which you want to connect.
3. In the **Port** box, type the port number that you want to use.

   Port 389 is the default port for LDAP; port 3268 is the default port for the Active Directory Global Catalog.
4. Click **OK**.

The following printout shows the results of an Ldp Connect operation. The rootDSE information is displayed in the Ldp details pane.

```
ld = ldap_open("sea-rk-dc-01", 389);
Established connection to sea-rk-dc-01.
Retrieving base DSA information...
Result <0>: (null)
Matched DNs:
Getting 1 entries:
>> Dn:
1> currentTime: 10/1/1999 15:49:25 Pacific Standard Time Pacific
Daylight Time;
1> subschemaSubentry:
CN=Aggregate,CN=Schema,CN=Configuration,DC=reskit,DC=com;
1> dsServiceName: CN=NTDS Settings,CN=SEA-RK-DC-
01,CN=Servers,CN=Default-First-Site-
Name,CN=Sites,CN=Configuration,DC=reskit,DC=com;
3> namingContexts: CN=Schema,CN=Configuration,DC=reskit,DC=com;
CN=Configuration,DC=reskit,DC=com; DC=reskit,DC=com;
1> defaultNamingContext: DC=reskit,DC=com;
1> schemaNamingContext: CN=Schema,CN=Configuration,DC=reskit,DC=com;
1> configurationNamingContext: CN=Configuration,DC=reskit,DC=com;
1> rootDomainNamingContext: DC=reskit,DC=com;
16> supportedControl: 1.2.840.113556.1.4.319; 1.2.840.113556.1.4.801;
1.2.840.113556.1.4.473; 1.2.840.113556.1.4.528; 1.2.840.113556.1.4.417;
1.2.840.113556.1.4.619; 1.2.840.113556.1.4.841; 1.2.840.113556.1.4.529;
1.2.840.113556.1.4.805; 1.2.840.113556.1.4.521; 1.2.840.113556.1.4.970;
1.2.840.113556.1.4.1338; 1.2.840.113556.1.4.474;
1.2.840.113556.1.4.1339; 1.2.840.113556.1.4.1340;
1.2.840.113556.1.4.1413;
2> supportedLDAPVersion: 3; 2;
```

```
11> supportedLDAPPolicies: InitRecvTimeout; MaxConnections;
MaxConnIdleTime; MaxActiveQueries; MaxNotificationPerConn; MaxPageSize;
MaxQueryDuration; MaxTempTableSize; MaxResultSetSize; MaxPoolThreads;
MaxDatagramRecv;
1> highestCommittedUSN: 191396;
2> supportedSASLMechanisms: GSSAPI; GSS-SPNEGO;
1> dnsHostName: SEA-RK-DC-01.reskit.com;
1> ldapServiceName: reskit.com:sea-rk-dc-01$@RESKIT.COM;
1> serverName: CN=SEA-RK-DC-01,CN=Servers,CN=Default-First-Site-
Name,CN=Sites,CN=Configuration,DC=reskit,DC=com;
1> supportedCapabilities: 1.2.840.113556.1.4.800;
1> isSynchronized: TRUE;
1> isGlobalCatalogReady: TRUE;
```

Note The rootDSE attribute values also can be retrieved from an LDAPv3 server by using a base-level search with a null base distinguished name and with the filter (objectClass=*). (For more information about LDAP searches, see "Name Resolution in Active Directory" in this book.)

For more information about rootDSE, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. Search the SDK on the keyword "rootDSE".

Extended LDAP Controls

Windows 2000 supports several LDAP controls that extend the functionality of the LDAPv3 protocol. Microsoft has defined these LDAP controls to increase the functionality of Active Directory. These controls provide functionality that is not provided by current Internet Engineering Task Force (IETF) RFCs. The rootDSE indicates all controls that are in effect for the contacted server through the object identifier (also known as "OID") values in the *supportedControl* attribute.

Extended LDAP control functionality is useful to programmers who are using LDAP to perform directory operations. Some of the operations that can be implemented using extended controls are deleting trees, paging and sorting search results, and showing deleted objects. (For more information about showing deleted objects, see "Active Directory Name Resolution" in this book.)

For more information about using LDAP controls, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. Search the SDK using the keyword "LDAPControl" (one word).

Note LDAP control object identifiers are required only by the LDAP API. Most developers use ADSI, which uses other mechanisms, such as search preference flags, to achieve the same functionality. For more information about using ADSI, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Attribute Range Option

The LDAP protocol reads a multivalue attribute as a single entity, which can be inconvenient in the time that it takes when the number of values is large or, in some cases, makes reading the attribute impossible. The Range option can be specified as part of an attribute description to retrieve the values of a multivalue attribute incrementally. An attribute description includes an attribute type (for example, *member*) and a list of options, one of which can be the Range option. When presented in a searchRequest message, the Range option specifies a zero-relative range of elements (for example, 0-9) to be retrieved. By specifying the Range option followed by a range specifier, only the number of values in that range are retrieved.

To retrieve a range of values in Ldp, open a search (on the **Browse** menu, click **Search),** and then, in the **Search** dialog box, click **Options**. In the **Attributes** box, specify an attribute and the Range option. The attribute name and the Range option must be enclosed in quotation marks (" ").

For example, to read six members of a group at a time, use the group distinguished name as the search base and type the following in the **Attributes** box: **"member;range=0-5"**. This search will return six values for an object with multiple values in the member attribute.

For more information about using the Range option, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. Search the SDK using the keywords "range specifier" and "enumerating groups."

Directory Partitions

To scale to tens of millions of objects, a forest is partitioned into domains. Each Active Directory domain controller can be a member of one domain, and domain controllers within the same domain contain the same information. Domain controllers from different domains share the same configuration and schema data, but they do not share the same domain data. The means to distributing storage in this manner is the *directory partition*, which is also called a "naming context."

In Active Directory, a directory partition is a portion of the directory namespace. Each directory partition contains a hierarchy (subtree) of directory objects in the directory tree. The same directory partition can be stored as copies (replicas) on many domain controllers, and the copies are updated through directory replication.

Because a domain defines a security boundary and also represents the logical boundary of objects that belong to the same administration, each domain is mapped to a different directory partition so that the objects belonging to two different domains can be maintained and replicated independently. Likewise, information that is relevant to the entire forest is replicated separately.

Directory Partition Subtrees

Every domain controller contains the following three directory partitions:

Configuration Contains the Configuration container, which stores configuration objects for the entire forest in cn=configuration,dc=*forestRootDomain*. Updates to this container are replicated to all domain controllers in the forest. Configuration objects store information about sites, services, and directory partitions. You can view the contents of the Configuration container by using ADSI Edit.

Schema Contains the Schema container, which stores class and attribute definitions for all existing and possible Active Directory objects in cn=schema,cn=configuration,dc=*forestRootDomain*. Updates to this container are replicated to all domain controllers in the forest. You can view the contents of the Schema container in the Active Directory Schema console.

Domain Contains a <*domain*> container (for example, the Reskit.com container), which stores users, computers, groups, and other objects for a specific Windows 2000 domain (for example, the Reskit.com domain). Updates to the <*domain*> container are replicated to only domain controllers within the domain and to Global Catalog servers if the update is made to an attribute that is marked for replication to the Global Catalog. The <*domain*> container is displayed in the Active Directory Users and Computers console. The hierarchy of domain directory partitions can be viewed in the Active Directory Domains and Trusts console, where trust relationships between domains can be managed.

Each directory partition is a contiguous portion of the directory tree, and each one starts at a single point (the directory partition *head*) and spreads to either leaf nodes (for the schema and configuration directory partitions) or to the heads of other directory partitions below it (for domain directory partitions). Each directory partition, therefore, has exactly one directory partition immediately above it in the tree (except for a tree root domain directory partition, which has only the rootDSE above it) and possibly more directory partitions immediately below it. For domain directory partitions, this order manifests itself in the hierarchical infrastructure discussed in "Active Directory Logical Structure" in this book.

Note You cannot rename the topmost object in a directory partition, which means that you cannot rename a domain, the Schema container, or the Configuration container.

Directory Partition Hierarchy

There is an important distinction between the physical storage of a directory partition and its logical position in the directory tree. Physically, all objects are stored in a single database table, regardless of the directory partition to which they are assigned by virtue of their object names. Logically, the head of a directory partition appears in the naming hierarchy as the topmost object—that is, the <*domain*> container, the Configuration container, and the Schema container each has a distinguished name that identifies its position in the hierarchy. The respective directory partitions contain those objects, which are called "heads"; the domain directory partition contains an object named dc=*domainName*; the Configuration directory partition contains an object named cn=configuration,dc=*forestRootDomain*; and the schema directory partition contains an object named cn=schema,cn=configuration,dc=*forestRootDomain*.

Figure 2.5 shows a conceptual diagram of the directory tree hierarchy, including the directory root (rootDSE) and the default directory partitions below the directory root. In any Active Directory forest, the configuration directory partitions and schema directory partitions are always in these locations.

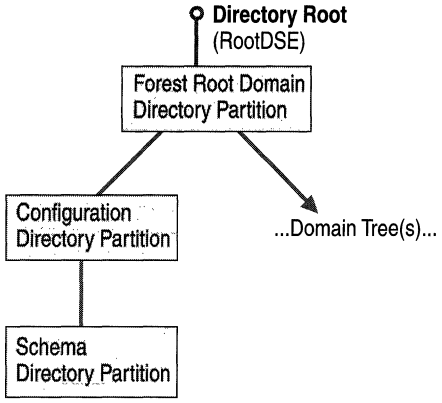


Figure 2.5 Default Active Directory Partitions

Each domain controller in the forest holds a master copy (replica) of the configuration and schema directory partitions, which are copied to a domain controller during domain controller promotion. All updates to configuration and schema directory partitions are replicated to every domain controller in the forest. In this way, site, service, domain, and schema information is kept consistent throughout the forest.

Forest Root Domain

Because the forest root domain is the first domain created in a forest, it is the root domain in the domain namespace hierarchy. In naming only, the topmost object of the configuration directory partition (the Configuration container) is the child of the forest root domain object in the hierarchy. The distinguished name of the Configuration container (cn=configuration,dc=*forestRootDomain*) reflects this naming hierarchy.

Although the distinguished name of the Configuration container indicates that it is a child of the forest root domain object, the configuration directory partition is not physically part of the forest root domain directory partition but is a separate directory partition that is replicated to all domain controllers in the forest; in contrast to the configuration directory partition, the forest root domain directory partition is replicated to only domain controllers in that domain. Similarly, the topmost object in the schema directory partition (the Schema container) is the child of the Configuration container. The distinguished name of the Schema container (cn=schema,cn=configuration,dc=*forestRootDomain*) shows the location of the schema to be within the forest root domain. Although the Schema container is a child of the Configuration container, the schema directory partition is not physically part of the configuration directory partition nor part of the forest root domain directory partition.

For more information about replication of directory partitions, see "Active Directory Replication" in this book.

Configuration Directory Partition

The configuration directory partition is created initially when the first Windows 2000 domain is created during the installation of Active Directory;thereafter, it is replicated to every domain controller in the forest. When a child domain or a new tree-root domain is created in the forest or when an additional domain controller is added to an existing domain, the configuration directory partition is copied to the new domain controller.

Viewing the Configuration Container

You can view the Configuration container by using ADSI Edit. When you open ADSI Edit, the Configuration container for the forest of the domain to which you are connected is displayed, along with the current domain directory partition and the Schema directory partition. Figure 2.6 illustrates the contents of the Configuration container as it is displayed in ADSI Edit.

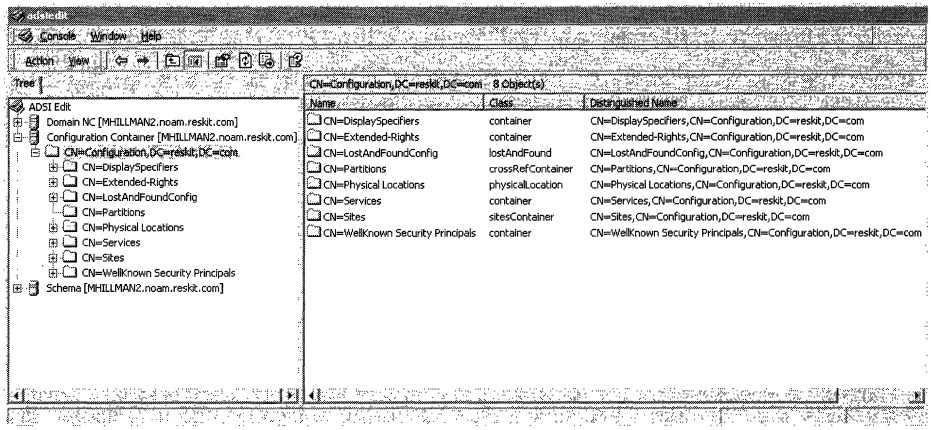


Figure 2.6 Contents of the Configuration Container

The following objects are child containers within the Configuration container.

DisplaySpecifiers Contains the objects that define different user interfaces for each object class in the schema that requires a graphical user interface (for example, context menus and property pages). The display specification system uses the information that is stored in the display specifiers to form different user interfaces for administrators and for end users. One set of elements, such as property pages, context menus, and so forth, can be associated with administrative applications, and a different set of elements can be associated with end-user applications. For example, display specifiers are responsible for what you as an administrator see when you use ADSI Edit; they are also responsible for what a user sees in the product user interface. What you see and what the user sees are different, even though what is seen in both cases references the same objects. The display specification system stores information for property sheets, context menus, icons, creation wizards, and localized class and attribute names.

The DisplaySpecifiers container stores other containers that correspond to each locale that is supported by Windows 2000. A *locale* is either a language or a language in combination with a country/region. Windows 2000 supports more than 150 locales, such as French (Belgium), Arabic (Saudi Arabia), and so forth. The names of locale containers are the hexadecimal representations of the locale identifiers (LCIDs). For example the English (United States) locale container is 409.

Display specifier objects (class *displaySpecifier*) are named by appending the LDAP Display Name of the class object with the string "-Display." For example, the *user* class has a corresponding display specifier object called "user-Display." Thus, when an Active Directory administrative tool displays an object of a particular class, the object is displayed according to information contained in the display specifier object whose name contains the same name as the respective class within the container for the current locale.

Because Active Directory allows the schema to be modified by creating new classes and attributes or modifying existing classes, display specifier objects can be modified to reflect any new user interface elements that schema modifications require. For more information about display specifiers, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. Follow the links to the "Windows 2000 Active Directory Programmer's Guide."

Extended-Rights Contains the set of all extended rights for the forest, stored as *controlAccessRight* objects. Access control on custom actions or operations are called *control access rights*, or extended rights. Access control determines who is permitted to perform operations on Active Directory objects. Access to standard actions or operations is controlled by two major types of permissions: container operations and attribute-based access. Other operations can have semantics that are not tied to specific attributes, and these operations might also require access control. For example, the *user* class can be granted a **Send As** right that can be used by Exchange Server, Outlook, or any other e-mail program, to determine whether a particular user can have another user send e-mail messages on their behalf. To add an extended right to Active Directory, you create a *controlAccessRight* object in the Extended-Rights container. For more information about extended rights, see "Access Control" in this book. For more information about creating extended rights objects, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. Follow the links to the "Windows 2000 Active Directory Programmer's Guide."

LostAndFoundConfig Provides storage for global configuration objects that are being created in containers that are simultaneously being deleted elsewhere on the network. If an object has been created in or moved to a location that no longer exists after replication, the "lost" object is added to the LostAndFoundConfig container. A LostAndFound container in each domain directory partition serves the same purpose for domain-specific objects.

Partitions Stores the cross-references to every directory partition in the forest, including the configuration partition, the schema partitions, and all domain directory partitions. These cross-references to directory partitions make referrals to other domains possible during LDAP searches. Domain directory partitions can be viewed and managed in Active Directory Domains and Trusts.

Physical Locations [Is not implemented in Windows 2000, but is reserved for future use.]

Sites Identifies all of the sites in the enterprise network, the domain controllers in those sites, and the replication topology. The contents take the form of transports between sites, subnets, and the first site created, which is called Default-First-Site-Name. The contents of the Sites container can be viewed and managed in Active Directory Sites and Services. For more information about Sites container objects, see "Active Directory Replication" in this book.

Services Stores network-wide, service-specific information that applications use to connect to instances of services in the forest, irrespective of the computer on which the service runs. This service information includes system volumes, network services, and routing and remote access services. The contents of the Services container can be viewed and managed in Active Directory Sites and Services. For more information about the Services container objects, see "Service Publication in Active Directory" in this book, and see Windows 2000 Server Help.

Well-Known Security Principals Contains the special identities that are defined by the Windows 2000 security system, such as Everyone, Local System, Principal Self, Authenticated User, and Creator Owner.

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

- The information is truly of 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 that is stored in another domain is not sufficient.
- The volatility of the information is very low.
- The volume of information is very small.

Note 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

Three administrative tools in Windows 2000 enable you to manage different portions of the Configuration container. The following tools are available on the **Start** menu; point to **Programs** and then to **Administrative Tools**, which is the default menu.

- Active Directory Sites and Services.

  You can manage the objects in the cn=sites,cn=configuration,dc=*ForestRootDomain* container and the cn=services,cn=configuration,dc=*ForestRootDomain* container.

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

- Active Directory Domains and Trusts.

  You can manage the trust relationships between the domain directory partitions represented in the cn=partitions,cn=configuration,dc=*forestRootDomain* container. For more information about managing trust relationships, see "Active Directory Logical Structure" and "Authentication" in this book.

- Active Directory Schema.

  You can manage *classSchema* and *attributeSchema* objects stored in the Schema container (cn=schema cn=configuration,dc=*forestRootDomain*). Active Directory Schema is available as an MMC snap-in that you can install from the MMC **Console** menu. However, there are special requirements for installing this tool that do not apply to the installation of other MMC snap-ins. For more information about how to install the Active Directory Schema MMC snap-in and about managing the schema, see "Active Directory Schema" in this book.

For more information about managing configuration data, see "Active Directory Replication" and "Active Directory Diagnostics, Troubleshooting, and Recovery" in this book.

Schema Directory Partition

The schema for Active Directory consists of a set of object classes, attributes, and syntaxes. The schema also defines rules that ensure that objects are created and modified with consistency. Active Directory contains a default set of classes and attributes that cannot be modified. However, if you have the credentials to do so and if schema modification is enabled for the domain controller, you can extend the schema by adding new attributes and classes to represent application-specific classes. These changes must be targeted at the domain controller that holds the schema master role for the forest.

For more information about enabling schema modification and extending the schema, see "Active Directory Schema" in this book. For more information about single-master roles, see "Managing Flexible Single-Master Operations" in this book, and see Windows 2000 Server Help. For more information about default classes and the schema, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

If all of 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. This snap-in is not available by default and must be installed separately. 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.

For more information about installing the Active Directory Schema snap-in, see "Active Directory Schema" in this book. For more information about managing schema objects, see Windows 2000 Server Help.

Domain Directory Partitions

When you create a new domain, a domain directory partition is created in Active Directory as an instance of the class *domainDns* and is added to the list of domain partitions in the Partitions container.

Viewing the Contents of a Domain Directory Partition

The topmost 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:

Note Some containers are visible in the Active Directory Users and Computers console only in the Advanced Features view. To view all of the containers in Active Directory Users and Computers, on the **View** menu, click **Advanced Features**. In addition to displaying more containers, when viewing an object's properties in the **Advanced Features** view, object details and security property pages are also displayed. The **Object** tab displays class, creation, and modification information about the object The **Security** tab can be used to set permissions on an object so that unauthorized users do not have access to protected information.

Users Default storage area for new user accounts that are created through legacy APIs that are not Active Directory–aware. When a Windows NT 4.0 domain or a Windows NT 3.51 domain is upgraded to Windows 2000, the user accounts and groups are moved automatically to the Users container. The Users container also supports the Windows NT 4.0 tool User Manager (Usrmgr). This container cannot be renamed.

Note In client applications, the Users container and other special containers (such as Computers, System, Domain Controllers, Infrastructure, Deleted Objects, and LostAndFound) can be dependably located by using well-known GUID containers. For more information about using ADSI to locate special containers, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

System (Advanced Features) Built-in system settings for the various system service containers and objects. (For more information about the System container, see "Contents of the System Container" later in this chapter.)

LostAndFound (Advanced Features) Storage area for new domain objects whose containers were deleted elsewhere at the same time that the object was created. If an object has been created in or moved to a location that is missing after replication, the "lost" object is added to the LostAndFound container. The LostAndFoundConfig container in the configuration directory partition serves the same purpose for forest-wide objects.

Builtin Objects that represent the default built-in groups (for example, Builtin, Administrators).

Deleted Objects A special container, not visible in the UI, to which objects are moved when they are deleted. The deleted objects are stored as tombstones, which are eventually removed by garbage collection. The contents of the Deleted Objects container are visible if you search by using the 1.2.840.113556.1.4.417 control, which enables you to see deleted objects. (For more information about viewing deleted objects and about LDAP searches, see "Name Resolution in Active Directory" in this book.)

Domain Controllers Default container for new Windows 2000 domain controllers. The Domain Controllers container cannot be renamed.

Infrastructure Holds information used by the domain controller that holds the infrastructure master role, which keeps interdomain object references up to date. The Infrastructure master creates deleted child objects in the Infrastructure container. When these deleted objects replicate, they have the effect of removing outdated phantom records. (For more information about phantom records, see "Phantom Records" later in this chapter.)

Computers Default storage area for "new" computer objects that were originally created through legacy APIs that are not Active Directory–aware. When a Windows NT 4.0 domain or a Windows NT 3.51 domain is upgraded to Windows 2000, the computer accounts are moved automatically to the Computers container.

ForeignSecurityPrincipals Proxy objects for security principals from Windows NT 4.0 domains or Windows NT 3.51 domains or from different forests that have been added to Windows 2000 groups.

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 that are 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.

Contents of the System Container

The System container stores per-domain operational information, which includes the default 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.* Administrator security descriptor holder. Windows 2000 implements protection of administrative groups by a background task that computes the set of memberships and checks whether their security descriptors are well-known protected security descriptors. This task is executed only on the domain controller that has the primary domain controller emulator (PDC emulator) role. (For more information about security descriptors, see "Access Control" in this book. For more information about the PDC emulator role, see "Managing Flexible Single-Master Operations" in this book.)

- *Default Domain Policy.* Lists the security groups and default permissions for the domain. It stores policies for passwords, lockouts, Kerberos, Encrypting File System (EFS) data recovery, and trusted root certificates. It also holds the Application Categories container. Each application that is deployed has one or more associated categories that can be used to organize the applications in an organization. Categories appear when you add or change programs in Add/Remove Programs in Control Panel. A drop-down list displays all the categories. The object class of Application Category is *classStore*. The *classStore* is where COM components and applications are published. The Application Deployment wizard writes to the *classStore.* (For more information about the default domain policy, see "Group Policy" in this book.)
- *Dfs Configuration.* Lists the Fault Tolerant Distributed file system (Dfs) configuration and Dfs volume information. (For more information about Dfs, see "Distributed File System" in this book.)
- *File Replication Service.* Lists the Domain System Volume (SYSVOL share) and provides replication schedule from Sunday through Saturday 12:00 a.m. to 12:00 a.m. (For more information about the File Replication Service, see "File Replication Service" in this book.)
- *FileLinks.* Used by the Distributed Link Tracking Server service (TrkSvr) to store information about linked files that have moved across NTFS volumes. Includes the ObjectMoveTable, which tracks moved files, and the VolumeTable, which maps volume IDs to computer IDs. (For more information about distributed link tracking, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources
- *IP Security.* Contains the IP Security policies that are applied to local computers, domain member servers, domains, organizational units, or any Group Policy object in Active Directory. Depending upon your organization's guidelines, IP Security policies can store multiple security actions, called rules, so that one policy can be applied to multiple computers. These security specifications apply to all users who log on to the computer. (For more information about IP Security policies, see "Internet Protocol Security" in the *TCP/IP Core Networking Guide.*
- *Meetings.* Microsoft® NetMeeting® uses the "Meetings" folder to publish network meeting objects.
- *MicrosoftDNS.* Active Directory-integrated zone database records are created in this container, and the contents are replicated to all domain controllers in the domain. When DNS data is stored in Active Directory, each DNS zone is an Active Directory container object (*dnsZone*). The *dnsZone* object contains a *dnsNode* object for every unique name within that zone. The *dnsNode* object has a *dnsRecord* multivalue attributed that contains a value for every resource record associated with an object's name. (For more information about Active Directory-Integrated zones, see "Windows 2000 DNS" in the *TCP/IP Core Networking Guide.*

- *Policies.* Contains Group Policy objects, which specify user and computer configurations for groups of users and computers. This container is identified by GUID and includes version information that is used to ensure that information is synchronized with Group Policy template information; status information that indicates whether the Group Policy object is enabled or disabled; and a list of components, or extensions, that have settings in the Group Policy object. (For more information about Group Policy, see "Group Policy" in this book.)

  Note In addition to the Policies container, Group Policy objects are also stored in a Group Policy template and are identified by a GUID. The Group Policy template is located in the system volume and is used to store file type data for the Group Policy object.

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

- *RpcServices.* Includes the Remote Procedure Call (RPC) name service lookup for domains by using versions of Windows earlier than Windows 2000.
- *WinsockServices.* Windows Sockets services that publish themselves using the registration and resolution (RnR) APIs are published in this container.

For more information about services that are published in the System container, see "Active Directory Service Publication" in this book.

Directory Data Store

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 This file stores the database that is in use on the domain controller. It contains the values for the domain and a replica of the values for the forest (the Configuration container data).

%SystemRoot%\System32\Ntds.dit This file is the distribution copy of the default directory that is used when you promote a Windows 2000–based computer to a domain controller. The availability of this file allows you to run the Active Directory Installation Wizard (Dcpromo.exe) without your having to use the Windows 2000 Server operating system CD. During the promotion process, Ntds.dit is copied from the %SystemRoot%\System32 directory into the %SystemRoot%\NTDS directory. Active Directory is then started from this new copy of the file, and replication updates the file from other domain controllers.

Linked Attributes

Some interobject references in the directory require back-references for either usability or administrative purposes. For example, if *managedBy* is an object attribute, you can look at ObjectA and determine that ObjectA is managed by ObjectB. Likewise, it is sometimes helpful to be able to look at ObjectB and determine what objects ObjectB manages (the values of the *managedObjects* attribute). Active Directory maintains referential integrity between objects that reference each other so that when one object is moved in the directory tree, the reference between it and other objects is maintained. This referencing is accomplished through linked attributes.

Two attributes that are linked are marked in the schema as having the same link-pair identifier—one is marked as the forward link and the other as the back link. For reasons that relate to security and replication, only the forward link attribute can be modified. For example, in the *managedBy/managedObjects* link pair, *managedBy* is the forward link. Therefore, to adjust the *managedObjects* attribute on a user object, you must go to the objects that you want to add or remove from the user's *managedObjects* value and modify the *managedBy* value on each object. Back-link attributes are computed when they are requested by a user action.

Note When you extend the schema, you have to know when to make an object a link object. For more information about extending the schema, see "Active Directory Schema" in this book.

To find all of the objects that ObjectB manages, links are examined for all records in which the link pair is *managedBy/managedObjects* and the back-link attribute identifies ObjectB. The link pairs of those records provide the database identifiers of all the records (objects) that are managed by ObjectB.

The *managedBy* and *managedObjects* example uses a single-value forward link and a multivalue back link, respectively, but there is no requirement that the forward link be a single-value link. For example, distribution list membership is implemented both as a forward-link and as a back-link pair. The back-link objects would be the objects that store the *isMemberOfDl* attribute. The forward-link *member* attribute is a multivalue attribute, which allows a user to be a member of more than one distribution list. The back link must always be a multivalue link because it is impossible to restrict who creates links to various objects.

Table 2.8 shows link values for an object (ObjectB) that is the manager of several other objects (ObjectA, ObjectC, and ObjectD). The distribution list (DL1) is an example of an object that has several objects as members.

Table 2.8 Example of Forward-Link and Back-Link Values

Linked object	Back-linked object	Link pair
ObjectA	ObjectB	*managedBy/managedObjects*
ObjectC	ObjectB	*managedBy/managedObjects*
ObjectD	ObjectB	*managedBy/managedObjects*
DL1	ObjectE	*member/isMemberOfDl*
DL1	ObjectF	*member/isMemberOfDl*
DL1	ObjectG	*member/isMemberOfDl*

When an object that is linked is deleted, all of its linked attribute values are deleted. In the preceding example, if ObjectA were deleted, the *managedObjects* multivalue attribute on ObjectB would suddenly (and with no change to any replication-related metadata) lose a value. Similarly, if ObjectB were deleted, the value of the *managedBy* attribute on ObjectA would suddenly be blank. Nothing about the object changes in either case, except that the attribute value is gone.

Searching on Back Links

When you request the value of a back link on a particular object (for example, "What objects are managed by ObjectB?"), the system searches for all objects whose corresponding forward link names the original object (that is, "What objects have ObjectB as the value in their *managedBy* attribute?"). The results of that search and, hence, the apparent contents of the back-link attribute, depend on the LDAP port to which the client is bound; that is, the results can differ, depending on whether the client binds to the local domain (LDAP port 389) or the Global Catalog (LDAP port 3268).

For example, suppose that you are looking at the user object named "JohnDoe." You are interested in discovering the groups in which JohnDoe has memberships. Suppose further that JohnDoe is an object in the child domain B that has a parent domain A. If you bind to the JohnDoe object in domain B and read the *memberOf* attribute, you receive a list of all group memberships in domain B, including both domain local and global groups; however, you do not see any memberships in groups outside domain B. On the other hand, if you bind to the copy of the JohnDoe object in the Global Catalog and read the *memberOf* attribute, you see the group memberships in all universal groups in the forest. You do not see any domain local group memberships, however, because local groups are not replicated to the Global Catalog. Thus, to see all of an object's memberships, you must search both the local and Global Catalog copies of the object.

For example, suppose you are interested in learning what the groups are to which JohnDoe has memberships. The system implicitly searches for all objects whose forward links name the object (that is, the group objects that have JohnDoe as a value for the *member* attribute). Suppose further that JohnDoe is an object in the child domain B that has a parent domain A. When there is more than one domain in a forest, you must take into account the following group behaviors:

- By definition, global groups cannot contain members from other domains. Therefore, in our example, only global groups in one domain (the domain to which JohnDoe belongs) can possibly have JohnDoe as a member.
- Domain local groups can contain members from other domains; however, although objects in these groups are replicated to the Global Catalog, their *member* attribute is not.

In the example, if you bind to the JohnDoe object in domain B and read the *memberOf* attribute, Active Directory lists all groups in domain B that have JohnDoe as a member, including both local and global groups; however, no groups except for domain B (the domain to which JohnDoe belongs) are visible.

If you bind to the copy of the JohnDoe object in the Global Catalog and read the *memberOf* attribute, the groups that are listed depend on what domain contains the Global Catalog server, assuming that there is not a Global Catalog server in both domains.

- If the Global Catalog server is in domain A, Active Directory lists the group memberships in all global groups in the forest. However, Active Directory does not list domain local groups in domain B because although domain local group objects are replicated to domain controllers, their *member* attributes are not. Thus, to see all of an object's memberships, you must search both the local domain (domain B) and Global Catalog copies (in domain A) of the object, unless the domain controller is a Global Catalog server.
- If the Global Catalog server is also a domain controller in domain B, Active Directory lists both the global groups and the domain local groups of which JohnDoe is an immediate member. When there is a Global Catalog server in the local domain, this local Global Catalog server is the best server to search.

Note Memberships in domains that are external to the forest are not found in either type of search because they are outside the scope of the forest. These memberships must be discovered by using the respective external cross-reference. (For more information about external cross-references, see "Name Resolution in Active Directory" in this book.)

Group Members from External Domains

If you add a member of a trusted domain from a different forest to a group in your domain, Samsrv.dll creates a placeholder object of the class *foreignSecurityPrincipal*. This object represents the real object, about which Active Directory has no information because the object exists in a different forest. When you list the members of a group, Active Directory usually lists the distinguished names of the group members. For a member that is from an external domain, Active Directory displays the distinguished name of the foreign security principal object in the form of a NetBIOS name. For example, the user JohnD from the domain Acquired.com would appear as JohnD in "acquired" as shown in Figure 2.7.

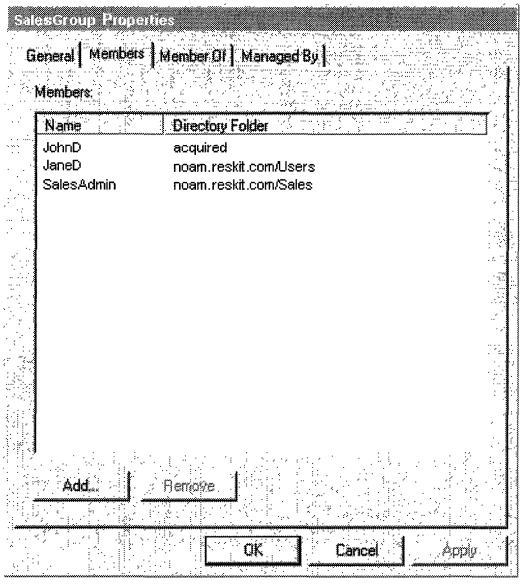


Figure 2.7 Example of a Members Tab That Displays the Distinguished Name of a Foreign Security Principal

If you open the properties on the foreign group member, an informational message like the one in Figure 2.8 appears. This message explains that the member is not a real object in Active Directory but a placeholder for the object. The object SID is displayed in the title bar of the dialog box.

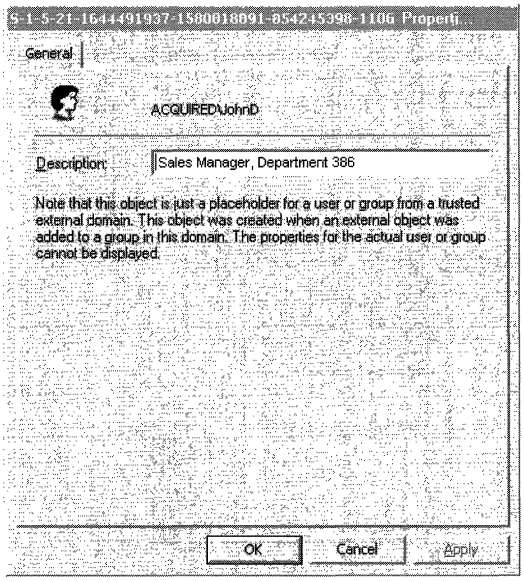


Figure 2.8 Properties for a Member from an External Domain

You can use the object's SID in an LDAP query to determine the LDAP name of the object. Such a query involves enumerating all trusted domains and then issuing a query on each one for the object whose *objectSid* attribute value matches the SID of the foreign security principal object.

Phantom Records

In Active Directory, all references from one object to another stored as the database identifier of the referenced object. For example, a user object might have an attribute that defines that user's manager; the value for that attribute is the database identifier of the user object that represents the manager in the database. If the referenced object does not exist (for example, a user account in one domain has a manager in a different domain, and the contacted server is not a Global Catalog), a "phantom" is created as a record in the database, and the database identifier of that record is used. A phantom record contains the GUID, the SID (in the case of references to security principals), and the distinguished name of the object that is being referenced. If a copy of the object named in the attribute exists in the local database, no phantom is needed. If the object is located in an external directory partition, the local database uses a phantom record. For example, if an object in the domain dc=noam,dc=reskit,dc=com holds a reference to an object in dc=europe,dc=reskit,dc=com, a phantom for that object and its parent exist in the domain dc=noam,dc=reskit,dc=com. The *infrastructure master* deletes phantom objects when the objects that they reference are renamed or deleted. For more information about the infrastructure master, see "Managing Flexible Single-Master Operations" in this book, and see Windows 2000 Server Help.

Database Write Operations

Operations are written to the Active Directory database as transactions, which are the units of work performed by a database. Transactions are *atomic*—that is, they are either completed in full or are not applied at all. If for any reason an error occurs and a transaction is unable to complete all of its steps, the system is returned to the state that existed before the transaction began. An example of an atomic transaction is an account transfer transaction. Money is removed from account A and placed into account B. If the system fails after it removes the money from account A, the transaction processing system puts the money back into account A and returns the system to its original state—that is, it rolls back the transaction.

In Active Directory, write operations on a single object are transacted—that is, a transaction cannot be applied across multiple objects. Active Directory writes a transaction synchronously to the transaction log file and then to the database. First, a change is made to an in-memory copy of the object. Then the change is written to the log file, which ensure that the change is effected, even if the database shuts down after that point. The database engine continually updates the database file with recent changes. The database update works from memory, not from the log files, so it keeps pace with the updates rather than waiting for the server to be available. This method of performing updates is referred to as "advancing the checkpoint," where the *checkpoint* is the point in time at which all changes that have been made thus far have been fully written to the database.

Log-based Recovery

The Active Directory logging and recovery system is designed to guarantee data integrity and consistency in the case of a system crash. Logging is the process of recording database operations in a log file. *Recovery* is the process of using the log file to restore a database after a system crash to the most recent state that is recorded in the log file.

Note Because Active Directory is replicated (if you have at least two domain controllers in a domain), you can recover from a disaster by restoring from backup and allowing replication to replicate data that has changed since the last backup.

For efficient disk usage, Active Directory uses circular logging. Circular logging keeps the log file size to a minimum by overwriting data that is no longer needed as rapidly as possible. By using circular logging, the directory database engine automatically deletes unneeded log files every time the checkpoint is advanced.

For more information about backing up and restoring Active Directory, see "Active Directory Backup and Restore" in this book. For more information about allocating log file space, see "Active Directory Diagnostics, Troubleshooting, and Recovery" in this book. For more information about replication of database transactions, see "Active Directory Replication" in this book.

Attribute Indexing

For efficient searches on common attributes, Active Directory supports indexing. Attributes can be indexed to decrease the time required to locate a record in a large database—that is, a certain attribute or combination of attributes can be used to uniquely identify a record.

By default, attributes that are searched often, such as *surname*, *cn* (common name), *userPrincipalName*, and so forth, are indexed. You can select other attributes for indexing by using the Active Directory Schema console. When you open the properties for an attribute object, you can see whether the attribute is already selected for indexing; if it is not, you can select it, which sets an index flag on the attribute. The value of this flag is replicated, and the indexing is performed by the DSA when the schema is refreshed. Likewise, if you reverse the selection, the change is made when the schema is refreshed.

Note Indexing attributes can affect update rate and database size. Attributes should be indexed only when you are certain that they will be used often for searching.

For more information about searching on attributes, see "Active Directory Name Resolution" in this book.

Object-Based Security

A fundamental relationship exists between Active Directory and the security services that are included with Windows 2000. Active Directory stores domain security policy information that has direct bearing on the use of the system, such as domain-wide password restrictions and system access privileges. In addition, Windows 2000 implements an object-based security model and access control for all objects in Active Directory. Each object in Active Directory is associated with a unique security descriptor that defines the access permissions that are required to read or update the object properties. Permissions can be assigned at the property level.

Security Identifiers

Each security principal (user, group, and computer, as well as the domain itself) has a SID, which is the property (*objectSid*) that authoritatively identifies the object to the security system. The SID of a user, group, or computer is derived from the SID of the domain to which the object belongs; this SID is the same as the SID of the domain except that it has one extra 32-bit component called the relative identifier.

Security Descriptors

In Windows 2000, a *security descriptor* is associated with each object. The security descriptor defines the access control information that is associated with the object. Security descriptors include the following:

- The *header*, which contains control flags and pointers to the four parts of the security descriptor.
- The *owner*, which is a SID that indicates what user or group owns the object. The owner of an object has some inherent rights to the object.
- The *primary group*, which is a SID that is present for POSIX compliance.
- The *discretionary access control list (DACL)*, which contains a list of access control entries that define who does and does not have specific types of access to the object. A DACL is controlled by the owner of the object. The owner can grant this control to others.
- The *system access control list (SACL)*, which contains a list of system access control entries that specify audit and alarm message generation when particular subjects attempt specific types of access to the object. SACL is controlled by security administrators (who are, by default, members of the Administrators group).

Default Object Security

When an object is created in Active Directory, its security descriptor can be specified manually by the object creator. If no security descriptor is specified, a default security descriptor is applied to the object. The default security descriptor is computed according to the following rules:

- The *Owner* usually is assigned by default. If the creator's access token contains a default owner, the default is set as the "owner." Otherwise, the user SID of the object's creator is assigned. The only case in which the owner is not the user SID of the creator of the object is when the creator is a member of the Domain Administrators group. In that case, the default owner is set to the Domain Administrators group SID. Thus, all objects that are created by an administrator are owned by all members of the Domain Administrators group.
- The *Primary group* is usually assigned by default. If the creator's access token contains a default primary group, this default group is set as the primary group. Otherwise, a NULL SID is assigned.

- *DACL* is assigned as a new object's DACL if an explicit (not inherited) DACL is provided. The parent container's DACL is checked for any inheritable access control entries. If one is found, the inherited DACL is merged with the explicit DACL as the new object's DACL. If an explicit DACL is not provided, the default DACL that is provided by the Active Directory schema, if any, is used. If no default DACL is provided by Active Directory, the creator's access token is checked for a default DACL. If a default DACL is found in the owner's access token, it is assigned as the new object's DACL. If there is no default DACL in the creator's access token, no DACL is assigned to the new object. In this case, unconditional access is granted to everyone.
- *SACL* is assigned as being any explicit SACL that is provided. The parent container's SACL is checked for any inheritable access control entries. If inheritable access control entries are found, the inherited SACL is merged with the explicit SACL as the new object's SACL. If an explicit SACL is not available, the default SACL that is provided by the Active Directory schema, if any, is used. If no default SACL is provided by Active Directory, no SACL is assigned.

Note An object manager is different from an object's owner. Each object type has an object manager that handles creation of the object. Active Directory is the object manager for directory objects. NTFS is the object manager for file system objects.

For more information about how access control, access tokens, security principals, and security descriptors are used by the security subsystem, see "Access Control" in this book.

Installing Active Directory

Windows 2000 Server can be installed as a new operating system or as an upgrade to an existing Windows NT Server operating system. In either case, adding Active Directory to a server is a separate operation from installing Windows 2000 Server. You first install Windows 2000 Server; then you install Active Directory on the servers you want to use as domain controllers.

When you install Active Directory, certain conditions are required for the process to succeed. Because most conditions are addressed during Windows 2000 Server installation and because debugging of the required network and system configuration is most effectively achieved prior to installing Active Directory, the process of installing Active Directory is implemented as a separate program (Dcpromo.exe) subsequent to installation of the operating system.

After you install Windows 2000 Server, you can use one of the following options to install Active Directory:

- Use the Active Directory Installation Wizard. The wizard can be started as follows:

  In the **Windows 2000 Configure Your Server** dialog box that appears when you start the server computer, you can select the **Active Directory** option for installing Active Directory. *This method is highly recommended.* **Windows 2000 Configure Your Server** provides guidance to ensure proper installation of Active Directory.

  –Or–

  On the **Start** menu, click **Run**. Then type **dcpromo** in the **Run** dialog box.
- Use a text file to perform an unattended ("silent") installation of Active Directory. The Active Directory Installation Wizard can be started from a command prompt and an answer file specified by typing **dcpromo /answer:*<answerfile name>*** to install Active Directory by using programmed responses to the wizard that you prepare in a special text file.

Note When Windows NT 4.0–based domain controllers are upgraded to Windows 2000, the Active Directory Installation Wizard automatically opens after the installation of the operating system is completed. The role of the server does not change to the role of a domain controller until the wizard has been run.

The Active Directory Installation Wizard also is used to remove Active Directory from a server.

For more information about using unattended installation, see "Unattended Setup for Installation or Removal of Active Directory" later in this chapter. For more information about using the **Configure Your Server** wizard, see Windows 2000 Server Help. For more information about removing Active Directory, see "Removing Active Directory" later in this chapter.

Active Directory Configurations

When you are installing Active Directory on a stand-alone server (a member of a workgroup) or member server (a member of a Windows 2000 domain) that is running Windows 2000 Server, you can create one of the following configurations:

- The first domain in a new tree in a new forest.
- The first domain in a new tree in an existing forest.
- A child domain in an existing tree.
- An additional domain controller in an existing domain.

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.

Note During the installation of Active Directory, you can stop the replication process and allow noncritical replication to finish later. To stop the replication process, click the **Finish Replication Later** button when it appears. Replication then continues after the computer is restarted as part of the normal replication activity of the domain controller. The domain controller does not advertise itself until replication is complete.

Installation of Active Directory occurs through different pathways, depending on the nature of the Windows 2000 Server installation. The following are the scenarios for installing Windows 2000 Server:

- When you install Windows 2000 Server as the first operating system (fresh install) or replace or upgrade an existing operating system on a computer that is not a domain controller, you create either a Windows 2000 stand-alone server or member server, and then install Active Directory as one of the following:
  - A new domain
  - An additional domain controller in an existing domain

- When you upgrade an existing Windows NT 4.0 backup domain controller, the Active Directory Installation Wizard starts automatically after the upgrade setup is completed. In this case, there are two choices; do either of the following:
  - Add Active Directory, which creates an additional domain controller in an existing domain.
  - Convert the backup domain controller to a member server joined to an existing domain.
- When you upgrade an existing Windows NT 4.0–based primary domain controller, you have to install Active Directory as a new domain.

Figure 2.9 shows the various pathways from Windows 2000 Server setup to Active Directory installation.

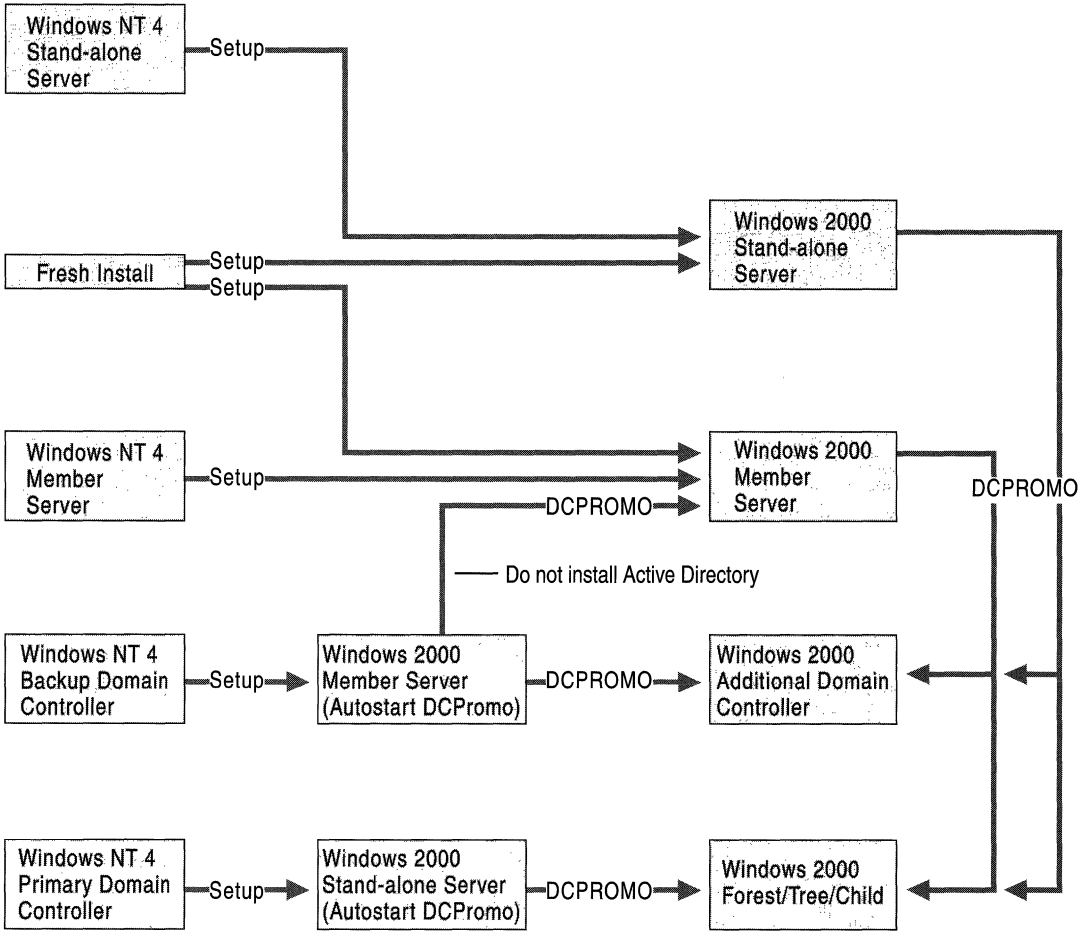


Figure 2.9 Windows 2000 Server Setup and Active Directory Installation

For more information about trees and forests, see "Active Directory Logical Structure" in this book. For more information about the Ntds.dit file, see "Directory Data Store" earlier in this chapter. For more information about directory partitions, see "Directory Partitions" earlier in this chapter.

Installation Prerequisites and Verifications

The Active Directory Installation Wizard confirms several configuration and security parameters before it proceeds with Active Directory installation. Different checks are required for different conditions, depending on the type of domain controller that is being installed. The purpose of this verification routine is to validate the parameters that you specified with respect to the directory service that this server is about to join.

Note The Active Directory Installation Wizard requires 200 megabytes (MB) of disk space for the Active Directory database and 50 MB for the ESENT transaction log files. File size requirements for the Active Directory database and log files depend on the number and type of objects in the domain database (or databases held by the forest, if the computer is serving as a Global Catalog server).

When you start the Active Directory Installation Wizard, the following checks are performed before the wizard user interface actually appears:

- The current user is logged on to the local computer as a member of the Administrators group on the local computer.
- The computer is running Windows 2000 Server.
- A previous installation or removal of Active Directory has not taken place without the computer having first been restarted.
- The Active Directory Installation Wizard is not currently running.
- An installation or removal operation of Active Directory is not already in progress.
- At least one logical disk drive is formatted with the NTFS v5 file system.

For more information about formatting a logical disk drive to NTFS v5, see "Running Setup" and "File Systems" in the *Microsoft® Windows® 2000 Professional Resource Kit*.

Verify Unique Names

The following name checks are performed for the installation of domain controllers in an existing domain and new domains in an existing forest:

- For both the installation of an additional domain controller in the domain and the installation of a new domain in an existing forest, the server name must not exist as a relative distinguished name of an NTDS Settings object in the site to which the domain controller is being added. If the wizard finds an NTDS Settings object with the same relative distinguished name as the server name, the wizard deletes the existing object and assumes that you are requesting a reinstallation.
- For the installation of a new domain in an existing forest, the NetBIOS name of the domain also must not exist as a relative distinguished name of a cross-reference object in the cn=partitions,cn=configuration,dc=*forestRootDomain* container.

Note For the installation of a new forest, the preceding checks are not performed because there is no source domain controller.

For more information about NTDS Settings objects, see "Active Directory Replication" in this book. For more information about cross-reference objects, see "Name Resolution in Active Directory" in this book.

Verify That TCP/IP Is Installed

In all cases of domain controller installation, the wizard checks the network configuration for the presence of TCP/IP. If TCP/IP is not installed or a DHCP-provided address is not available, you must install and configure TCP/IP and provide a subnet and default gateway IP address.

Note Providing a static IP address is not a requirement for the installation of a domain controller.

Verify That DNS Client Is Configured

The wizard checks to see whether the DNS client is configured on the server. DNS client configuration is part of TCP/IP configuration and consists of providing the IP address of one or more DNS servers on the network. Because DNS is used to locate domain controllers, the DNS database should contain a resource record that can be used to locate a domain controller in each domain. When Active Directory is installed on a server that is to become the first domain controller in a new domain, the wizard attempts to find a DNS server that supports dynamic updates and that is authoritative for the name of the domain in which the server is to be a domain controller. The installation computer cannot find such a DNS server if it does not exist (for example, when you are installing a new forest) or if the DNS client on the server is not configured with the valid IP address of at least one DNS server that should be used for DNS name resolution. In either case, you are prompted to choose whether to allow the wizard to install and configure the DNS server locally during the installation process or to install and configure DNS manually after Active Directory is installed. In the case of an additional domain controller in an existing domain, a DNS server that is running on the network is assumed and a search for an authoritative DNS server is not performed.

For more information about automatic DNS configuration and DNS requirements for Active Directory, see "Windows 2000 DNS" in the *TCP/IP Core Networking Guide*.

Get and Validate the DNS Domain Name

During the creation of a new domain, you must provide a name for the domain and specify the parent domain, if any. The wizard must determine that the parent domain exists and that the name for the new domain is unique in the forest. If the name that you provide for the domain is not valid, the Active Directory Installation Wizard requests a different name.

Get and Validate the NetBIOS Name

Unless you are creating a new domain as the result of a primary domain controller upgrade, a NetBIOS name is generated from the DNS domain name. In the case of the primary domain controller upgrade, the NetBIOS name is the previous NetBIOS name of the domain. The Active Directory Installation Wizard prompts you to either change or to accept the NetBIOS name that is derived from the domain name. The wizard then verifies that the NetBIOS name is unique on the network.

Enter Administrative Password

For an upgrade, the existing server accounts are migrated. For a fresh installation, all local accounts and passwords (except the accounts that belong to Power Users) are preserved. During an Active Directory installation, the user does not have the option of entering a new administrative password. During a removal of Active Directory, you have the option of entering an administrative password for the local Administrator account, which is created by the Active Directory Installation Wizard.

Get Credentials for the User

To create a new domain controller, the Active Directory Installation Wizard requires an administrative account and password. Unless you are creating a new forest, administrative credentials are checked by the Active Directory Installation Wizard.

- To create an additional domain controller in an existing domain, the credentials that are provided must be sufficient to join the computer to the domain that is being replicated and to create an NTDS Settings object in the Configuration container that is being replicated.
- To create a child domain, the credentials of a member of the Enterprise Admins group must be provided, or a child domain must have been previously created in Active Directory by a member of the Enterprise Admins group.
- To create a new tree-root domain, the credentials of a member of the Enterprise Admins group must be provided.

Note The wizard requests credentials in the form of a user name, password, and domain. The wizard accepts the user name only in the form of a logon user identifier (for example, johnSmith). A user principal name (UPN), which has the form johnSmith@*domainName*, is not accepted.

If you are not logged on as a domain administrator, the wizard prompts you to provide the name and password for an appropriate administrative account.

Note When you install or remove a domain in the forest, the domain controller that holds the domain naming master role must be available and reachable by the RPC protocol.

For more information about the domain naming master, see "Managing Flexible Single-Master Operations" in this book, and see Windows 2000 Server Help.

Get and Verify File Paths

During Active Directory installation, the locations for the Active Directory database file, log file, and system volume (Sysvol) share are set. The system volume provides a default Active Directory location for files that categorically must be shared for common access throughout the domain. Creation of the system volume requires an NTFS v5 volume. If an NTFS v5 volume cannot be found, or if there is not sufficient free disk space to complete the installation, installation of the domain controller cannot proceed. The system volume objects are created later, after the computer has been restarted.

Note For reasons of security, it is recommended that you install Ntds.dit on an NTFS volume as well, although it is not required by the Active Directory Installation Wizard.

The wizard provides the opportunity to select the location for the Ntds.dit file, the Active Directory log files, and the Sysvol directory. For optimum domain controller performance, select separate physical hard disks for the Ntds.dit file and for the Active Directory log files.

Configure Site

The Active Directory Installation Wizard determines the site to which to add the new domain controller. It checks existing sites to determine whether the subnet of the computer that you are installing is found. If the subnet is not found in the site that contains the source domain controller, you can select an existing site to which to add the computer. If you want to create a new site for this domain controller, you can create the new site after Active Directory is installed. The domain controller then can be moved from the installation site to the new site.

Note When you are using an unattended installation, you can specify a site with the SiteName parameter in the answer file.

The site determination process is as follows:

- The Active Directory Installation Wizard attempts to use the Locator to find the site in which the computer is currently located. If the Locator does not return a site for the computer (that is, the computer's subnet is not associated with a site), the wizard assumes that the site for the new domain controller is the site of the source domain controller.
- When you are installing the first domain in a forest, the default site, Default-First-Site-Name, is used.

- When the wizard has determined the correct site, it then checks to be sure a site object exists in Active Directory for that site. It also checks to be sure that a server object exists for the additional domain controller.
  - If a server object does not exist, the wizard creates an object for the server.
  - If the server object does exist, the associated NTDS Settings object is deleted and then recreated for the new domain controller. (An NTDS Settings object is created for each domain controller in the forest. If the NTDS Settings object already exists, the wizard performs as if this domain controller is being reinstalled.)

Note The NTDS Settings object is always created on a remote server (the source domain controller) for an additional domain controller. For a new domain, the NTDS Settings object is created on the computer that has the domain naming master role for a new domain. During replication of the Configuration container, the NTDS Settings object is then replicated to all domain controllers. (For more information about the domain naming master role, see "Managing Flexible Single-Master Operations" in this book, and see Windows 2000 Server Help.

Directory Service Configuration

After it verifies all of the required components, the Active Directory Installation Wizard confirms the settings that you have made. When you accept the settings, the process of actually configuring the directory service begins. This process can be cancelled by clicking **Cancel**.

For all types of installation, the wizard performs the following operations:

- Sets the values registry entries.
- Sets up Active Directory performance counters.
- Configures the computer to automatically enroll for an X.509 domain controller certificate from the first certificate authority that processes the computer. This certificate is required for SMTP-based replication. (For more information about certificates and certification authorities, see "Authentication" and "Windows 2000 Certificate Services and Public Key Infrastructure" in this book.)
- Starts the Kerberos v5 authentication service. (For more information about Kerberos authentication, see "Authentication" in this book.)
- Sets the LSA policy to be a domain controller when the computer is restarted. When you upgrade a primary domain controller, the wizard sets up domain security principals, local security principals, and LSA membership of the computer.
- Installs shortcuts to the Active Directory administration tools.

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 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 that is 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 of 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.

Setting Services to Start Automatically

During the installation of Active Directory, the following services are configured to start automatically:

- RPCLocator, which allows distributed applications to use the Microsoft RPC name service. The RPC Locator manages the RPC name service database. (For more information about the RPC Locator, see "Service Publication in Active Directory" in this book.)
- The Net Logon service, which runs the domain controller Locator algorithm. Net Logon also is responsible for creating a secure channel between clients and domain controllers during the logon process, registering service (SRV) resource records in DNS, and supporting the Windows NT 4.0 replication protocol (LMRepl).

- The KDC service, which runs on a physically secure server and maintains a database with account information for all security principals in its realm — the Kerberos v5 authentication protocol equivalent of a Windows 2000 domain.
- IsmServ (Intersite Messaging [ISM]service), which is used for mail-based replication between sites. Active Directory includes support for replication between sites by using SMTP over IP transport. SMTP support is provided by the SMTP service, which is a component of IIS. The set of transports that are used for communication between sites must be extensible; therefore, each transport is defined in a separate add-in DLL. These add-in DLLs are loaded into the ISM service, which runs on all domain controllers that are candidates for performing communication between sites. The ISM service directs send requests and receive requests to the appropriate transport add-in DLLs, which then route the messages to the ISM service on the destination computer.
- TrkSvr (Distributed Link Tracking Server service), which runs on each domain controller in a domain. This service enables client applications to track linked documents that have been moved to a location in another NTFS v5 volume in the same domain, in another domain, or in a workgroup. The Distributed Link Tracking Server service helps resolve shortcuts and OLE links to NTFS-resident files that have undergone a name change, a path change, or both.
- W32time (distributed time service), which synchronizes clocks between clients and servers that run Windows 2000. Time synchronization is automatic.

For more information about Net Logon and the domain controller locator, see "Name Resolution in Active Directory" in this book. For more information about the KDC and the Kerberos v5 authentication protocol, see "Authentication" in this book.

Setting Security

During the installation of Active Directory, security is enabled on directory service and file replication directories for access control, and actions allowed on domain objects are set through Group Policy.

Access Control

Default access control lists are configured on file and directory objects. Access control lists are also configured for the following registry keys and file system objects, including all child objects:

- HKEY_LOCAL_MACHINE\SOFTWARE
- HKEY_LOCAL_MACHINE\SYSTEM
- HKEY_USERS\.DEFAULT
- PROGRAM FILES
- %WINDIR%

For more information about access control, see "Access Control" in this book.

Group Policy

Group Policy is replicated from only the first domain controller in a domain to all additional domain controllers. In the case of the first domain controller, default Group Policy is configured by using the following security templates in the %Windir%\Inf directory:

- DCFirst.inf is used to define the default Password, Lockout, and Kerberos Group Policy settings for the default Group Policy object for the domain.
- DefltDC.inf is used to define the Audit and User Rights Group Policy settings for the default Group Policy object for the domain controller.
- DCUp.inf is used to define Windows 2000–specific settings during the upgrade of a Windows NT 4.0–based domain controller.

Note There is a default policy for domains, as well as a default policy for domain controllers. The domain controller policy has precedence over the domain policy. For example, if you want to grant the Add Workstation to Domain privilege to a user, you modify the default domain controller policy rather than the default domain policy.

For more information about domain and domain controller Group Policy settings, see "Group Policy" in this book.

Pre-Windows 2000 Security

For all types of installation, the Active Directory Installation Wizard provides the option of minimizing permissions to accommodate pre-Windows 2000 applications that require permissions that are less strict than those granted by Windows 2000–based domain controllers. If you have Windows NT 4.0–based Remote Access Service servers or Microsoft SQL Servers that are running on Windows NT 3.*x*–based or Windows NT 4.0–based computers, or if these applications are running on Windows 2000–based computers that are located in Windows NT 3.*x* domains or Windows NT 4.0 domains, the **Pre-Windows 2000 compatible permissions** option provides the permissions that these applications require for anonymous read access to particular user and group object attributes. **Pre-Windows 2000 compatible permissions**, which is the default setting, adds the Everyone group to the Pre-Windows 2000 Compatible Access local group. This group has access to the user and group object attributes that existed in Windows NT 4.0 and that are required by server applications to function with Active Directory.

Note The Everyone group contains every user account in the forest, including the Guest account and Anonymous/NullSession. Thus, the **Pre-Windows 2000 compatible permissions** option allows all users, including anonymous users, to have read access to domain user and group attributes.

Members of the Pre-Windows 2000 Compatible Access group have read access to the following attributes:

- All attributes on user objects that existed in Windows NT 4.0 (for example, SID, Name, logon hours, user account control).
- All attributes on group objects.

If all of your server-based applications are running on Windows 2000–based servers that are members of Windows 2000 domains, select the **Windows 2000-only permissions** option. This option prevents anonymous users from being able to read user and group information.

For more information about permissions, see "Access Control" in this book. For more information about remote access, see "Routing and Remote Access Service" in the *Microsoft® Windows® 2000 Server Resource Kit Internetworking Guide*.

Changing Pre-Windows Permissions After Active Directory Installation

If you subsequently upgrade all of your servers and domains to Windows 2000, you can remove the Everyone group from the Pre-Windows 2000 Compatible Access group. Likewise, if you incorporate Windows NT 3.*x* or Windows NT 4.0 server applications into your Windows 2000 domain or if you add a Windows NT 3.*x* or Windows NT 4.0 domain to your forest, you can add the Everyone group to the Pre-Windows 2000 Compatible Access group.

Caution Each time you change the group membership, you must reboot every domain controller in the domain for the change to take effect.

▶ **To add or delete the Everyone group to or from the Pre-Windows 2000 Compatible Access group**

1. On the **Start** menu, point to **Programs**, **Accessories**, and then click **Command Prompt**.
2. To add the Everyone group, at the command prompt, type:

   net localgroup "Pre-Windows 2000 Compatible Access" Everyone /add
3. To delete the Everyone group, at the command prompt, type:

   net localgroup "Pre-Windows 2000 Compatible Access" Everyone /delete

SAM Database

When a primary domain controller in Windows NT 4.0 is upgraded to Windows 2000, the Active Directory Installation Wizard opens at the end of the setup. Accounts in the registry-based SAM database are migrated to Active Directory; the existing SAM is deleted; and a new, smaller registry-based SAM is created that is used for starting the domain controller in Directory Services Restore Mode for system repair.

Note In both mixed-mode and native-mode, when you upgrade a primary domain controller in Windows NT 4.0 to a Windows 2000–based domain controller (as the first domain controller in the domain) and when you upgrade a backup domain controller in Windows NT 4.0 to a Windows 2000–based domain controller, the previous SAM database is deleted so that it is not available for password attacks.

On every new domain controller, whether it is upgraded from an existing Windows NT 4.0–based server or freshly installed as a new operating system, you are prompted for an Administrator account password that is to be used for authenticating to this SAM database when the computer is started in Directory Services Restore Mode.

If Active Directory is removed from the server, the new SAM is available for local user and group accounts on the member server. The computer SID does not change during the installation or removal of Active Directory.

Creating a New Domain

When the new domain is not the first domain in a new forest, its creation depends on other domains in the forest. Various new accounts are created; trust relationships are created; and cross-reference objects are created to incorporate the new domain into the forest.

Note Creating a new forest has no effect on any existing domain and, therefore, does not use a source domain controller during the installation of Active Directory.

Regardless of the type of domain that you are creating, the Active Directory Installation Wizard performs the following operations during the installation process:

- Sets the computer Domain Name System (DNS) root domain name to the name of the new domain by using this format:

  <computerName>.<domainName>...<forestRootDomainName>
- Determines whether the server is joined to a domain. If the computer is a member of a domain (member server), the wizard either removes the computer from the domain and reuses the account or alerts you that the computer account for the server must be removed from the domain by an administrator.
- Creates a computer account in the Domain Controllers container in the new domain. The account is added to the Domain Controllers global group in the Users container. This account allows the computer to authenticate to other domain controllers.
- Applies the password you have provided for the administrator account that is used when the domain controller is started in Directory Services Restore Mode.
- Creates a cross-reference object in the Configuration container. When the configuration directory partition is replicated to the new domain controller, a cross-reference object is created on the domain naming master and is then replicated throughout the forest. This object is used by LDAP to locate resources in other domains. (For more information about cross-reference objects, see "Name Resolution in Active Directory" in this book.)
- Removes the **Start** menu shortcut to the local security settings and adds two new shortcuts to the following Group Policy security setting nodes:
  - Domain security settings for all users and computers.
  - Security settings that are specifically targeted at domain controllers.
- Creates the Sysvol folder that contains the following:
  - Sysvol share.
  - Net Logon shares. (These usually host logon scripts and policy objects for non-Windows 2000–based network clients.)
  - File system junctions.
  - User logon scripts for Windows 2000–based clients and clients that are running Windows 95, Windows 98, or Windows NT 4.0.

- Windows 2000 Group Policy.
- File Replication service (FRS) staging directories and files that are required to be available and synchronized between domain controllers.

During the installation of Active Directory, only the directory folders are created. After Active Directory is installed and the domain controller is restarted, File Replication service (FRS) actually creates the system volume objects in the local directory and enables Sysvol replication on the domain controller.

Note On servers that are upgraded from Windows NT 4.0, files in the original Net Logon share (Repl\Export\Scripts) are moved to the \Sysvol\Sysvol\%Fqdn\Scripts folder in the Sysvol tree.

Operations for the Forest Root Domain

The following operations occur when you create the forest root domain:

- The Schema container and the Configuration container are created.
- The Active Directory Installation Wizard assigns the PDC emulator, RID master, domain naming master, schema master, and infrastructure master roles to the domain controller.

Operations for a New Child Domain

The following operations occur when you create a child domain in an existing tree:

- Verification of the name that you provide as a valid child domain name.
- Location of a source domain controller in the parent domain and synchronization of the system time of the child domain with the system time of the source domain controller.
- Creation of parent-child trust objects in the System folder on both the parent domain and the child domain. These objects (class *trustedDomain*) identify two-way transitive trust relationships between the child domain and the parent domain.
- Replication of the Active Directory Schema container and the Configuration container from the parent domain.

Operations for a New Tree-Root Domain in an Existing Forest

The following operations occur when you create a new domain as a new tree in an existing forest:

- Location of a source domain controller in the forest root domain and synchronization of domain system time with the system time of the source domain controller.
- Creation of a tree-root trust relationship between the tree root domain and the forest root domain, and creation of a *trustedDomain* object in both domains. The tree-root trust relationship is two-way and transitive.
- Assignment of the PDC emulator, relative identifier, and infrastructure single-master operation roles to the domain controller by the Active Directory Installation Wizard.

For more information about trust relationships, see "Active Directory Logical Structure" in this book. For more information about single-master operations, see "Managing Flexible Single-Master Operations" in this book, and see Windows 2000 Server Help.

Operations for an Additional Domain Controller

To add another domain controller to a domain that already exists, install Active Directory on a computer that is running Windows 2000 Server. The same verification and configuration processes occur during the creation of an additional domain controller that occur during the creation of a new domain. There are no specific namespace or TCP/IP checks. If any of these operations fail, the installation of Active Directory cannot proceed.

- Joining of the computer to the domain. If the computer already is joined to the domain, the computer account is joined. If the computer has an account in a different domain, it's unjoined.
- Forced synchronization from the source server to the RID master, which ensures that a relative identifier pool is quickly provided to the new domain controller. The RID master does not have to be available during the installation of Active Directory, but it must be available at some point after the installation to transfer relative identifiers to the new domain controller.

If these operations are successful, the wizard begins the replication process.

Replicating Directory Partitions

When you create a new domain in an existing forest, the schema directory partition and the configuration directory partition are always updated on the new domain controller through replication. When you create an additional domain controller in an existing domain, the domain directory partition also is updated through replication in addition to the schema directory partition and configuration directory partition.

The computer on which you are installing Active Directory uses the domain controller Locator to find a domain controller in the parent domain (for a new child domain) or in its own domain (for an additional domain controller in an existing domain) to act as the source domain controller for replication. The computer queries the source domain controller for the distinguished names of the Configuration container and the Schema container by posting an LDAP query that is based on the NULL distinguished name and retrieving the rootDSE attributes. It replicates the schema directory partition and configuration directory partition (in that order), referenced only by their distinguished names. After the directory partitions have been replicated to the computer on which you are installing Active Directory, the GUIDs of the containers are established from the replicated data, although the directory partitions continue to be referenced solely by the distinguished name string for the duration of the installation process.

Note Failure to fully replicate any of the directory partitions results in the failure to install Active Directory. To ensure complete synchronization, there is a critical point in the replication process beyond which the process cannot be terminated: Prior to replication of the attributes from the domain directory partition, you can cancel the installation process (roll it back). After the replication of the domain directory partition attributes, you cannot cancel the installation process.

For more information about the domain controller Locator, see "Name Resolution in Active Directory" in this book.

DNS Installation and Configuration

When you install the first domain controller in a domain, if DNS is not available on the network, you can elect to have DNS installed and configured automatically during the installation of Active Directory.

The Active Directory Installation Wizard asks whether to install and configure the DNS service automatically if either one of the following conditions is true:

- You are creating a new forest, and the Active Directory Installation Wizard does not find any DNS servers that are running on the network
- You are creating a new domain, and dynamic update is not available.

You also can install the DNS Manager snap-in and use the tool to configure DNS either before or after Active Directory is installed. For more information about automatic DNS configuration and DNS requirements for Active Directory, see "Windows 2000 DNS" in the *TCP/IP Core Networking Guide*.

Operations That Occur Following Installation

The following operations are performed by the Active Directory Installation Wizard after Active Directory has been installed successfully on a computer and the computer has been restarted:

- If a new forest is created, a command to invoke the Comma Separated Value DE (CSVDE) tool is placed in the registry. The command causes all localized display specifiers to be imported into the directory after the computer is restarted.
- If a new domain is created (which includes a new forest) and you elected to install and configure the DNS service, DNS installation and configuration takes place.

- The system volume objects are created in the Sysvol directory and FRS signals Net Logon that Sysvol is ready. Net Logon then shares out Sysvol and publishes the computer as a domain controller. If the computer is an additional domain controller, NtFrs must "seed" (replicate in) the system volume with files and directories from another domain controller in the domain. The domain controller is not advertised as a domain controller and Sysvol is not shared until the "seeding" is complete. SysvolSeeding is an entry in the registry that helps manage the seeding process.

Note Informational File Replication service messages are generated in the Services log in Event Viewer that documents events during Sysvol seeding. Use Event Viewer to view messages that are related to the creation and seeding of Sysvol.

For more information about Adminpak.msi, managing servers remotely, and NetShell, see Windows 2000 Server Help.

Removing Active Directory

Use the same application to remove Active Directory that you use to install it—the Active Directory Installation Wizard. When you start the wizard on a domain controller, the system is identified as a server that contains Active Directory, and the wizard prompts you for the information that is required to remove Active Directory.

Note For all domain controllers, the Active Directory Installation Wizard removes the shortcuts to Group Policy security settings, and it restores the shortcut on the **Administrative Tools** menu to provide access to the local security settings for the member server or for the stand-alone server.

Figure 2.10 shows the pathways that are followed by the wizard to change a computer from a domain controller to either a stand-alone server or a member server.

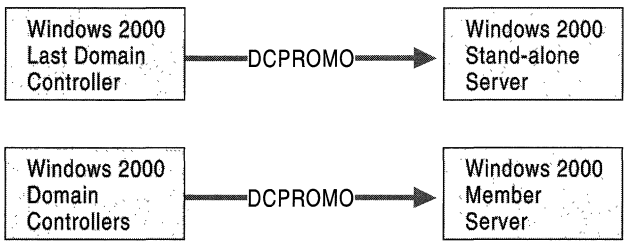


Figure 2.10 Servers That Are Created by Removing Active Directory

Administrative Credentials

To remove Active Directory, you must provide administrative credentials as follows:

- To remove Active Directory from a domain controller that is the last domain controller in a child domain, you must provide enterprise administrator credentials or be a member of the Enterprise Admins group.
- To remove Active Directory from a domain controller that is the last domain controller in a tree-root domain, you must provide credentials for or be logged on as a member of the Enterprise Admins group.
- To remove Active Directory from a domain controller that is the last domain controller in the forest, you must log on to the domain as Administrator or as a member of the Domain Admins group.
- To remove Active Directory from a domain controller that is not the last domain controller in the domain, you do not have to provide credentials. However, you must be logged on as a member of either the Domain Admins group or the Enterprise Admins group.

Removal from an Additional Domain Controller or the Last Domain Controller

When you remove Active Directory from either an additional domain controller or from the last domain controller in the domain, the following operations are common to both procedures. If any operation fails, the removal of Active Directory cannot proceed.

- Replication of changes to the configuration directory partition and the schema directory partition. For an additional domain controller, replication of changes to the configuration, schema, and domain directory partitions.
- Transfer of any single-master roles that the domain controller is holding to another domain controller.

Note In the case of the last domain controller in the domain, transfer would apply only to the forestwide, single-master roles of a schema master or a domain-naming master.

- Removal of the system volume objects from the directory database; removal of the system volume objects from the NtFrs database; and deletion of the Sysvol directory hierarchy (NtFrs). NtFrs requests that Net Logon remove the share from the system volume.
- Removal the NTDS Settings object and cross-reference objects.
- Updating of DNS to remove the domain controller Locator records. (When the NTDS Settings object is deleted, the DSA notifies the Net Logon service, and the Net Logon service removes the records.)

- Creation of the local SAM database in the same manner as during a fresh installation, including creation of the administrator account and setting the password.
- Modification of the LSA membership policy to distinguish whether the computer is a stand-alone server or a member server.
- Stopping Net Logon and other services. The same services that were started during the installation of Active Directory procedure are stopped. Services that relate only to the directory service are configured to not start automatically.

Removal of an Additional Domain Controller

The following operations are specific to removing an additional domain controller. If any operation fails, domain controller demotion cannot proceed.

- Location of a source domain controller in the same domain where the additional domain controller account exists and connecting to it in order to replicate changes.
- Setting the computer account type to member server and moving the computer account for the additional server from the Domain Controllers container to the Computers container.

Removal of the Last Domain Controller

The following operations are specific to removing the last domain controller in the domain. If any operation fails, domain controller demotion cannot proceed.

- Checking that no child domains exist. If none is found, Active Directory removal proceeds.
- Location of a source domain controller in the parent domain and connecting to that source domain controller in order to replicate changes.
- Removal of Active Directory objects from the forest that are specific to this domain. The wizard contacts the domain naming master and removes the NTDS Settings and cross-reference objects.
- Removal of trust objects on the parent server. The *trustedDomain* objects in the System folder are deleted.
- Placement of the server in a workgroup called "Workgroup."

If the NTDS Settings object is not removed successfully from Active Directory (for example, if a server fails during the removal of Active Directory), you must remove the object manually. For information about removing configuration data when the removal of Active Directory is not successful, see "Active Directory Diagnostics, Troubleshooting, and Recovery" in this book.

Unattended Setup for Installation or Removal of Active Directory

You can use an *answer file* to run the Active Directory Installation Wizard without your having to be present for interactive responses. Create the answer file, and then run it at a command prompt on the installation computer. The instructions for creating the answer file are located in the "Microsoft Windows 2000 Guide to Unattended Setup" (Unattend.doc) on the Windows 2000 operating system CD. The Unattend.doc file is part of the Deploy.cab file in the \Support\Tools folder. In Windows 98 or Windows 2000, use Windows Explorer to extract this document. In Windows 95 and earlier, or from MS-DOS, use the **Extract** command to open the file. You can save the Unattend.doc file by any name in any convenient location.

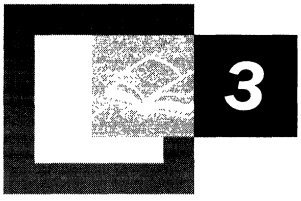
The answer text file contains all of the parameters that the Active Directory Installation Wizard needs to install Active Directory, including the domain type (additional or new domain) and the configuration of the domain that is being created (new forest, new tree root, or new child). Add the parameter values to the answer file.

An answer file can be used to install Windows 2000 Server and can include the installation of Active Directory. Alternatively, you can create an answer file that contains only the [DCInstall] section of the Unattend.doc file. This answer file can be run after Windows 2000 Server Setup is complete and after you have logged on to the system. To run the answer file, at the command prompt, type the following:

dcpromo /answer: <*answer_file*>

In addition to instructions for creating an answer file, the Unattend.doc file contains sample answer files for installing Active Directory. For more information about using answer files, see "Automating Server Installation and Upgrade" in the *Deployment Planning Guide*.

CHAPTER 3



Name Resolution in Active Directory

Finding information in Active Directory™, the Microsoft® Windows® 2000 directory service, involves first locating an Active Directory server (domain controller) for logging on to a domain and then finding the information that you need in Active Directory. Both processes use name resolution. When you are locating a domain controller, the Domain Name System (DNS) resolves (by DNS name resolution) a domain name or computer name to an Internet Protocol (IP) address. During the search for information in Active Directory, Windows 2000 resolves (by Lightweight Directory Access Protocol [LDAP] name resolution) a distinguished name to a domain controller that holds the entry for that name.

In This Chapter

Related Information in the Resource Kit

- For more information about DNS, Transmission Control Protocol/Internet Protocol (TCP/IP) networks, subnets, and subnet masks, see the *Microsoft® Windows® 2000 Server Resource Kit TCP/IP Core Networking Guide*.
- For information about planning and deploying of domains and domain controllers, see the *Microsoft® Windows® 2000 Server Resource Kit Deployment Planning Guide*.

Locating Active Directory Servers

When an application requests access to Active Directory, an Active Directory server (domain controller) is located by a mechanism called the domain controller locator (Locator). Locator is an algorithm that runs in the context of the Net Logon service. Locator can find domain controllers by using DNS names (for IP/DNS-compatible computers) or by using Network Basic Input/Output System (NetBIOS) names (for computers that are running Microsoft® Windows® version 3.*x*, Microsoft® Windows® for Workgroups, Microsoft® Windows NT® version 3.5 or later, Microsoft® Windows® 95, Microsoft® Windows® 98, or for computers on a network where IP transport is not available).

Note In this chapter, the term "Windows NT 4.0–compatible Locator" refers to the locator process that is used by clients that are running Windows 3.*x*, Windows for Workgroups, Windows NT 3.5 or later, Windows 95, or Windows 98 to locate a domain controller in any domain, as well as by clients that are running Windows 2000 to locate a domain controller that is in either a Microsoft® Windows NT® version 3.51 domain or Microsoft® Windows NT® version 4.0 domain.

The focus of this chapter is on the process of locating a domain controller. For more information about DNS resolution of host IP addresses, see "Introduction to DNS" in the *TCP/IP Core Networking Guide*.

Domain Controller Name Registration

Every Windows 2000–based domain controller registers two types of names at startup:

- A DNS domain name with the DNS service (for example, noam.reskit.com).
- A NetBIOS name with Windows Internet Name Service (WINS) or another transport-specific service (for example, noam).

When a user starts a computer and logs on to a domain, the computer must do one of two things:

- If the name of the logon domain is a DNS name, the computer must query DNS to find a domain controller with which to authenticate.
- If the name of the logon domain is a NetBIOS name, the computer must send a mailslot message to find a domain controller for the specified domain.

After the computer has found a domain controller, the information is cached so that a new query is not required for subsequent logon sessions.

DNS Domain Name Registration

Active Directory supports dynamic registration of domain controller addresses in DNS. After Active Directory has been installed during domain controller creation, the Net Logon service dynamically creates records in the DNS database that are used to locate the server. Dynamic update (described in Request for Comments (RFC) 2136) is a recent addition to the DNS standard; this addition to the standard defines a protocol for dynamically updating a DNS server with new or changed resource record values. Before the advent of this new protocol, administrators had to manually create the records that are stored on DNS servers. The implementation of DNS server that is included with Windows 2000 supports dynamic updates, as does the Berkeley Internet Name Domain (BIND) version 8.*x* implementation of DNS. (For more information about BIND DNS, see "Windows 2000 DNS" in the *TCP/IP Core Networking Guide*.)

Every Windows 2000–based domain controller dynamically registers service records (SRV records) in DNS, which allow servers to be located by service type (for example, LDAP) and protocol (for example, Transmission Control Protocol [TCP]). Because domain controllers are LDAP servers that communicate over TCP, SRV records can be used to find the DNS computer names of domain controllers. In addition to registering LDAP-specific SRV records, Net Logon also registers Kerberos v5 authentication protocol–specific SRV records to enable locating servers that run the Kerberos Key Distribution Center (KDC) service. (For more information about the Kerberos v5 authentication protocol and the KDC, see "Authentication" in this book.)

Every Windows 2000–based domain controller also dynamically registers a single host resource record (an A resource record), which contains the name of the domain (*DnsDomainName*) where the domain controller is and the IP address of the domain controller. The A resource record makes it possible for clients that do not recognize SRV records to locate a domain controller by means of a generic host lookup.

You can disable the Net Logon registration of an A resource record that maps the Active Directory domain name to the IP address of the domain controller. For example, if a Web server registers the same name as the name of an Active Directory domain, you do not want non-Web servers to register A resource records for this name. Otherwise, if the Web browser located the domain controller instead of the Web server, the browser would receive the message that the site for which it was searching was not found. In another example, if a mail server is not enabled to do mail exchanger (MX) resource record lookup and, therefore, relies on A resource records for DNS lookup, the names that are used for mail servers must not be identical to the names that are used by other services, such as Active Directory.

▶ To disable Net Logon registration of the A record for a domain controller

1. On the **Start** menu, click **Run**.
2. Type **regedt32.exe** or **regedit.exe**, and then click **OK**.
3. In the registry editor, navigate to HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services \Netlogon\Parameters.
4. If the entry exists, double-click the **DnsRegisterARecords** entry.
5. In the **DWORD Editor** dialog box (in Regedt32.exe) or the **Edit DWORD Value** (in Regedit.exe), type **0** in the text box, and then click **OK**.
6. If the entry does not exist, create the entry as follows:
   - In Regedt32.exe, on the **Edit** menu, click **Add Value**.

     In the **Value Name** box, type **DnsRegisterARecords**.

     In the **Data Type** drop-down list box, click **REG_DWORD**, and then click **OK**.

     In the **DWORD Editor** dialog box, type **0** in the **Data** box, and then click **OK**.

     –Or–
   - In Regedit.exe, on the **Edit** menu, click **New**.

     Click **DWORD Value**.

     Type **DnsRegisterARecords** for the value name. A value of **0** is assigned automatically.
7. Close the registry editor.

Caution Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. There are programs available in Control Panel or Microsoft Management Console (MMC) for performing most administrative tasks. These programs provide safeguards that prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your system. Registry editors bypass the standard safeguards that are provided by these administrative tools. Modifying the registry is recommended only when no administrative tool is available. Before you make changes to the registry, it is recommended that you back up any valuable data on the computer. For instructions about how to edit registry entries, see Help for the registry editor that you are using. For more information about the registry, see the *Microsoft Windows 2000 Resource Kit* Technical Reference to the Windows 2000 Registry (Regentry.chm).

NetBIOS Domain Name Registration

A domain controller registers its NetBIOS name (*DomainName*[1C]) by broadcasting or directing a NetBIOS name registration request to a NetBIOS name server, such as a WINS server. Registering the NetBIOS name makes it possible for Windows-based clients that are not DNS-enabled to find the domain controllers that are running Windows 2000, Windows NT 4.0, or Windows NT 3.51. In this case, the client finds the domain controller by sending a Net Logon mailslot request that is based on the NetBIOS domain name.

Note NetBIOS recognizes domain controllers by the [1C] registration.

For more information about registering names with WINS, see "Windows Internet Name Service" in the *TCP/IP Core Networking Guide*.

SRV Resource Records

When a Windows 2000–based domain controller starts up, the Net Logon service uses dynamic updates to register SRV resource records in the DNS database, as described in "A DNS RR for specifying the location of services (DNS SRV)." For more information about this draft, see the Internet Engineering Task Force (IETF) link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. Follow the links to Internet Drafts, and then use a keyword search.

The SRV record is used to map the name of a service (in this case, the LDAP service) to the DNS computer name of a server that offers that service. In a Windows 2000 network, an LDAP resource record locates a domain controller.

A workstation that is logging on to a Windows 2000 domain queries DNS for SRV records in the general form:

_Service._Protocol.DnsDomainName

Active Directory servers offer the LDAP service over the TCP protocol; therefore, clients find an LDAP server by querying DNS for a record of the form:

_ldap._tcp.*DnsDomainName*

Note The service and protocol strings require an underscore (_) prefix to prevent potential collisions with existing names in the namespace.

_msdcs Subdomain

There are possible implementations of LDAP servers other than Windows 2000–based domain controllers. There are also possible implementations of LDAP directory services that employ Global Catalog servers but are not servers that are running Windows 2000. To facilitate locating Windows 2000–based domain controllers, in addition to the standard *_Service._Protocol.DnsDomainName* format, the Net Logon service registers SRV records that identify the well-known server-type pseudonyms "dc" (domain controller), "gc" (Global Catalog), "pdc" (primary domain controller), and "domains" (globally unique identifier, or GUID) as prefixes in the _msdcs subdomain. This Microsoft-specific subdomain allows location of domain controllers that have Windows 2000–specific roles in the domain or forest, as well as the location by GUID when a domain has been renamed. To accommodate locating domain controllers by server type or by GUID (abbreviated "dctype"), Windows 2000–based domain controllers register SRV records in the following form:

*_Service._Protocol.DcType.*__msdcs.__*DnsDomainName*

The addition of the _msdcs subdomain means that two sets of DNS names can be used to find an LDAP server: *DnsDomainName* is used to find an LDAP server or Kerberos server that is running TCP (or, in the case of a Kerberos server, either TCP or the User Datagram Protocol [UDP]), and the subdomain _msdcs.*DnsDomainName* is used to find an LDAP server that is running TCP and also functioning in a particular Windows 2000 role. The name "_msdcs" is reserved for locating domain controllers. The single keyword "_msdcs" was chosen to avoid cluttering the DNS namespace unnecessarily. Other constant, well-known names (pdc, dc, and gc) were kept short to avoid exceeding the maximum length of *DnsDomainName*.

SRV Records Registered by Net Logon

The list that follows provides the definitions of the names associated with registered SRV records. It also describes the lookup criteria supported by each record and the checks performed by Net Logon as each record is registered. Text in bold type denotes constant record components; text in italic type denotes variable names.

In the descriptions of registered SRV records, *DnsDomainName* refers to the DNS domain name that is used during creation of the domain controller when the domain tree is joined or created (that is, while the computer is running the Active Directory Installation Wizard). *DnsForestName* refers to the DNS domain name of the forest root domain.

The following is a list of the owner names of the SRV records that are registered by Net Logon. An owner name is the name of the DNS node to which the resource record pertains.

_ldap._tcp.*DnsDomainName***.**

Allows a client to locate a server that is running the LDAP service in the domain named by *DnsDomainName*. The server is not necessarily a domain controller—that is, the only assumption that can be made about the server is that it supports the LDAP application programming interface (API). All Windows 2000 Server–based domain controllers register this SRV record (for example, _ldap._tcp.reskit.com.).

_ldap._tcp.*SiteName***._sites.***DnsDomainName***.**

Allows a client to locate a server that is running the LDAP service in the domain named in *DnsDomainName* in the site named by *SiteName*. *SiteName* is the relative distinguished name of the site object that is stored in the Configuration container in Active Directory. The server is not necessarily a domain controller. All Windows 2000 Server–based domain controllers register this SRV record (for example, _ldap._tcp.charlotte._sites.reskit.com.).

_ldap._tcp.dc._msdcs.*DnsDomainName***.**

Allows a client to locate a domain controller (dc) of the domain named by *DnsDomainName*. All Windows 2000 Server–based domain controllers register this SRV record.

_ldap._tcp.*SiteName***._sites.dc._msdcs.***DnsDomainName***.**

Allows a client to locate a domain controller for the domain named by *DnsDomainName* and in the site named by *SiteName*. All Windows 2000 Server–based domain controllers register this SRV record.

_ldap._tcp.pdc._msdcs.*DnsDomainName***.**

Allows a client to locate the server that is acting as the primary domain controller (also known as a "PDC") in the mixed-mode domain named in *DnsDomainName*. Only the PDC emulator master of the domain (the Windows 2000–based domain controller that advertises itself as the primary domain controller to computers that need a primary domain controller) registers this SRV record.

_ldap._tcp.gc._msdcs.*DnsForestName***.**

Allows a client to locate a Global Catalog (gc) server for this forest. Only domain controllers that are functioning as Global Catalog servers for the forest named in *DnsForestName* register this SRV record (for example, _ldap._tcp.gc._msdcs.reskit.com.).

_ldap._tcp.*SiteName***._sites.gc._msdcs.***DnsForestName***.**

Allows a client to locate a Global Catalog (gc) server for this forest in the site named in *SiteName*. Only domain controllers that are serving as Global Catalog servers for the forest named in *DnsForestName* register this SRV record (for example, _ldap._tcp.charlotte._sites.gc._msdcs.reskit.com.).

_gc._tcp.*DnsForestName***.**

Allows a client to locate a Global Catalog (gc) server for this domain. The server is not necessarily a domain controller. Only a server that is running the LDAP service and functioning as the Global Catalog server for the forest named in *DnsForestName* registers this SRV record (for example, _gc._tcp.reskit.com.).

Note In Windows 2000, a Global Catalog server is a domain controller. Other non-Windows 2000 implementations of directory services can also register servers as Global Catalog servers.

_gc._tcp.*SiteName***._sites.***DnsForestName***.**

Allows a client to locate a Global Catalog (gc) server for this forest in the site named in *SiteName*. The server is not necessarily a domain controller. Only a server that is running the LDAP service and functioning as the Global Catalog server for the forest named in *DnsForestName* registers this SRV record (for example, _gc._tcp.charlotte._sites.reskit.com.).

_ldap._tcp.*DomainGuid***.domains._msdcs.***DnsForestName***.**

Allows a client to locate a domain controller in a domain on the basis of its GUID. A GUID is a 128-bit number that is automatically generated for referencing objects in Active Directory—in this case, the domain object. This operation is expected to be infrequent; it occurs only when the *DnsDomainName* of the domain has changed, the *DnsForestName* is known, and *DnsForestName* has not also been renamed (for example, _ldap._tcp.4f904480-7c78-11cf-b057-00aa006b4f8f.domains._msdcs.reskit.com.). All domain controllers register this SRV record.

_kerberos._tcp.*DnsDomainName***.**

Allows a client to locate a server that is running the Kerberos KDC service for the domain that is named in *DnsDomainName*. The server is not necessarily a domain controller. All Windows 2000 Server–based domain controllers that are running an RFC 1510–compliant Kerberos KDC service register this SRV record.

_kerberos._udp.*DnsDomainName***.**

Same as **_kerberos._tcp.***DnsDomainName*, except that UDP is implied.

_kerberos._tcp.*SiteName***._sites.***DnsDomainName***.**

Allows a client to locate a server that is running the Kerberos KDC service for the domain that is named in *DnsDomainName* and is also in the site named in *SiteName*. The server is not necessarily a domain controller. All Windows 2000 Server–based domain controllers that are running an RFC 1510–compliant Kerberos KDC service register this SRV record.

_kerberos._tcp.dc._msdcs.*DnsDomainName***.**

Allows a client to locate a domain controller that is running the Windows 2000 implementation of the Kerberos KDC service for the domain named in *DnsDomainName*. All Windows 2000 Server–based domain controllers that are running the KDC service (that is, that implement a public key extension to the Kerberos v5 protocol Authentication Service Exchange subprotocol) register this SRV record.

_kerberos.tcp.*SiteName***._sites.dc._msdcs.***DnsDomainName***.**

Allows a client to locate a domain controller that is running the Windows 2000 implementation of the Kerberos KDC service for the domain that is named in *DnsDomainName* and that is also in the site named in *SiteName*. All Windows 2000 Server–based domain controllers that are running the KDC service (that is, that implement a public key extension to the Kerberos protocol Authentication Service Exchange subprotocol) register this SRV record.

_kpasswd._tcp.*DnsDomainName***.**

Allows a client to locate a Kerberos Password Change server for the domain. All servers that provide the Kerberos Password Change service (which includes all Windows 2000–based domain controllers) register this name. This server at least conforms to "Kerberos Change Password Protocol." (For more information about this draft, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. Use a keyword search to locate the draft.) The server is not necessarily a domain controller. All Windows 2000 Server–based domain controllers that are running an RFC 1510–compliant Kerberos KDC service register this SRV record.

_kpasswd._udp.*DnsDomainName***.**

Same as **_kpasswd._tcp.***DnsDomainName*, except that UDP is implied.

If multiple domain controllers have the same criteria, multiple records exist with the same owner name. A client that is looking for a domain controller with specific criteria would receive all the applicable records from the DNS server. The client would pick one of the returned records to select a domain controller, as described in "A DNS RR for specifying the location of services (DNS SRV)." For more information about this draft, see the Internet Engineering Task Force (IETF) link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. Follow the links to Internet Drafts, and then use a keyword search.

For information about the Kerberos v5 authentication protocol and Kerberos subprotocol extensions, see "Authentication" in this book.

Host Records for Non-SRV-Aware Clients

Net Logon registers the following DNS A records for the use of LDAP clients that do not support DNS SRV records (that is, that are "non-SRV-aware"). The Locator does not use these records.

The following owner names of A (host) records are registered by Net Logon:

DnsDomainName.

Allows a non-SRV-aware client to locate any domain controller in the domain by looking up an A record. A name in this form is returned to the LDAP client through an LDAP referral. (For more information about LDAP referrals, see "LDAP Referrals" later in this chapter.) A non-SRV-aware client looks up the name; an SRV-aware client looks up the appropriate SRV resource record.

gc._msdcs.*DnsForestName.*

Allows a non-SRV-aware client to locate any Global Catalog server in the forest by looking up an A record. A name in this form is returned to the LDAP client through an LDAP referral. A non-SRV-aware client looks up this name; an SRV-aware client looks up the appropriate SRV resource record.

Net Logon also registers a DNS CNAME (alias) record for use by Active Directory replication. The Locator does not use this record.

The owner name of the CNAME record is:

*DsaGuid.***_msdcs.***DnsForestName.*

Allows a client to locate any domain controller in the forest by looking up an A record. The only information that is known about the domain controller is the GUID of the directory system agent (also known as the "DSA") object for the domain controller and the name of the forest in which the domain controller is located. This record is used to facilitate renaming a domain controller.

Other SRV Record Content

The following information is also included in an SRV record:

Priority The priority of the server. Clients attempt to contact the server with the lowest priority.

Weight A load-balancing mechanism that is used when selecting a target host from those that have the same priority. Clients randomly choose SRV records that specify target hosts to be contacted, with probability proportional to the weight

Port Number The port where the server is listening for this service.

Target The fully qualified domain name of the host computer.

The following example illustrates the combined information that is contained in A resource records and SRV resource records. A domain controller named Phoenix in the domain reskit.com has an IP address of 157.55.81.157. It registers the following A records and SRV records with DNS:

```
phoenix.reskit.com   A   157.55.81.157
_ldap._tcp.reskit.com        SRV  0 0 389 phoenix.reskit.com
_kerberos._tcp.reskit.com    SRV  0 0 88 phoenix.reskit.com
_ldap._tcp.dc._msdcs.reskit.com  SRV  0 0 389 phoenix.reskit.com
_kerberos._tcp.dc._msdcs.reskit.com  SRV  0 0 88 phoenix.reskit.com.
```

When the appropriate SRV records and A records are in place, a DNS lookup of _ldap._tcp.dc._msdcs.reskit.com returns the names and addresses of all domain controllers in the domain.

For more information about A records, SRV records, DNS, and dynamic updates, see "Introduction to DNS" and "Windows 2000 DNS" in the *TCP/IP Core Networking Guide*.

Domain Controller Location Process

Each Windows 2000–based domain controller registers its DNS domain name on the DNS server and registers its NetBIOS name by using a transport-specific mechanism (for example, WINS). Therefore, a DNS client locates a domain controller by querying DNS, and a NetBIOS client locates a domain controller by querying the appropriate transport-specific name service. Because the code for the Windows 2000 IP/DNS-compatible Locator and the Windows NT 4.0–compatible Locator is shared, both DNS clients and NetBIOS clients are supported.

The process for locating a domain controller can be summarized as follows:

1. On the client (the computer that is locating the domain controller), the Locator is initiated as a remote procedure call (RPC) to the local Net Logon service. The Locator API (DsGetDcName) is implemented by the Net Logon service.
2. The client collects the information that is needed to select a domain controller and passes the information to the Net Logon service by using the DsGetDcName API.
3. The Net Logon service on the client uses the collected information to look up a domain controller for the specified domain in one of two ways:
   - For a DNS name, Net Logon queries DNS by using the IP/DNS-compatible Locator—that is, DsGetDcName calls DnsQuery to read the SRV records and A records from DNS after it appends an appropriate string to the front of the domain name that specifies the SRV record.
   - For a NetBIOS name, Net Logon performs domain controller discovery by using the Windows NT 4.0–compatible Locator—that is, by using the transport-specific mechanism (for example, WINS).

Note In Windows NT 4.0 and earlier, "discovery" is a process for locating a domain controller for authentication in either the primary domain or a trusted domain.

---

4. The Net Logon service sends a datagram to the discovered domain controllers ("pings" the computers) that register the name. For NetBIOS domain names, the datagram is implemented as a mailslot message. For DNS domain names, the datagram is implemented as an LDAP UDP search.
5. Each available domain controller responds to the datagram to indicate that it is currently operational and then returns the information to DsGetDcName.
6. The Net Logon service returns the information to the client from the domain controller that responds first.
7. The Net Logon service caches the domain controller information so that it is not necessary to repeat the discovery process for subsequent requests. Caching this information encourages the consistent use of the same domain controller and, thus, a consistent view of Active Directory.

DsGetDcName API

The following parameters are the DsGetDcName API parameters that Net Logon uses to collect information from the client and to compose the DNS or WINS query. As described in "Domain Controller Location Process" earlier in this chapter, the API is called remotely in a datagram that is sent to the discovered domain controllers, and the domain controller provides the information to the client.

ComputerName The name of the computer that collects the information. The value of this parameter is usually NULL, which denotes the local computer. The DsGetDcName API is passed by RPC to the specified computer.

DomainName The name of the domain to be queried. This name can be either a DNS-style name (for example, reskit.com.) or a flat, NetBIOS-style name (for example, reskit). If a DNS-style name is specified, the name can be specified with or without a trailing dot.

DomainGuid The GUID of the domain being queried. This value is used when a domain has been renamed. If this value is specified and *DomainName* has been renamed, DsGetDcName attempts to locate a domain controller in the domain that has the specified *DomainGuid.*

SiteName The name of the site in which the domain controller that is returned should be located. This parameter is usually not specified. When the site is not specified, the domain controller that is returned is in the site that is closest to the one in which *ComputerName* is located.

Flags Additional information that the application can use to process the request. Flags include the following:

DS_FORCE_REDISCOVERY. Requires that a domain controller be determined, even if a domain controller is currently known in the cache. This flag can be used when an additional domain controller becomes available or when an existing domain controller has been detected to be unavailable. This function guarantees only the domain controller that was returned when the domain controller was initially entered into the cache. The DS_FORCE_REDISCOVERY flag should not be specified unless this function has been called recently without the flag. An attempt to gain access to the cached domain controller should be made. Only if this initial attempt to gain access fails should the flag be used to call the function again.

DS_DIRECTORY_SERVICE_REQUIRED. Requires that the returned domain controller support Directory Server API (is running Windows 2000 Server).

DS_DIRECTORY_SERVICE_PREFERRED. Prefers that the returned domain controller support Active Directory (is running Windows 2000 Server). If no such domain controller is available, a domain controller that is running Windows NT 4.0 or earlier is returned. If no domain controller that supports a directory service is available, DsGetDcName returns the name of the closest non-Active Directory domain controller; however, DsGetDcName returns the non-Active Directory domain controller information only after the attempt to find an Active Directory domain controller has timed out.

DS_GC_SERVER_REQUIRED. Requires that the returned domain controller be a Global Catalog server for the forest of domains that has the specified domain as the root. This flag cannot be set if the DS_PDC_REQUIRED flag is set.

DS_PDC_REQUIRED. Requires that the returned domain controller be the primary domain controller for the domain. This flag cannot be set if the DS_GC_SERVER_REQUIRED flag is set. If this flag is specified, the DS_DIRECTORY_SERVICE_PREFERRED flag and DS_WRITABLE_REQUIRED flag are ignored.

DS_WRITABLE_REQUIRED. Requires that the returned domain controller host a writable copy of Active Directory (or Security Accounts Manager [SAM]). If the specified *DomainName* is a NetBIOS name, this flag causes DsGetDcName to find either a primary domain controller or a Windows 2000 Server–based domain controller. If the specified *DomainName* is a DNS name, this flag is ignored.

DS_IP_REQUIRED. Requires that the IP address of the discovered domain controller be returned.

DS_KDC_REQUIRED. Requires that the returned domain controller currently be running the KDC service.

DS_TIMESERV_REQUIRED. Requires that the returned domain controller be currently running the Windows Time Service.

DS_GOOD_TIMESERV_PREFERRED. Prefers that the returned domain controller be a "reliable" time server. The Windows Time Service can be configured to declare one or more domain controllers as "reliable" time servers. This flag is intended for use only by the Windows Time Service. The behavior of the flag is subject to change to achieve the implementation that best supports the Windows Time Service (W32time).

DS_IS_FLAT_NAME. Specifies that the *DomainName* parameter be a NetBIOS name. As such, the IP/DNS-compatible Locator is not tried, and the Windows NT 4.0–compatible Locator is used. (For more information about locating a domain controller by using a NetBIOS name, see "Windows NT 4.0–Compatible Locator Process for Non-IP/DNS Clients" later in this chapter.) This flag cannot be specified with the DS_IS_DNS_NAME flag. It is valid to set neither DS_IS_FLAT_NAME nor DS_IS_DNS_NAME; however, DsGetDcName() takes longer to find a domain controller because it must try both the DNS-style and NetBIOS names. In addition, it is potentially ambiguous to specify neither flag. For example, if you specify a domain name of "reskit," a domain with a NetBIOS name of "reskit" exists on your network, and a different domain with a DNS-style name of "reskit" also exists on your network, DsGetDcName() might find a domain controller in either domain.

DS_IS_DNS_NAME. Specifies that the *DomainName* parameter is a DNS name. (For more information about the *DomainName* parameter, see "Windows NT 4.0–Compatible Locator Process for Non-IP/DNS Clients" later in this chapter.) This flag cannot be specified by using the DS_IS_FLAT_NAME flag.

DS_AVOID_SELF. Specifies that the domain controller returned should not be the domain controller named by *ComputerName*. If *ComputerName* is not a domain controller, this flag is ignored. This flag can be used to get the name of another domain controller in the domain.

DS_RETURN_FLAT_NAME. Specifies that the returned *DomainControllerName* and returned *DomainName* must be flat names. If a flat name is not available, an error message is returned. This flag cannot be specified by using the DS_RETURN_DNS_NAME flag.

DS_RETURN_DNS_NAME. Specifies that the returned *DomainControllerName* and returned *DomainName* must be DNS names. If a DNS name is not available, an error message is returned. This flag cannot be specified with the DS_RETURN_FLAT_NAME flag. This flag implies the DS_IP_REQUIRED flag.

DS_ONLY_LDAP_NEEDED. Specifies that the server returned must be an LDAP server. The server returned is not necessarily a domain controller. No other services are implied to be present at the server. The server returned does not necessarily have either a writable Configuration container or a writable Schema container. The server returned is not necessarily used to create or modify security principles. This flag can be used with the DS_GC_SERVER_REQUIRED flag to return an LDAP server that also hosts a Global Catalog server. The returned Global Catalog server is not necessarily a domain controller. No other services are implied to be present at the server. If this flag is specified, the following flags are ignored: DS_PDC_REQUIRED, DS_TIMESERV_REQUIRED, DS_GOOD_TIMESERV_PREFERRED, and DS_KDC_REQUIRED.

For more information about the DsGetDcName API and DsGetDcName flags, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Finding a Domain Controller in the Closest Site

During a search for a domain controller, the Locator attempts to find a domain controller in the site closest to the client. When the domain that is being sought is a Windows 2000 domain, the domain controller uses the information stored in Active Directory to determine the closest site. When the domain being sought is a Windows NT 4.0 domain, domain controller discovery occurs when the client starts and uses the first domain controller that it finds.

As described in "SRV Records Registered by Net Logon" earlier in this chapter, each Windows 2000–based domain controller registers DNS records that indicate the site where the domain controller is located. The site name (the relative distinguished name of the site object in Active Directory) is registered in several records so that the various roles the domain controller might perform (for example, Global Catalog server or Kerberos server) can be associated with the domain controller's site. When DNS is used, the Locator searches first for a site-specific DNS record before it begins to search for a DNS record that is not site-specific (thereby preferentially locating a domain controller in that site).

A client computer stores its own site information in the registry, but the computer is not necessarily located physically in the site associated with its IP address. For example, a portable computer that was moved to a new location contacts a domain controller in its home site, which is not the site to which the computer is currently connected. In this situation, the domain controller looks up the client site on the basis of the client IP address by comparing the address to the sites that are identified in Active Directory, and returns the name of the site that is closest to the client. The client then updates the information in the registry.

The domain controller stores site information for the entire forest in the Configuration container. The domain controller uses the site information to check the IP address of the client computer against the list of subnets in the forest. In this way, the domain controller ascertains the name of the site in which the client is assumed to be located, or the site that is the closest match, and returns this information to the client.

Active Directory Site and Subnet Objects

A site is a collection of subnets that have high-speed connections. In Active Directory, a site is defined by a site object in the cn=Sites,cn=Configuration,dc=*ForestRootDomain* container. A subnet is an addressed segment within a site and is represented by an object in the cn=Subnets,cn=Sites,cn=Configuration,dc=*ForestRootDomain* container.

The site in which a domain controller is located is identified in the Configuration container by the domain controller object that is located within the cn=Servers container beneath the site object for a particular site. A domain controller can identify the site of a client by using the subnet object in the Sites container. Each subnet object has a *siteObject* property ("attribute") that links it to a site object; the value of the *siteObject* property is the distinguished name of the site object. This link enables a domain controller to identify clients that have an IP address in the specified subnet as being in the specified site.

Subnet names in Active Directory take the form "network/bits masked" (for example, the subnet object 172.16.72.0/22 has a subnet of 172.16.72.0 and a 22-bit subnet mask). If this subnet had a *siteObject* property value that contained the distinguished name of the Seattle site object, all IP addresses in the 172.16.72.0/22 subnet would be considered to be in the Seattle site. The *siteObject* property is a single value, which implies that a single subnet maps to a single site. However, multiple subnet objects can be linked to the same site object. The directory administrator manually creates subnet objects and, hence, the *siteObject* property value.

The Configuration container (including all of the site and subnet objects in it) is replicated to all domain controllers in the forest. Therefore, any domain controller in the forest can identify the site in which a client is located, compare it to the site in which the domain controller is located, and indicate to the client whether that domain controller's site is the closest site to the client.

For more information about site and subnet objects, see "Active Directory Replication" in this book. For more information about networks, subnets, and subnet masks, see "Introduction to TCP/IP" in the *TCP/IP Core Networking Guide*.

Mapping IP Addresses to Site Names

During Net Logon startup, the Net Logon service on each domain controller enumerates the site objects in the Configuration container. Net Logon on each domain controller is also notified of any changes made to the site objects. Net Logon uses the site information to build an in-memory structure that is used to map IP addresses to site names.

When a client that is searching for a domain controller receives the list of domain controller IP addresses from DNS, the client begins querying the domain controllers in turn to find out which domain controller is available and appropriate. Active Directory intercepts the query, which contains the IP address of the client, and passes it to Net Logon on the domain controller. Net Logon looks up the client IP address in its subnet-to-site mapping table by finding the subnet object that most closely matches the client IP address and then returns the following information:

- The name of the site in which the client is located, or the site that most closely matches the client IP address.
- The name of the site in which the current domain controller is located.
- A bit that indicates whether the found domain controller is located (bit is set) or not located (bit is not set) in the site closest to the client.

The domain controller returns the information to the client. The response also contains various other pieces of information that describe the domain controller. The client inspects the information to determine whether to try to find a better domain controller. The decision is made as follows:

- If the returned domain controller is in the closest site (the returned bit is set), the client uses this domain controller.
- If the client has already tried to find a domain controller in the site in which the domain controller claims the client is located, the client uses this domain controller.
- If the domain controller is not in the closest site, the client updates its site information and sends a new DNS query to find a new domain controller in the site. If the second query is successful, the new domain controller is used. If the second query fails, the original domain controller is used.

If the domain that is being queried by a computer is the same as the domain to which the computer is joined, the site in which the computer resides (as reported by a domain controller) is stored in the computer registry. The client stores this site name in the **DynamicSiteName** registry entry in HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Netlogon\Parameters. Therefore, the DsGetSiteName API returns the site in which the computer is located.

Never change dynamically determined values. To override the dynamic site name, add the **SiteName** entry with the REG_SZ data type in HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Netlogon\Parameters. When a value is present for the **SiteName** entry, the **DynamicSiteName** entry is not used. For more information about **SiteName** and **DynamicSiteName**, see the *Microsoft Windows 2000 Resource Kit* Technical Reference to the Windows 2000 Registry (Regentry.chm).

Caution Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. There are programs available in Control Panel or Microsoft Management Console (MMC) for performing most administrative tasks. These programs provide safeguards that prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your system. Registry editors bypass the standard safeguards that are provided by these administrative tools. Modifying the registry is recommended only when no administrative tool is available. Before you make changes to the registry, it is recommended that you back up any valuable data on the computer. For instructions about how to edit registry entries, see Help for the registry editor that you are using. For more information about the registry, see the *Microsoft Windows 2000 Resource Kit* Technical Reference to the Windows 2000 Registry (Regentry.chm).

If the domain being located is the same as the domain to which the computer is joined and the computer has not physically moved to a different site since the last query, the dynamically determined site name in the registry is the actual site in which the computer is located. As such, the client finds a domain controller in the correct site without having to retry the operation. If the site name in the registry is not the current site of the computer (for example, if the computer is portable), the domain controller location process serves to update the site information in the registry.

Automatic Site Coverage

There is not necessarily a domain controller in every site. For various reasons, it is possible that no domain controller exists for a particular domain at the local site. By default, each domain controller checks all sites in the forest and then checks the replication cost matrix. A domain controller advertises itself (registers a site-related SRV record in DNS) in any site that does not have a domain controller for that domain and for which its site has the lowest-cost connections. This process ensures that every site has a domain controller that is defined by default for every domain in the forest, even if a site does not contain a domain controller for that domain. The domain controllers that are published in DNS are those from the closest site (as defined by the replication topology).

For example, given one domain and three sites, a domain controller for that domain might be located in two of the sites, but there might be no domain controller for the domain in the third site. Replication to the domain that does not have a domain controller in the third site might be too expensive in terms of cost or replication latency. To ensure that a domain controller can be located in the site closest to a client computer, if not the same site, Windows 2000 automatically attempts to register a domain controller in every site. The algorithm that is used to accomplish automatic site coverage determines how one site can "cover" another site when no domain controller exists in the second site.

Determining Site Coverage on the Basis of Cost

Given one domain and sites A, B, and C, site A has no domain controllers for the domain. If a client in site A attempts to locate a domain controller, which domain controller should be returned? The answer depends on which site covers site A for the domain. Site coverage is determined according to site-link costs, and domain controllers register themselves in sites accordingly.

In the example, a site link exists between site A and both of the other sites—that is, the connections between domain controllers in site A, site B, and site C are configured for replication over site links in Active Directory Sites and Services. (For more information about site links and site-link costs, see "Active Directory Replication" in this book.) Costs are associated with site links based on the expense of transferring data over the connections. The administrator uses the speed of the connection between sites to assign a cost to the communication link, and replication uses the cost to establish the least expensive route for replication traffic.

Site A and site B are connected by site link AB. Site A and site C are connected by site link AC, with the following costs:

- Site link AB cost = 50.
- Site link AC cost = 100.

The link between site A and site C has a much higher cost than the link between site A and site C. The administrator configured this cost based on the expensive Integrated Services Digital Network (ISDN) line that connects site A and site C, and the administrator would prefer that resources in site B be used when possible. The site coverage algorithm ensures that a domain controller in site B registers itself as a domain controller for site A. In this way, clients in Site A that are looking for a domain controller find one from site B, instead of possibly finding one from site C. For more information about site link cost, see "Active Directory Replication" in this book.

Site Coverage Algorithm

During registration of SRV records in DNS, the following algorithm is used to determine which domain controllers register site SRV records that designate them as preferred domain controllers in sites that do not have a specific domain represented.

For every domain controller in the forest, follow this procedure:

1. Build a list of *target sites*—sites that have no domain controllers for this domain (the domain of the current domain controller).
2. Build a list of *candidate sites*—sites that have domain controllers for this domain.
3. For every target site, follow these steps:
    1. Build a list of candidate sites of which this domain is a member. (If none, do nothing.)
    2. Of these, build a list of sites that have the lowest site link cost to the target site. (If none, do nothing.)
    - If more than one, break ties (reduce this list to one candidate site) by choosing the site with the largest number of domain controllers.
    - If more than one, break ties by choosing the site that is first alphabetically.
    - Register target-site-specific SRV records for the domain controllers for this domain in the selected site.

Cache Time-out and Closest Site

If a domain member computer requests a domain controller while all domain controllers in its site are offline, the Locator necessarily returns a domain controller in a different site. The location of this domain controller is stored in the client cache. The cache lifetime is controlled by the **CloseSiteTimeout** entry in the registry.

In addition, the domain controller performs authentication, and a secure channel is set up. On subsequent location attempts, the lifetime of the cache and the lifetime of the secure channel are secondary to the location of a domain controller in the closest site.

If the domain controller that is stored in the client cache is not in a site that is close to the client, Net Logon attempts to find a close domain controller when either of the following events occurs:

- An interactive logon process uses pass-through authentication on the secure channel.
- The value in the **CloseSiteTimeout** registry entry has elapsed since the last attempt, and any other attempt is made to use the secure channel (for example, pass-through authentication of network logons).

Thus, Net Logon attempts to find a close domain controller only on demand. The default value of the **CloseSiteTimeout** period is 15 minutes; the maximum value is 49 days, and the minimum value is 60 seconds. The implications of this setting are that if the time-out value is too large, a client never tries to find a close domain controller if there is not one available at startup. If the value of this setting is too small, secure channel traffic is unnecessarily slowed down by discovery attempts.

For more information about creating the **CloseSiteTimeout** entry, see the *Microsoft Windows 2000 Resource Kit* Technical Reference to the Windows 2000 Registry (Regentry.chm).

Clients with No Apparent Site

Sometimes the client pings a domain controller and the client IP address cannot be found in the subnet-to-site mapping table. In this case, the domain controller returns a NULL site name, and the client uses the returned domain controller.

For more information about locating sites, see "Active Directory Replication" in this book.

Types of Locators

On the basis of parameters passed to Net Logon in the DsGetDcName API, the process of locating a domain controller proceeds in one of two ways:

- The IP/DNS-compatible Locator is used if the domain name passed to DsGetDcName is a DNS-compatible name. The Net Logon service on the client looks up the name in DNS (by calling DnsQuery) after it appends an appropriate string to the front of the domain name. The DNS service supports a query for determining the set of domain controllers. If the client site name is known, the client DNS query specifies the site. DNS returns the IP addresses of domain controllers that match the DNS query. The client Net Logon service sends an LDAP UDP message to one or more of the domain controllers that have been returned by DNS in order to determine whether any of the specified domain controllers are running and support the specified domain.
- The Windows NT 4.0–compatible Locator is used if the domain name passed to DsGetDcName is a NetBIOS name. The Net Logon service on the client sends a transport-specific logon request query to locate a domain controller in a particular domain and then sends a mailslot message to one or more of the domain controllers to determine whether any of the domain controllers it found are running and support the specified domain.

For more information about the DsGetDcName Locator API, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. For more information about DNS and IP address resolution, see "Introduction to DNS" and "Windows 2000 DNS" in the *TCP/IP Core Networking Guide*.

IP/DNS-Compatible Locator Process for Windows 2000 Clients

The IP/DNS-compatible Locator uses DNS SRV records to locate the closest domain controller on the basis of the site name registered for a particular domain controller.

Note A site name is registered in DNS as a DNS label, which implies that the site name must syntactically conform to the rules of a DNS label. For example, a label cannot be more than 63 octets long and cannot contain illegal characters such as a dot. (For more information about DNS naming rules, see "Introduction to DNS" and "Windows 2000 DNS" in the *TCP/IP Core Networking Guide*.)

The IP/DNS-compatible Locator algorithm runs in the context of the Net Logon service that is running (typically) on the client. The following process takes place after the Net Logon service has established that a DNS name is being requested.

1. Locator queries DNS (calls DnsQuery) and specifies one of the criteria-specific DNS host names. If the client has site information, it first queries the site-specific DNS names. On the basis of parameters to the DsGetDcName API and the success or failure of a previous lookup, DNS is queried as follows:

Note When DsGetDcName is called without specifying a site, DsGetDcName always attempts to find a domain controller in the site of the client or the site closest to the client, as described in "Automatic Site Coverage" earlier in this chapter. The details of site-specific discovery are omitted in the discussion that follows.

- If the DS_PDC_REQUIRED flag is specified, look up the **_ldap._tcp.pdc._msdcs.***DnsDomainName* name. Return any success or failure to the caller.
- If the DS_GC_SERVER_REQUIRED flag is specified and the *SiteName* parameter is specified, look up the **_ldap._tcp.***SiteName*.**_sites.gc._msdcs.***DnsForestName* name. If no domain controller is found, go to the next step. Otherwise, return any success or failure to the caller.
- If the DS_GC_SERVER_REQUIRED flag is specified and the *SiteName* parameter is not specified, look up the **_ldap._tcp.gc._msdcs.***DnsForestName* name. Return any success or failure to the caller.
- If the DS_KDC_REQUIRED flag is specified and the *SiteName* parameter is specified, look up the **_kerberos._tcp.***SiteName*.**_sites.dc._msdcs.***DnsForestName* name. If no domain controller can be found, go to the next step. Otherwise, return any success or failure to the caller.
- If the DS_KDC_REQUIRED flag is specified and the *SiteName* parameter is not specified, look up the **_kerberos._tcp.dc._msdcs.***DnsForestName* name. Return any success or failure to the caller.
- If the DS_ONLY_LDAP_NEEDED flag is specified and the *SiteName* parameter is specified, look up the **_ldap._tcp.***SiteName*.**_sites.***DnsDomainName* name. If no computer is found, go to the next step. Otherwise, return any success or failure to the caller.

- If the DS_ONLY_LDAP_NEEDED flag is specified and the *SiteName* parameter is not specified, look up the **_ldap._tcp.***DnsDomainName* name. Return any success or failure to the caller.
- If the *SiteName* parameter is specified, look up the **_ldap._tcp.***SiteName***._sites.dc._msdcs.***DnsDomainName* name. If no domain controller is found, go to the next step. Otherwise, return any success or failure to the caller.
- If the *SiteName* parameter is not specified, look up the **_ldap._tcp.dc._msdcs.***DnsDomainName* name. If no domain controller has that name (which is not the same as "If no domain controllers can be found"), go to the next step. Otherwise, return any success or failure to the caller.
- If the DomainGuid parameter is specified, look up the **_ldap._tcp.***DomainGuid***.domains._msdcs.***DnsForestName* name. Otherwise, return any success or failure to the caller.

2. If IP is not supported or DNS is not supported (indicated by an error message that is returned from the DnsQuery API), call the Windows NT 4.0–compatible Locator.
3. If the specified name cannot be found (perhaps because the domain has been renamed), call the Windows NT 4.0–compatible Locator.
4. DNS returns a list of IP addresses that match the target domain in the SRV records (that is, IP addresses of domain controllers in the specified domain) that are sorted by priority and weight, as described the Internet Draft "A DNS RR for specifying the location of services (DNS SRV)." (For more information about this draft, see the Internet Engineering Task Force [IETF] link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. Follow the links to Internet Drafts, and then use a keyword search.) The client pings each IP address in the order returned. The ping is a UDP LDAP query to port 389. The client pings each domain controller from the list. After each ping, the client waits one-tenth of a second for a response to the ping (or to any previous ping), and then pings the next domain controller. Choosing the domain controllers at random provides a first level of load balancing. Doing multiple pings in quick succession ensures that the discovery algorithm terminates in a finite amount of time.
5. The pinging continues until a viable response has been received or all of the returned domain controllers have been tried.
6. When a domain controller responds to the ping, the information supplied in the response is compared to the information specified to DsGetDcName. If the information does not match, the response is ignored.
7. The first domain controller to respond to a ping is returned to the caller.

Windows NT 4.0–Compatible Locator Process for Non-IP/DNS Clients

When the IP/DNS Locator cannot be used, domain location proceeds by using the Windows NT 4.0–compatible Locator. The IP/DNS Locator is not used under any of the following conditions:

- A NetBIOS domain name is supplied to DsGetDcName.
- IPX or NetBEUI is the only available transport.
- A Windows NT 3.51 or Windows NT 4.0 domain controller is being located.
- A Windows 3.1, Windows for Workgroups 3.1, Windows for Workgroups 3.11, Windows NT 3.51, Windows NT 4.0, Windows 95, or Windows 98 client is doing the locating.

In these cases, the Windows NT 4.0–compatible Locator is used to locate a domain controller in the domain. In general, the Windows NT 4.0–compatible Locator works as follows:

1. The client discovers a domain controller in a domain by sending a NETLOGON_SAM_LOGON_REQUEST message to \mailslot\net\ntlogon mailslot on the *DomainName*[1C] NetBIOS name of the domain whose domain controller is being discovered.
2. The *DomainName*[1C] NetBIOS name is registered by every Windows NT domain controller in the domain. The specific name resolution mechanism and datagram delivery mechanism that are used are transport-specific. The client uses the first domain controller that responds to the message.

Note The algorithm described here locates any domain controller in the domain. A similar algorithm, implemented in NetGetDcName in Windows NT 4.0, is used when the caller specifies finding a primary domain controller.

Domain Controller Request

A client requests a domain controller by using the NETLOGON_SAM_LOGON_REQUEST message. In Windows 2000, this message is enhanced to contain the following new information:

- The **NtVersion** field specifies that Windows 2000–specific fields be present and that the response must contain more information.

- The **RequestedDomainName** field is used by Net Logon to identify the queried domain if the **RequestedDomainGuid** field is not present. This field immediately follows the existing **DomainSid** field.
- The **RequestedDomainGuid** field is used by Net Logon to identify the queried domain. This field immediately follows the **RequestedDomainName** field.

A Windows NT 3.51–based or Windows NT 4.0–based domain controller ignores the additional information in the query message.

Note When a Windows 2000 domain has an external trust relationship with a non–Windows 2000 domain (a Windows NT 3.51 or Windows NT 4.0 domain), discovery of domain controllers in the external trusted domain is performed immediately when the client starts. Discovery at logon is not possible because the accounts in the trusted domain, which is outside the forest, are not available in the Global Catalog. The Global Catalog is required for logging on to the domain.

Domain Controller Response

A domain controller responds to the NETLOGON_SAM_LOGON_REQUEST message with a NETLOGON_SAM_LOGON_RESPONSE mailslot message as follows:

1. A Windows NT 3.51–based or Windows NT 4.0–based domain controller returns a NETLOGON_SAM_LOGON_RESPONSE mailslot message.
2. A Windows 2000–based domain controller returns a NETLOGON_SAM_LOGON_RESPONSE mailslot message to a client that is running Microsoft® Windows® version 3.1, Microsoft® Windows® for Workgroups version 3.1, Microsoft® Windows® for Workgroups version 3.11, Windows NT 3.51, or Windows NT 4.0.
3. A Windows 2000–based domain controller returns a NETLOGON_SAM_LOGON_RESPONSE_EX message to a Windows 2000–based client. All characters in this message are in the UTF-8 character set. The response message contains the following additional information:
   - The **NtVersion** field indicates that Windows 2000–specific fields are present.
   - The **DomainGuid** field returns the GUID of the domain.
   - The **DnsDomainName** field returns the DNS name of the domain.
   - The **DnsForestName** field returns the DNS domain name of the forest in which the domain controller is located.

- The **DcSiteName** field returns the name of the site in which the domain controller is located.
- The **ClientSiteName** field returns the name of the site in which the client is located.
- The **DcSockAddr** field returns the IP address of the domain controller.
- The **Flags** field indicates the following information about the domain controller:

  Whether it is the primary domain controller.

  Whether it supports the Global Catalog.

  Whether it supports Active Directory.

  Whether it is in the site closest to the client.

The Windows NT 4.0–compatible Locator cannot completely implement the DsGetDcName flag, which requires that a domain controller that supports the directory service API is returned (that is, the returned domain controller must be running Windows 2000). When the DomainGuid parameter is NULL, WINS is configured, and no Windows 2000–based domain controllers are among the 25 closest domain controllers in the domain, then DsGetDcName fails as though no domain controller were available.

Requerying to Find the Closest Site

A client that is running Windows 2000 and using the Windows NT 4.0–compatible Locator to find a domain controller attempts to find a domain controller in the site closest to it. After the client finds a domain controller, the mailslot response is inspected to determine the following:

- Whether the DNS domain name of the domain was returned. (The value is not TRUE if the domain controller is running Windows NT 3.51 or Windows NT 4.0.)
- Whether the ClientSiteName was returned. (The value is not TRUE if there is no subnet object that matches the IP address of the client.)
- Whether the domain controller did not indicate that it is in the site closest to the client.

If all of these conditions have the value TRUE, DsGetDcName calls DsGetDcName again and passes the DNS domain name and the site name. This new call to DsGetDcName uses the IP/DNS Locator to try to find a domain controller in the named site. If the new call finds a domain controller, that domain controller is returned to the client as the result of the original DsGetDcName call. If the new call does not find a domain controller, the original domain controller is returned.

Finding Information in Active Directory

Client applications use various mechanisms to find information in Active Directory. Most requests for directory objects are carried out either through the Active Directory Service Interfaces (ADSI) LDAP provider or through the LDAP API. Ultimately, every request is subject to the LDAP rules for locating objects. Active Directory processes LDAP requests for locally stored directory information (that is, information specific to the current domain directory partition) and implements a referral mechanism to locate objects stored in other directory partitions. If the object does not exist, an error is returned that states that the object is not in the directory.

Resolving Names in Directory Operations

When any directory operation is requested by a client, the domain controller that is contacted resolves names by using its "knowledge" of the entire directory to determine whether the domain controller can complete the operation or whether it must refer the client to another server for part or all of the operation.

LDAP finds an object in the directory according to the path that is specified in the distinguished name (also known as the "DN") of the object. Every object is stored in the directory database according to its relative distinguished name (also known as the "RDN") and parent identifier, not according to its distinguished name. A distinguished name is a series of relative distinguished names that lead from the object's relative distinguished name to the relative distinguished name at the top of the distinguished name hierarchy. Therefore, if you know the relative distinguished name of an object, you can always determine the full distinguished name by following the references to the parent objects and ultimately to the root object. For example, the distinguished name of a user object might be cn=*UserName*,ou=*OrganizationalUnit*,dc=*DomainName*,...dc=*DomainName*, where the series of relative distinguished names denoted by dc=*DomainName* identifies the DNS domain of the object. This portion of the distinguished name can be matched to the tree of domain names that is formed by certain attribute values that are stored in cn=Partitions,cn=Configuration, dc=*ForestRootDomain*.

Note The objects in cn=Partitions,cn=Configuration,dc=*ForestRootDomain* are cross-reference objects; they contain information that Active Directory can use to construct the directory tree hierarchy.

Because every domain controller has the information about all directory partitions in the forest, splitting a distinguished name into a suffix (which identifies the relative path within the domain) and a prefix (the dc= components that identify the domain itself) is always a local operation. If the local domain controller stores a copy of the domain in question, the domain controller can verify the prefix of the distinguished name and perform the requested operation. If the local domain controller does not store a copy of the domain in question, it returns either a referral to another server or an error message that states that the object does not exist.

Components of an LDAP Search

An LDAP search has the potential to retrieve information about all objects within a specific scope that have certain characteristics—for example, the telephone number of every person in a department.

The following are used to accomplish an LDAP search:

- A *search base* (the distinguished name of the search base object) defines the location in the directory from which the LDAP search begins.
- A *search scope* defines how deep to search within the search base.
  - *Base*, or zero level, indicates a search of the base object only.
  - *One level* indicates a search of objects immediately subordinate to the base object, but does not include the base object itself.
  - *Subtree* indicates a search of the base object and the entire subtree of which the base object distinguished name is the topmost object.
- A *filter* allows certain entries in the subtree and excludes others.
- A *selection* indicates what attributes to return from objects that match the filter criteria.
- Optional controls affect how the search is processed.

Figure 3.1 illustrates the base distinguished name and the search scope of an LDAP search.

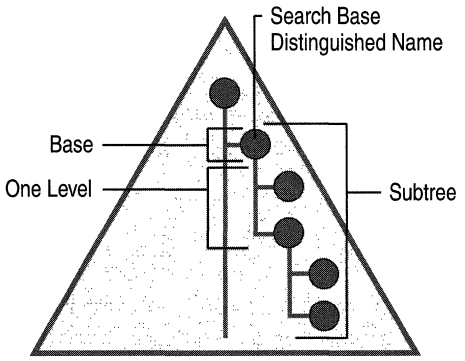


Figure 3.1 LDAP Search Base and Search Scope

Figure 3.2 shows the base distinguished name for a container object.

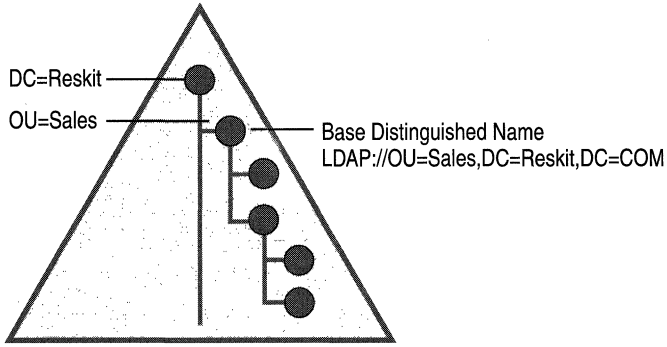


Figure 3.2 Base Distinguished Name for an LDAP Search

Search Filters

By using search filters, you can define search criteria that provide better control to achieve more effective and efficient searches. For example, you might be interested in all the users whose surname is "Smith," or you might want to find out all the team members who report to the manager named "Mary Jones." ADSI supports LDAP search filters as defined in RFC 2254. These search filters are represented by UTF-8 strings. Table 3.1 illustrates some commonly used search filter strings.

Table 3.1 Common LDAP Search Filters

Filter	Description
(objectCategory=*)	All objects.
(&(objectClass=user)(!(cn=susan)))	All user objects except "susan".
(cn=sm*)	All objects with a surname that starts with "sm".
(&(objectClass=contact)(|(sn=Smith)(sn=Johnson)))	All contacts with a surname equal to "Smith" or "Johnson".

The search filters shown in Table 3.1 use one of the following formats:

(<*attribute*><*operator*><*value*>)

–Or–

(<*operator*>(<*filter1*>)(<*filter2*>)...)

Table 3.2 shows some of the most frequently used search filter operators.

Table 3.2 Commonly Used LDAP Search Filter Operators

Operator	Description
=	Equal to
~=	Approximately equal to
<=	Lexicographically less than or equal to
>=	Lexicographically greater than or equal to
&	AND
|	OR
!	NOT

For more information about the LDAP search query syntax and operators, see the Microsoft Platform SDK link and the Request for Comments (RFC) link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. On the Request for Comments (RFC) site, follow the links to RFC 2254.

ObjectCategory vs. ObjectClass in a Search Filter

Because of the existence of the class inheritance hierarchy in the schema, every object in Active Directory is in fact a member of many classes—four or five on the average. For this reason, the *objectClass* index is prohibitively large (for example, 4*n*, where *n* is the number of objects in the system). In addition, *objectClass* has poor selectivity for many possible class values. For example, a search filter of (*objectClass=securityPrincipal*) returns every user and group object in the system.

On the other hand, *objectCategory* usually refers to the most specific class in the object's class hierarchy. Although *objectClass* can have multiple values, the attribute *objectCategory* has only one. Every Active Directory object has an *objectCategory* attribute whose value is a *classSchema* object.

Every *classSchema* object has an attribute called *defaultObjectCategory*, which is the object category of an instance of the class if none is specified by the user. For most classes, the *defaultObjectCategory* value is the class itself. In the search filter, you can specify *objectCategory=X*, where *X* is the *ldapDisplayName* of a class, and LDAP automatically expands the filter to *objectCategory=<defaultObjectCategory* of class *X*>. The *objectCategory* attribute has a syntax of distinguished name, and LDAP automatically converts the value for *objectCategory* to the distinguished name format. For example, if you use *objectCategory*=contact in the filter, the filter changes to *objectCategory*=cn=person,cn=schema,cn=configuration, dc=<*ForestRootDomain*> ("person" is the *defaultObjectCategory* for the class *contact*).

For more information about class inheritance, see "Active Directory Schema" in this book.

LDAP Referrals

When a requested object exists in the directory but is not present on the contacted domain controller, name resolution depends on that domain controller's knowledge of how the directory is partitioned. In a partitioned directory, by definition, the entire directory is not always available on any one domain controller.

An LDAP referral is a domain controller's way of indicating to a client application that it does not have a copy of a requested object (or, more precisely, that it does not hold the section of the directory tree where that object would be, if in fact it exists) and giving the client a location that is more likely to hold the object, which the client uses as the basis for a DNS search for a domain controller. Ideally, referrals always reference a domain controller that indeed holds the object. However, it is possible for the referred-to domain controller to generate yet another referral, although it usually does not take long to discover that the object does not exist and to inform the client. Active Directory returns referrals in accordance with RFC 2251.

In its Configuration container, every domain controller has information about the other domains in the forest. When an operation in Active Directory requires action on objects that might exist in the forest but are not located in the particular domain that is stored on a domain controller, that domain controller must send the client a message that describes where to go to continue this action—that is, the client is "referred" to a domain controller that is presumed to hold the requested object.

Clients do not need to know the name or location of a child domain in order to contact a domain controller in that domain. They can query the root domain and reach the appropriate domain controller by being referred there. Two situations generate this type of domain controller response:

- The base distinguished name of the operation is not in this directory, but the domain controller has knowledge of another LDAP directory where it might be found (an "external referral").
- The base distinguished name of the operation is in this directory, but the operation requires proceeding into portions of the directory tree that are not stored on this domain controller (a subordinate referral).

Every domain controller contains information (called "knowledge") about how the directory is partitioned, and this information can be used with DNS to find the correct Active Directory domain.

Knowledge References

Active Directory stores information about the existence and location of directory partitions, including the names of the directory partitions, the name of the server that is holding read-only copies (partial directory partitions stored on Global Catalog servers), and the name of the server that is holding writable copies (full directory partitions). Active Directory uses this information (known as "knowledge references") to generate referrals to other domain controllers.

Active Directory uses three kinds of knowledge references to generate referrals to other domain controllers:

- A subordinate reference, which is knowledge of a directory partition (or partitions) directly below a directory partition that is held by the domain controller.
- A cross-reference, which is knowledge of one directory partition and which is stored in a cross-reference object. On a specific domain controller, the combination of all cross-references provides knowledge of all directory partitions in the forest, regardless of their locations in the directory tree.

Note The state of cross-reference knowledge at any specific time is subject to the effects of replication latency.

- A superior reference, which is knowledge of a specifically designated referral location that is used when the domain controller has no knowledge of the search base.

Knowledge references form the glue that holds the pieces of the distributed directory together. Because Active Directory is logically partitioned and directory partitions are the discrete components of the directory that replicate between domain controllers, either all objects in a directory partition are present on a particular domain controller or no objects in the directory partition are present on the domain controller. For this reason, references have the effect of linking the partitions together, which allows operations such as searches to span multiple partitions.

In Active Directory, referrals are generated when the client requests that the directory locate an object where, based on the position at which the search begins, no copy exists in a local directory partition. When Active Directory can determine definitively that no such object exists in the directory (rather than that it might exist somewhere else even though no copy exists here), instead of sending a referral, the directory returns an error message to the client that no such object exists in the forest.

For more information about replication of directory partitions, see "Active Directory Replication" in this book.

Subordinate References

When a client requests a search, the domain controller searches all objects at or below the search base, within the directory partition that the domain controller holds. If a subtree search has a search base that includes child partitions, the domain controller uses subordinate references to return referrals (called *subordinate referrals*) to these partitions.

Subordinate referrals are returned as part of the data that is returned from the base distinguished name partition. The referral contains the distinguished name of the subordinate directory partition and the access point to which queries can be referred. An access point consists of a DNS name and a port number, which is the information that is required to contact a specific LDAP server. Access points are generated from information contained in the cross-reference object.

Cross-References

Cross-references are stored as directory objects of the class *crossRef* that identify the existence and location of all directory partitions, irrespective of location in the directory tree. Cross-references enable every domain controller to be aware of all directory partitions in the forest, not only the partitions that it holds. Because these objects are stored in the Configuration container, the knowledge that they store is replicated to every domain controller in the forest.

Values for the following attributes are required for each cross-reference:

- *nCName*. The distinguished name of the directory partition that the *crossRef* object references. ("nC" stands for "naming context," which is a synonym for "directory partition.") The combination of all of the *nCName* properties in the forest defines the entire directory tree, including the subordinate and superior relationships between partitions.
- *dNSRoot*. The DNS name of the domain where servers that store the particular directory partition can be reached. This value can also be a DNS host name.

Cross-reference objects are used to generate referrals to other directory partitions in the forest and to external directories.

Cross-reference objects are created in two ways:

- Internally by the system to refer to known locations that are within the forest.
- Externally by administrators to refer to locations that are external to the forest.

Internal Cross-References

An *internal cross-reference* is an object that is created by the system. For every directory partition in a forest, there is an internal cross-reference object in the Partitions container (cn=Partitions,cn=Configuration,dc=*ForestRootDomain*). When you create a new forest, the Active Directory Installation Wizard creates three directory partitions: the first domain directory partition, the configuration directory partition, and the schema directory partition. For each of these partitions, a cross-reference object is created automatically. Thereafter, when a new domain is created in the forest, another directory partition is created and the respective cross-reference object is created. Because these cross-reference objects are located in the Configuration container, they are replicated to every domain controller in the forest, and thus every domain controller has knowledge of the name of every partition in the forest (as well as their superior and subordinate relationships to each other). By virtue of this knowledge, any domain controller can generate referrals to any other domain in the forest, as well as to the schema and configuration directory partitions.

External Cross-References

An *external cross-reference* is a cross-reference object that can be created manually to provide the location of an object that is not stored in the forest. If your LDAP clients submit operations for an external portion of the global LDAP namespace against servers in your forest, and you want your forest's servers to refer the client to the correct location, you can create a cross-reference object for that directory in the Partitions container.

There are two ways that external cross-references are used:

- To reference external directories by their disjoint directory name (a name that is not contiguous with the name of this directory tree). In this case, when you create the cross-reference, you create a reference to a location that is not a child of any object in the directory.
- To reference external directories by a name that is within the Active Directory namespace (a name that is contiguous with the name of this directory tree). In this case, when you create the cross-reference, you create a reference to a location that is a child of a real directory object.

Note An external directory that is stored on a Windows 2000–based domain controller does not require an explicit cross-reference object. Because the domain component (dc=) portions of the distinguished names of all Windows 2000 domains match their DNS addresses and because DNS is the worldwide namespace, all Windows 2000–based domain controllers can generate external referrals to each other automatically.

Creating External Cross-References

The only time you have to create a cross-reference object is when you want to extend a search to a directory outside the forest that is a non-Windows 2000 LDAP directory service. In this case, you can use an LDAP editor, such as ADSI Edit or Ldp, to create objects of the class *crossRef* in the Partitions container that reference external directories.

Note To use ADSI Edit and Ldp, install the Support Tools that are located in the Support\Tools folder on the Windows 2000 Server operating system CD. To install the tools, double-click the **Setup** icon in that folder. For information about installing and using the Windows 2000 Support Tools and Support Tools Help, see the file Sreadme.doc in the Support\Tools folder of the Windows 2000 operating system CD. For more information about using ADSI Edit and Ldp, see *Microsoft Windows 2000 Resource Kit* Tools Help.

When you create a cross-reference object, you must provide the values for three attributes:

cn The name that describes the directory. For example, for the domain noam.reskit.com, your *cn* value might be "noam" or something else that describes that domain, such as "NorthAmerica."

nCName The distinguished name of the domain directory partition to which your cross-reference refers. If the domain name is noam.reskit.com, the value of *nCName* would be dc=noam,dc=reskit,dc=com.

dnsRoot The DNS host name of an LDAP server in the domain that is identified by *nCName* (for example, server1.noam.reskit.com). The value of *dnsRoot* can also be the domain name if you do not want to specify a server.

Note You must be able to resolve ("ping") the name in *dnsRoot*, which does not necessarily name another Windows 2000–based system; it might be the DNS address of an LDAP server instead of a domain controller. If the directory partition is a Windows 2000 domain from another forest, automatically generated knowledge is usually sufficient and no external cross-reference is required.

You can use either ADSI Edit or Ldp to create cross-reference objects in the Configuration container. However, Ldp requires that you provide the distinguished name of an object and its mandatory and optional attribute names and values when you add the object to Active Directory. For more information about using Ldp, see *Microsoft Windows 2000 Resource Kit* Tools Help.

ADSI Edit provides a convenient graphical user interface for creating cross-reference objects.

▶ **To use ADSI Edit to create a cross-reference object**

1. In ADSI Edit, expand the Configuration container.
2. Right-click the **CN=Partitions** container, click **New**, and then click **Object**.
3. For **Select a class**, you can create objects of only class *crossRef*, which is already selected. Click **Next**.
4. For the *cn* attribute, in the **Value** box, type a name that describes the location, and then click **Next**.
5. For the *nCName* attribute, in the **Value** box, type the distinguished name for the external domain, and then click **Next**.
6. For the *dnsHostname* attribute, in the **Value** box, type a DNS name for the server that hosts the domain directory partition, or type the domain name.
7. When you are sure that your entries are correct, click **Finish**.

To make use of cross-references, clients must be enabled to follow ("chase") referrals that are returned. Windows Address Book chases referrals by default. In Ldp, you can specify **Chase Referrals** in the search options. When you are using ADSI programmatically (for example, by using Active Data Objects [ADO] to search), you must specify whether to chase referrals. For more information about using ADSI programmatically, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Creating an External Cross-Reference for an External Location

To create a cross-reference to an external directory by referencing an external location, you give the *nCName* attribute a value that is the name of the actual external directory. For example, an external LDAP directory might use X.500 naming (such as o=*Organization Name*,c=*Country/Region*), which would be used for the value of the *nCName* attribute. Queries for this directory must specify the external object by name in the search base distinguished name. A request for a referral to such a location might come in the form of an LDAP Uniform Resource Locator (URL) embedded in an e-mail message or from an application that specifically names the directory distinguished name.

Creating an External Cross-Reference for an Internal Location

If you want a subtree search of a portion of your directory to always include an external LDAP directory that is not a Windows 2000 directory service, you can create a cross-reference to the external directory for an internal location. To create an internal location that references an external directory, give the *nCName* attribute of the cross-reference object a value that is an immediate child object of an existing directory object and that also matches the distinguished name of the external directory. Choose the location according to where you want the external directory to be locatable in Active Directory.

This type of cross-reference is especially useful for smoothly integrating dynamic directories. For example, you might use an instant messaging application such as Microsoft® NetMeeting® conferencing software to publish a list of current or planned conference calls. LDAP is an effective protocol for querying such a published list; however, short-lived, highly volatile data is inappropriate for Active Directory storage. Therefore, you might use an in-memory, nonreplicated LDAP server (one that can store volatile data) at an arbitrary point in the namespace. This "volatile" directory service can then be configured to inhabit a name inside your company's Active Directory namespace and be made available to company users through a cross-reference for an internal location in Active Directory. Users can use this location to find the list of conversations by directory tree navigation.

Suppose that your domain name is reskit.com and you have installed your messaging application on a non-Active Directory–aware LDAP server named vds.it.reskit.com. On that server, you would create a directory for your volatile data, such as cn=conversations,dc=reskit,dc=com. Then, on your Active Directory domain controller, you would create a cross-reference object and use the following attribute values:

- cn=conversation server
- nCName=cn=conversations,dc=reskit,dc=com
- dnsRoot=vds.it.reskit.com

When a user performs a subtree search of dc=reskit,dc=com, the client receives results from the Windows 2000–based domain controllers and also a subordinate referral to the volatile directory service server at vdserver.it.reskit.com, which instructs the client to continue the LDAP search from cn=conversations,dc=reskit,dc=com and below on that server.

Superior References

A superior reference is the distinguished name of a directory partition that is stored in the *superiorDNSRoot* attribute on the *crossRef* object for the forest root domain (the first domain created in the forest). A domain controller uses its superior reference to construct a referral only when a search base does not match any directory partition defined by the cross-reference objects. A superior reference contains no directory tree information; it consists of only an access point to which otherwise unanswerable queries can be referred.

By default, *superiorDNSRoot* does not store a value, but the directory uses the "dc=" components of the search base distinguished name to construct the equivalent of a superior referral. You can use the value in the *superiorDNSRoot* attribute to define a location to send all queries that cannot be resolved.

Ambiguous Name Resolution

Ambiguous name resolution (ANR) is the process of searching for a string value in a set of attributes by using one filter of the form (ANR=*string*).

ANR Attribute Set

By default, the following set of attributes is evaluated when you enter an ANR search string in an LDAP filter:

- *givenName* (first name)
- *sn* (surname, or last name)
- *displayName* (the name given the object when it is created)
- *RDN* (the relative distinguished name of the object)
- *legacyExchangeDN* (for enterprises that have upgraded a Microsoft® Exchange installation to a later version of Exchange that is synchronized with Active Directory, the distinguished name of the old Exchange mailbox that corresponds to the user in Active Directory)
- *physicalDeliveryOfficeName* (for example, Building A, Suite 1234)
- *proxyAddresses* (the collection of e-mail addresses over all e-mail address spaces that the Exchange server knows about)

When the (ANR=*string*) filter is encountered, the filter is expanded to include a search of every attribute in the ANR set.

In Active Directory Users and Computers, you can use the **Filter** or **Find** option (on the shortcut menu or on the toolbar), and select the **Custom** and **Advanced** options to enter an ANR filter. Alternatively, you can use an LDAP editor, such as Ldp. From this LDAP client, you can implement filtered LDAP searches and view the LDAP responses. For more information about using Ldp, see "Active Directory Data Storage" in this book, and see *Microsoft® Windows® 2000 Resource Kit* Tools Help.

ANR Matching of an Embedded Space

For the *givenName* and *sn* attributes, if a space is embedded in the string presented in an ANR filter, the string is split at the first such space and each piece of the string is evaluated separately. This feature enables you to search for a user object by providing the first few characters of the first name (*givenName*) and the first few characters of the last name (*sn*). For example, the filter ANR=dar st finds all objects that have a *givenName* attribute value that begins with "dar" and an *sn* attribute value that begins with "st". In this example, the filter would return a user who has a *givenName* attribute of "Darlene" and an *sn* attribute of "Stuart," as well as a user who has a *givenName* attribute of "Darren" and an *sn* attribute of "Strong."

First/Last and Last/First Evaluation

For the attributes *sn* and *givenName*, when a space is embedded in the string presented in an ANR filter, the filter is expanded to find the values in both respective positions. For example, the filter (ANR=dav st) finds both the user David Strong and the user Steven Davis.

Expanded ANR Filter

When an ANR filter is encountered in an LDAP search, the filter is expanded to construct an OR operation on the string for every attribute in the ANR set. For the *sn* and *givenName* attributes, the first/last and last/first matching are also applied.

The ANR filter of the form (anr= *xxx yyy*) is expanded to the following filter:

```
(| (displayName=xxx yyy*)
   (givenName=xxx yyy*)
   (physicalDeliveryOfficeName=xxx yyy*)
   (proxyAddresses=xxx yyy*)...
   (sn=xxx yyy*)
   (& (givenName=xxx*)(sn=yyy*))
   (& (givenName=yyy*)(sn=xxx*)))
```

The last three lines of the expanded filter are the portion of the filter that evaluates first name and last name.

Adding Attributes to the ANR Set

You can add attributes to the default ANR set by setting a flag on the *attributeSchema* object. By using ADSI Edit, connect to the schema and then open the properties on the attribute that you want to add. Set the *searchFlags* attribute value to a value that represents a bitwise OR operation of 4 and 1 to the existing value. The value **4** adds the attribute to the ANR set; the value **1** indexes the attribute.

For information about setting the *searchFlags* attribute, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Suppressing First/Last and Last/First Functionality

By default, Active Directory is configured to expand the ANR filter to evaluate the positions of two portions of a string that contains an embedded space. In the case of the *sn* and *givenName* attributes, the evaluation also includes checking whether the portion of the string that precedes the embedded space comes before or after the portion of the string that follows the space. If it comes before the portion of the string that follows the space, it is first/last functionality; if it comes after the portion of the string that follows the space, it is last/first functionality.

The *DSHeuristics* attribute on the Directory Service object (cn=Directory Service,cn=Windows NT,cn=Services,cn=Configuration,dc=*ForestRootDomain*) contains a string value that governs the use of first/last and last/first functionality in the first two character positions. The default value of *DSHeuristics* is **00**, which indicates that both functions are enabled. (For all positions, "0" means "perform the default behavior.") The first character in the string governs first/last functionality; the second character governs last/first functionality.

You can modify the first two characters of the string to suppress either one or both functionalities as follows:

10 = Suppress first/last functionality. (This can also be written "1" or "10000" because both mean that only the first character's behavior must be nondefault.)

01 = Suppress last/first functionality.

11 = Suppress last/first and first/last functionality.

Anonymous Queries

If you want users outside the Windows 2000 forest to be able to query Active Directory for white pages–type lookups, you can enable anonymous access to specific directory objects. In Active Directory, anonymous access can be enabled by making objects in a specific container or containers available to the Everyone group.

There are three key requirements for enabling anonymous access to Active Directory:

Server configuration. Read access must be granted to the Everyone group for the containers in Active Directory that are going to allow anonymous access.

Client configuration. An LDAP-compatible client, such as Windows Address Book, must be available to search in Active Directory.

Distinguished name format. Active Directory requires that clients use a search base that contains an LDAP distinguished name (called the "base DN"), which includes both the DNS domain name and the specific container to which anonymous access has been granted.

Note For anonymous access to be available for Internet users, anonymous access must be enabled on the Internet Information Services (IIS) Web server.

Using Access Control to Enable Anonymous Access

You can use access control to allow access to certain objects that you want to make available to anonymous users. You can do so by granting read access to the Everyone group for a container object that stores the public objects.

Members of the Everyone Group

In Active Directory, users who log on with authentication are automatically included in the Authenticated Users group. Users who log on without authentication are represented as Anonymous Logon. Users who log on as Guest are included in the Domain Guests group. In all cases, the users are members of the Everyone group. Therefore, providing access to the Everyone group covers all potential anonymous users in addition to all authenticated users.

Note On all computers that run Windows 2000 or Windows NT, there is a built-in Guest account, which does not require a password (the password can be blank) and is meant to be shared by users who do not have personal accounts. These users are, in a sense, anonymous users.In Active Directory, users who log on with authentication are automatically included in the Authenticated Users group. Users who log on without authentication are represented as Anonymous Logon. Users who log on as Guest are included in the Domain Guests group. In all cases, the users are members of the Everyone group. Therefore, providing access to the Everyone group covers all potential anonymous users in addition to all authenticated users.

By default, the Everyone group has read access to the domain object and its properties, but the access is not inherited by child containers. In Active Directory, users who log on with authentication are automatically included in the Authenticated Users group. Users who log on without authentication are represented as Anonymous Logon. Users who log on as Guest are included in the Domain Guests group. In all cases, the users are members of the Everyone group. Therefore, providing access to the Everyone group covers all potential anonymous users in addition to all authenticated users.

Caution Avoid granting anonymous access to the domain-level container at any level other than "this object only," which is the default setting. By using this default setting, you enable all users to read the properties of the domain object itself but not see any of the objects below it in the hierarchy.

Assigning Read Access for Everyone

By granting the Everyone group read access to a specific container of objects, you can enable anonymous access to only that portion of Active Directory. You can use Active Directory Users and Computers to assign access control to a container in which you have placed the user objects that you want to make available for public access. To see the security options, enable **Advanced Features**.

▶ **To enable anonymous access to an Active Directory container**

1. In the Active Directory Users and Computers console, if **Advanced Features** is not enabled, on the **View** menu, click **Advanced Features**.
2. Right-click the container to which you want to provide anonymous access.
3. Click **Properties**, click the **Security** tab, and then click **Advanced**.

   In the **Permission Entries** box, if the Everyone group is not listed, click **Add**. In the **Name** column, click **Everyone**, and then click **OK**.
4. In the **Permission Entry for *ContainerName*** dialog box, click the **Properties** tab.
5. In the **Apply onto** list, click **User objects**.
6. In the **Permissions** list, in the **Allow** column, click the permission or permissions that you want to allow (for example, **Read General Information**). Then click **OK**.
7. On every security warning message that appears, if any, click **Yes**.
8. In the **Access Control Settings** dialog box, click **OK**.

Caution Enabling anonymous queries weakens the inherent security in an Active Directory environment. Special care should be taken when you are deciding what containers and attributes are to be exposed to anonymous users.

Granting Read All Properties for Anonymous Queries

The built-in group Pre-Windows 2000 Compatible Access has Read All Properties access on user and group objects. By default, the Everyone group is not a member of this group. If you want to grant this level of access to anonymous users, you can add Everyone to this group. By doing so, you allow anonymous read access to all properties of all user and group objects. For more information about access control, see "Access Control" in this book.

Security Precautions for Anonymous Access

Any time that anonymous access is enabled where Internet access is available, it is critical to domain security that firewalls be configured to protect the ports that are used to gain entry to Active Directory.

A firewall is a combination of hardware and software that provides a security system, usually to prevent unauthorized access from the Internet to an internal network. A firewall prevents direct communication between network and external computers by routing communication through a proxy server outside the network. The proxy server determines whether it is safe to let a file pass through to the network.

Firewalls should be configured to protect the following ports:

- Port 389 for LDAP
- Port 636 for LDAP over Secure Sockets Layer (SSL)
- Port 3268 for the Global Catalog
- Port 3269 for the Global Catalog over SSL

For more information about configuring firewalls, see "Internet Protocol Security" in the *TCP/IP Core Networking Guide*.

Global Catalog and LDAP Searches

The Global Catalog enables searching for Active Directory objects in any domain in the forest without the need for subordinate referrals, and users can find objects of interest quickly without having to know what domain holds the object.

Global Catalog Servers

A Global Catalog server is a domain controller that stores extra information; its database stores rows for every object in the forest instead of rows for only the objects in one domain. The rows that store objects that occur in domain directory partitions other than the local domain partition hold only a subset of attributes for each object. In this way, the Global Catalog enables forest-wide searches without requiring replication of the entire contents of Active Directory to every domain controller. The Knowledge Consistency Checker (KCC) process creates a replication topology that ensures delivery of the contents of every directory partition to every Global Catalog server in the forest.

Note A Global Catalog server stores full (writable) copies of the schema and configuration directory partitions—the same as any domain controller.

By default, the server on which you install Active Directory to create the first domain in a new forest is a Global Catalog server. Thereafter, you must designate additional Global Catalog servers, if they are needed.

Searching the Global Catalog vs. Searching the Domain

The decision whether to search the Global Catalog or the domain is based on the scope of the search:

- When the scope of a search is the domain or an organizational unit, the query can be resolved within the domain partition by using an LDAP search.
- When the scope of a search is the forest, the query can be resolved within any partition by using a Global Catalog search.

Searches That Use the Global Catalog by Default

Any time that you specify port 3268, you are searching in the Global Catalog. In addition, the Global Catalog is searched by default under the following conditions:

- During the logon process when a user principal name is presented. The Global Catalog is searched to find the domain and account name on the basis of the user principal name.

- During the logon process to expand universal groups. Universal group membership can span domains. It is possible, therefore, that a user has a membership in a universal group that is not in the logon domain. For this reason, the Global Catalog is contacted to search the membership of universal groups. If a membership is found, the group is attached to the user's logon credentials.
- When you choose **Entire Directory** in a search-scope list.
- When you write the value for a distinguished name-valued property, where the distinguished name represents a nonlocal object. For example, if the member that you are adding is from a different domain, the Global Catalog is used to verify that the user object represented by the distinguished name actually exists.

Global Catalog Search Base

For an LDAP search, you must supply a valid base distinguished name. For a Global Catalog search, the base distinguished name can be any value, including the value "NULL" (" "). A base distinguished name of NULL effectively scopes the search on the search computer to the Global Catalog. If you use a NULL base distinguished name with a scope of one level or subtree and specify port 389 (the default LDAP port), the search fails. Therefore, if you submit a NULL search to the Global Catalog port and then change the port to the LDAP port, you must change the base distinguished name for the search to succeed.

Note Windows Address Book is configured automatically with the value "NULL" for server name, account name, and base distinguished name. The default port is port 3268; so to submit the search to port 389, you must provide a valid base distinguished name as defined in RFC 2247. A blank base distinguished name fails on either port.

Characteristics of a Global Catalog Search

The following characteristics differentiate a Global Catalog search from a standard LDAP search:

- Global Catalog queries are directed to port 3268, which explicitly indicates that Global Catalog semantics are required. By default, ordinary LDAP searches are received through port 389. If you bind to port 389, even if you bind to a Global Catalog server, your search includes a single domain directory partition. If you bind to port 3268, your search includes all directory partitions in the forest. If the server you attempt to bind to over port 3268 is not a Global Catalog server, the server refuses the bind.

- Global Catalog searches can specify a non-instantiated search base, indicated as "com" or " " (blank search base).
- Global Catalog searches cross directory partition boundaries. The extent of the LDAP search is the directory partition.
- Global Catalog searches do not return subordinate referrals. If you use port 3268 to request an attribute that is not in the Global Catalog, you do not receive a referral to it. Subordinate referrals are an LDAP response; when you query over port 3268, you receive Global Catalog responses, which are based solely on the contents of the Global Catalog. If you query the same server by using port 389, you receive referrals for objects that are in the forest but whose attributes are not referenced in the Global Catalog.

Note An external referral can be returned by the Global Catalog if a base-level search for an external directory is submitted and if the distinguished name of the external directory uses the domain component (dc=) naming attribute. This referral is returned according to the ability of Active Directory to construct a DNS name from the domain components of the distinguished name and not based on the presence of any cross-reference object. The same referral is returned by using the LDAP port; it is not specific to the Global Catalog. (For more information about constructing a DNS name from the domain components, see "Superior References" earlier in this chapter.)

Effect of Global Catalog When Searching Back Links and Forward Links

Some Active Directory attributes cannot be located specifically by finding a row in the directory database. A back link is an attribute that can be computed only by referencing another attribute, called a forward link. An example of a back-link attribute is the *memberOf* attribute on a user object, which relies on the group attribute *members* to derive its values. For example, if you request the groups of which a specific user is a member, the forward link *members*, an attribute of the group object, is searched to find values that match the user name that you specified.

Because of the way that groups are enumerated by the Global Catalog, the results of a back-link search can vary, depending on whether you search the Global Catalog (port 3268) or the domain (port 389), the kind of groups the user belongs to (global groups vs. domain local groups), and whether the user belongs to groups outside the local domain. Connecting to the local domain does not locate the user's group membership in groups outside the domain. Connecting to the Global Catalog locates the user's membership in global groups but not in domain local groups because local groups are not replicated to the Global Catalog. For more information about searching on back-link attributes, see "Active Directory Data Storage" in this book.

Searching for Deleted Objects

When an Active Directory object is deleted, it is stored in the Deleted Objects container for a configurable period of time to allow replication of the deletion to occur. By using the **Show Deleted Object** control (controlType = 1.2.840.113556.1.4.417), in conjunction with search commands, you can view Active Directory objects that have been deleted but not yet garbage collected. These objects are called *tombstones*. After they are deleted by garbage collection, they no longer exist in the directory database.

To retrieve tombstone objects, list the contents of the Deleted Objects container. You can use Ldp to find these objects by using an LDAP control.

▶ **To use Ldp to search the domain for deleted objects (tombstones)**

1. On the **Start** menu, click **Run**, and then type **ldp**.
2. Connect and bind to a domain controller in the domain whose tombstones you want to retrieve.
   - To connect, on the **Connection** menu, click **Connect**, and then type a server name and a port number.
   - To bind, on the **Connection** menu, click **Bind**, and then type an account name, password, and domain if you want to connect to a domain other than the domain to which you are currently logged on.
3. On the **Browse** menu, click **Search**.
4. In the **Search** dialog box, for **Base DN**, type the distinguished name of the domain whose tombstones you want to retrieve.
5. In the **Filter** box, use the filter **(isDeleted=*)**.
6. Under **Scope**, click **Subtree**.
7. Click **Options**.
8. In the **Search Options** dialog box, under **Search Call Type**, click **Extended**.
9. Click **Controls**. Then in the **Object Identifier** box, type the following:

   1.2.840.113556.1.4.417
10. Under **Control Type**, click **Server**.
11. To add the control to the **Active Controls** list, click **Check in**. Then click **OK**.
12. In the **Search Options** dialog box, click **OK**.
13. In the **Search** dialog box, click **Run**.

For more information about how to use Ldp, see *Microsoft Windows 2000 Resource Kit* Tools Help. For more information about using Ldp for directory management and troubleshooting tasks, see "Active Directory Diagnostics, Troubleshooting, and Recovery" in this book.

LDAP Search Clients

Several clients that are available with Windows 2000 Server provide varying degrees of sophistication for searching Active Directory.

Administrative Clients

Administrative clients such as the Active Directory Users and Computers MMC snap-in provide search and filter options when certain objects are selected. In addition, when you open Network Places, Entire Network, or Directory, the **Find** option that is available provides the same search capabilities as the **Find** option in Active Directory Users and Computers.

Using the Filter Options Command in Active Directory Users and Computers

In Active Directory Users and Computers, you can use filter options to define the information that you want to view. When you apply a filter, only the objects you specify in the filter are displayed in the filtered container. The default filtering option displays all types of objects (that is, no filter is applied). However, it is possible to select only certain types of objects to be displayed, such as users, groups, contacts, and so on. Also, you can customize the kind of information that is displayed within each object type by selecting fields and specifying a condition and value, or by entering an LDAP query. The filter remains in effect until you remove it. It is not displaced or overridden by any other filter.

Active Directory Users and Computers provides options for filtering that do not require you to create an LDAP query. (For more information about using standard filter options, see Windows 2000 Server Help.)

However, if the options in the standard filter user interface do not meet your needs, you can use advanced (customized) filter options to write an LDAP query that does. For example, the standard filter user interface allows you to select "users" as the objects to filter, but it does not provide the ability to specify all possible attributes for a user. For example, if you want to display only the user accounts that were created after a specific date, you can use an LDAP filter to retrieve only these users by using the *whenCreated* attribute value in an LDAP filter.

▶ **To use Filter Options to apply a filter to a container by using an LDAP query**

1. In Active Directory Users and Computers, in the console tree pane, click the container for which you want to filter objects.
2. On the **View** menu, click **Filter Options**.
3. In the **Filter Options** dialog box, click **Create custom filter**, and then click **Customize**.

4. Click the **Advanced** tab.
5. In the **Enter LDAP query** box, type an LDAP query string, for example:
   (&(objectCategory=user)(whenCreated=991122000000Z))

   Note The time format YYMMDDHHMMSSZ must be used to represent the two-digit year (YY), month (MM), day (DD), hour (HH), minutes (MM), and seconds (SS) and must end with an uppercase "Z". You can use zeros to fill in the time elements if you are not interested in the time of creation.

6. Click **OK** twice. Double-click the container to view the filtered objects.

A filter remains in place until you remove it. You can remove a filter in Active Directory Users and Computers by clicking the **Filter** icon on the toolbar and then, in the **Filter Options** dialog box, click **Show all types of objects**.

Using the Find Command in Active Directory Users and Computers

You also can use an LDAP query to search a container without applying a filter to the container. To create an LDAP query to display only specific objects in a container, use the **Find** option on the container shortcut menu, as described in the following procedure.

▶ **To use Find to search a container by using an LDAP query**

1. In Active Directory Users and Computers, right-click the container you want to search, and then click **Find**.
2. In the **Find** box, click **Custom search**, and then click the **Advanced** tab.
3. In the **Enter LDAP query** box, type an LDAP query string, for example:
   (&(objectCategory=user)(whenCreated=991122000000Z))
4. Click **Find Now** to display the search results.

Note In advanced filters, you can use matching rules to implement search flags if you know the correct LDAP control object identifier (also known as an "OID") value to use and how to compute the value. For example, you can search on the *userAccountControl* attribute to specify users who have disabled accounts, or you can search on the *groupType* attribute to find all the Global groups in a search base. For more information about using search flags and matching rules, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Windows Address Book

Windows Address Book is a generic LDAP search client that is designed to work with any LDAP server. Address Book is integrated into the Windows 2000, Microsoft® Internet Explorer version 4.0 and later, and Windows 95 and Windows 98 shells to provide the capability to search for people in one or more directory services, including Active Directory. Address Book version 5.0 is included with Windows 2000 and Microsoft® Internet Explorer 5, and provides a set of accounts that are preconfigured to enable easy access to information in several Internet "white pages" directories, such as InfoSpace and VeriSign.

Address Book Access to Active Directory

In Windows 2000, Address Book provides access to Active Directory as follows:

- It is automatically configured to search the Global Catalog of the forest to which the user is bound when the user selects **Search** and then **Find People** on the **Start** menu.
- It supports UTF-8 to expose Unicode characters, which enables customers to search for users and resources whose names contain non-ASCII characters. This feature is important in European and Asian countries.
- By using the property access control lists (ACLs) on an object, it can display an edit box if the user has permissions to modify the property.
- It provides flexible matching by using support for ambiguous name resolution.
- It can gain access to properties of objects that are created from extended object classes.
- It exposes an API for processing LDAP URLs. Address Book is included with Internet Explorer and registers itself as an LDAP URL (ldap://) handler.
- It supports chasing LDAP referrals (RFC 2251) when the Address Book searches in Active Directory over port 389.

By default, a server name and account name of NULL are configured for Address Book. Active Directory dynamically provides the server name that is cached during domain controller location for the server name, and it uses the logon name of the authenticated user as the account name. The Active Directory properties in Address Book show the default settings.

▶ **To view Active Directory properties in Address Book**

1. On the **Start** menu, point to **Programs**, point to **Accessories**, and then click **Address Book**.
2. On the **Tools** menu, click **Accounts**.
3. Click **Active Directory**, and then click **Properties**. The property sheet displays the directory service account settings, as shown in Figure 3.3.

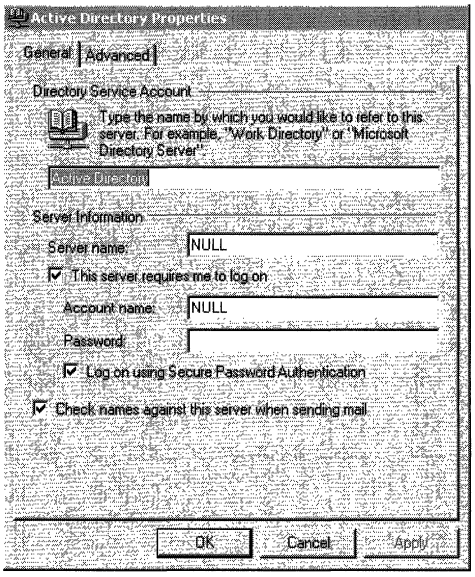


Figure 3.3 Server Settings in the Active Directory Properties Dialog Box in Address Book

Search Base in Address Book Searches

As an LDAP directory, Active Directory requires an RFC 2247–compliant distinguished name, or search base, to perform an LDAP search. By default, a search base distinguished name of NULL is configured in Address Book. During domain controller location, the Locator caches the DNS name of the found domain controller. When requesting a search of Active Directory, Address Book uses as the search base the cached distinguished name of the domain in which the logon account was authenticated.

The **Advanced** tab in the **Active Directory Properties** dialog box displays the settings that determine how Address Book searches are performed, as shown in Figure 3.4.

Note By default, the port setting identifies the Global Catalog port 3268. Clicking **Use Default** changes the port to the default LDAP port 389.

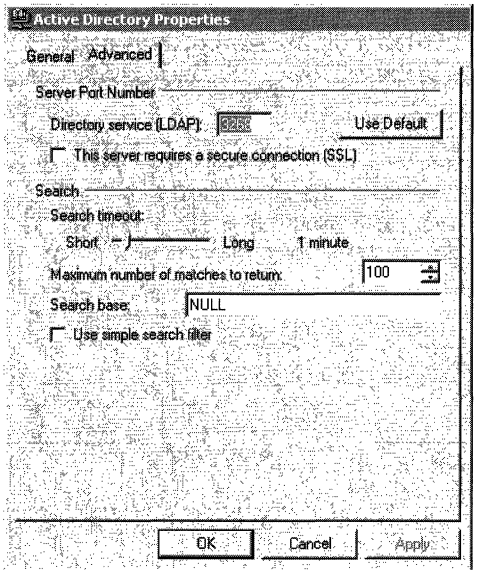


Figure 3.4 Advanced Search Settings in the Active Directory Properties Dialog Box in Address Book

Active Directory Availability on Windows 98 and Windows NT 4.0 Clients

Computers that are running Windows 98 or Windows NT 4.0 that have Internet Explorer 5 installed are not able to gain access to Active Directory unless the clients are configured with a server name and search base and, if the server requires an authenticated logon, an account name (for example, *domainName\userName*). These values must be entered in the **General** and **Advanced** tabs to define Active Directory as an LDAP server for Address Book.

For instructions about how to change the Address Book settings, see Address Book Help. For more information about domain controller location, see "Locating Active Directory Servers" earlier in this chapter.

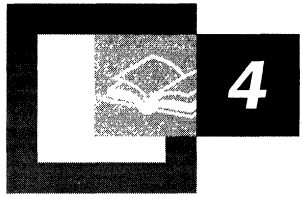
Ldp

Ldp is a tool that you can use to search in Active Directory by using LDAP filters. You also can use Ldp to add, delete, and modify objects in Active Directory and to perform extended LDAP operations by using LDAP controls. To use Ldp, install the Support Tools that are located in the Support\Tools folder on the Windows 2000 Server operating system CD. To install the tools, double-click the **Setup** icon in that folder. For information about installing and using the Windows 2000 Support Tools and Support Tools Help, see the file Sreadme.doc in the Support\Tools folder of the Windows 2000 operating system CD. You can run the Ldp tool from the **Start/Run** menu, or from the command line by typing **ldp**.

For more information about how to use Ldp, see Ldp Help in *Microsoft Windows 2000 Resource Kit* Tools Help. For more information about using Ldp for directory management and troubleshooting tasks, see "Active Directory Diagnostics, Troubleshooting, and Recovery" in this book.

CHAPTER 4

Active Directory Schema



Information about the schema for Active Directory™, the directory service that is included with Microsoft® Windows® 2000 Server, is essential for anyone who is managing the directory service or who is developing directory-aware applications for Microsoft® Windows® 2000–based servers. The material in this chapter covers how the schema for Active Directory is defined, how to modify ("extend") the schema, what safeguards are implemented in the service to protect the schema from being corrupted during the process of modifying it, what automatic checks are included to ensure schema consistency, and what precautions to take when you install new applications on a domain controller.

In This Chapter

Related Information in the Resource Kit

- For an overview of the Active Directory physical structure, see "Active Directory Data Storage" in this book.
- For more information about Active Directory tools, see "Active Directory Diagnostics, Troubleshooting, and Recovery" in this book.
- For information about service publication, see "Service Publication in Active Directory" in this book.

Introduction to the Active Directory Schema

In Active Directory, the schema contains definitions for the universe of objects that can be stored in the directory, and it enforces the rules that govern both the structure and the content of the directory. The schema consists of a set of classes, attributes, and syntaxes that represent an instance of one or more classes in the schema. A *class* is a category of objects that share a set of common characteristics. It is a formal description of a discrete, identifiable type of object that can be stored in the directory. Each object in the directory is an instance of one or more classes in the schema. An *attribute* describes the characteristics of some aspect of an object. Attributes define the types of information that an object can hold. For each class, the schema specifies the mandatory attributes and optional attributes that constitute the set of shared characteristics of the class. The values assigned to attributes define specific characteristics. A *syntax* is the data type of a particular attribute. Syntaxes determine what data type an attribute can have. Active Directory uses a set of standard syntaxes. The predefined syntaxes do not actually appear in the directory, and you cannot add new syntaxes. An everyday example of an object is a vehicle, which can belong to the class of trucks, the class of motorcycles, or the class of cars, and so forth. A car can be described by its make, model, and color. These are some of the attributes of the car. In the example of the car, the possible values for the color of the car might be red, blue, or gray. The syntax for color might be the nomenclature (such as 2B1R2Y) that denotes specific combinations of primary colors that comprise what one sees as the colors of automotive paints.

The schema specifies the relationships between classes of objects. Each object stored in the directory is an instance of one or more classes in the schema. *User*, *Computer*, and *printQueue* are examples of classes in Active Directory. For example, if the schema contains a class called *User*, the user accounts, Sue and Mary, are two objects in the directory that are instances of the class *User*. The object Mary might contain an optional attribute defined for this class called *phoneNumber*. This attribute for the object Mary of the class *User* might have the value 555-0100.

For example, the attribute *phoneNumber* can be defined to take values of the syntax String(numeric), which means that the value can contain only the digits 0 through 9.

The base schema that ships in Microsoft Windows 2000 contains all of the class and attribute definitions that are used by Windows 2000 and Windows 2000 components.

The schema itself is represented in Active Directory by a set of objects known as "schema objects." For each class in the schema, there is a schema object that defines the class. This object is called a *classSchema* object. For each attribute in the schema, there is also a schema object that defines the attribute. This object 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. Storing the schema in the directory has many advantages. One example is that when user applications locate the schema in the directory, they can read the schema to discover what types of objects and properties are available.

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 objects in Active Directory. Applications that are "directory-enabled" are programmed to recognize the attributes and syntaxes that are required to interact with the directory.

Location of the Schema in Active Directory

The objects stored in Active Directory are arranged in a logical hierarchy called the *Directory Information Tree (DIT)*. Active Directory includes a preconfigured database (commonly referred to as the *base DIT*) that contains the information that is required to install and run Windows 2000 and Active Directory. The base DIT is 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. A *directory partition* is a tree of directory objects that forms a unit of replication in Active Directory.

Schema objects are located in the Schema container. The Schema container is not a container in the sense of a special type of Active Directory object that contains other objects; the Schema container is a special purpose object class. The Schema container (cn=schema,cn=configuration,dc=< forest root *domainName*>) contains all of the class and attribute definitions that are required to locate objects in Active Directory and to create new objects. It is the topmost object of the schema directory partition.

The relationship of the schema partition and the Configuration and Schema containers is illustrated in Figure 4.1.

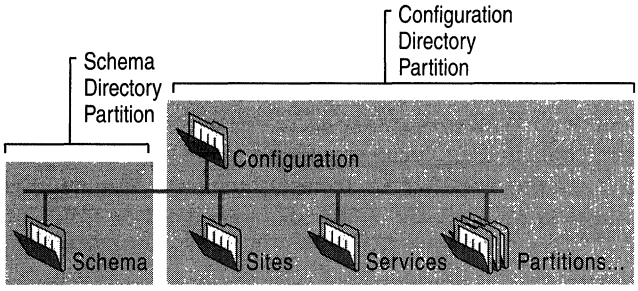


Figure 4.1 Location of the Schema Container

Note The schema is a directory partition in its own right to prevent potential dependency problems that can arise because of the way directory objects are replicated. For more information about the schema directory partition and why the schema is a separate directory partition, see "Name Resolution in Active Directory" in this book.

Finding the Schema Container

Every Active Directory object can be referenced by a unique and unambiguous name known as the *distinguished name* (also known as a "DN"). The distinguished name identifies the *domain* that holds the object as well as the complete path through the container hierarchy by which the object is reached. 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, see "Active Directory Logical Structure" in this book.

You can view the contents of the Schema container by using the Active Directory Schema console in Microsoft Management Console (MMC). 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 that is provided with Windows 2000 Server. To use ADSI Edit and Ldp, install the Support Tools that are located in the Support\Tools folder on the Windows 2000 Server operating system CD. To install the tools, double-click the **Setup** icon in that folder. For more information about using ADSI Edit and Ldp, see Microsoft® Windows® 2000 Support Tools Help. For information about installing and using the Windows 2000 Support Tools and Support Tools Help, see the file Sreadme.doc in the Support\Tools folder of the Windows 2000 operating system CD. For information about diagnosing and troubleshooting problems using the Ldp tool, see "Active Directory Diagnostics, Troubleshooting, and Recovery" in this book.)

It is possible to locate the Schema container without knowing the domain name. Installation scripts and other applications that might not know what domain they are to be used in are able to gain access to the schema because they bind to a special entry at the top of the logical namespace called rootDSE, which provides the schema location. The *rootDSE* (DSA-specific Entry) represents the top of the logical namespace and, therefore, the top of the Lightweight Directory Access Protocol (LDAP) search tree. The attributes of rootDSE identify, among other things, the directory partitions—that is, the domain, schema, and configuration directory partitions—as well as the forest root domain directory partition. One attribute, *schemaNamingContext*, provides the location of the schema so that applications that are connecting to any domain controller can find and read the schema. (For more information about the rootDSE, see "Name Resolution in Active Directory" in this book.)

▶ **To identify the Schema directory partition by using ADSI Edit**

1. Start the ADSI Edit console in MMC.
2. Right-click **ADSI Edit**, and then click **Connect to**.

   The **Connection dialog box** is displayed.
3. In the **Connection Point** check box, make sure **Naming Context** is selected.
4. Select **RootDSE** from the **Naming Context** box, and then click **OK**.
5. In the Console Tree, double-click **My Connection**.

   The **RootDSE folder** is displayed.

6. Right-click the **RootDSE** folder, and then click **Properties**.
7. In the **Select property to view** dialog box, select **schemaNamingContext** from the list of properties ("attributes").
8. In **Attribute Values**, view the **Value(s)** box to see the distinguished name of the schema directory partition.

Note The Schema Management snap-in is not one of the default MMC snap-ins that is provided with Windows 2000 Server. To make it appear in the list of available snap-ins, you must install the admin tools package (Adminpak.msi). To register the Schema Management snap-in, open your %SystemRoot%\System32 folder and run **Regsvr32** Schmmgmt.dll from the command prompt or from the **Run** command on the **Start** menu.

Subschema Subentry

The rootDSE also carries a mandatory attribute called the *subSchemaSubEntry*. Its value is the distinguished name of a *subSchema* object in the directory in which the server makes available the attributes (in *attributeTypes*) and classes (in *objectClasses*) of which the Active Directory schema is comprised. This special object, an instance of the unique *subSchema* class, is used for administering information about the schema, in particular the object classes and attribute types that are supported. This enables client applications to retrieve the information by querying the *subSchema* entry. Clients must only retrieve attributes from a *subSchema* entry by requesting a base object search of the entry, where the LDAP search filter is "(*objectClass=subSchema*)." The location of the *subSchemaSubEntry* container is as follows:

```
CN=Aggregate,CN=Schema,CN=Configuration,DC=<DomainName>,DC=<DomainRoot>
```

Schema Files

Active Directory data is distributed among all domain controllers in the forest. No single domain controller stores all Active Directory data for the entire forest, but every domain controller does hold a copy of the schema. The Active Directory data that is in use on a particular domain controller is stored in a file named Ntds.dit. The location of the Ntds.dit file is an option that is set during the promotion process while you create the directory. The default location for the database and database log files is %SystemRoot%\Ntds. (For more information about the Ntds.dit file, see "Active Directory Data Storage" in this book.)

Another file, the Schema.ini initialization file contains the information that is necessary for creating the default directory objects and the default security for the DIT, as well as the Active Directory display specifiers. For information about display specifiers, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. Although this file is named Schema.ini, the schema itself is actually preloaded and is contained in the base version of Ntds.dit that is installed by the Active Directory Installation wizard.

Active Directory Schema Objects

The attributes and classes in Active Directory are stored in the Schema container as directory objects called *schema objects*. The Schema container itself is represented in Active Directory by an object that is an instance of the Directory Management Domain (*dMD*) class.

For more information about Active Directory *attributeSchema* and *classSchema* objects, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

attributeSchema Objects

Attributes are data items that are used to describe the classes that are defined in the schema. They are defined in the schema separately from the classes, which allows a single attribute definition to be applied to many classes.

Attributes are *attributeSchema* objects. Each *attributeSchema* object is an instance of the *attributeSchema* class. The *attributeSchema* object lists, among other things, the following information:

- The LDAP display name of the attribute.
- The object identifier for the attribute.
- The globally unique identifier (GUID) for the attribute.
- The syntax of the attribute.
- The range for the attribute. For integers, range defines the minimum and maximum value; for strings, range defines the minimum and maximum length.
- Whether the attribute is a multivalue attribute. Note that multivalue attributes hold a set of values with no particular order. There is no guarantee that multivalue attributes are ever going to be returned in the order in which they were stored (or in any other order).
- Whether and how the attribute is indexed.

Single-Value or Multivalue Attributes

Attributes might be single-value or multivalue. Single-value and multivalue attributes are defined by the *singleValued* attribute being set to TRUE or FALSE. The Active Directory Schema console reports this as "single-valued" or "multivalued" rather than as an attribute-value pair.

A multivalue attribute can contain multiple values, all of uniform syntax. Note that multivalue attributes hold a set of values with no particular order. There is no guarantee that multivalue properties are ever going to be returned in the order in which they were stored (or any other order).

Note The LDAP protocol reads a multivalue attribute as a single entity. This can be inconvenient or even impossible when the number of values in a multivalue attribute becomes large. An Internet draft titled "Incremental Retrieval of Multivalued Properties" defines an option called Range that can be specified as part of an attribute description to retrieve the values of a multivalue attribute incrementally. Servers might or might not honor the range option. Servers that support the range option include the object identifier 1.2.840.113556.1.4.802 in the *supportedControls* operational attribute on the rootDSE. Clients must not use the range option unless this object identifier is present. The range option is a constant, case-insensitive string value (Range=), followed by a range-specifier that lists the initial and terminal values in the range.

For more information about the retrieval of multivalue attributes, see the Internet Engineering Task Force (IETF) link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. Follow the links to Internet Drafts, and then use a keyword search.

Indexed Attributes

Making an attribute indexed means that directory searches involving that attribute are going to be more efficient than if the attribute had no index. Attributes are indexed when the least significant bit in their *searchFlags* attribute is set to the value 1. Changing the value of the bit to 1 dynamically builds an index; changing the value to 0 or deleting it drops an index for the attribute in question. The index is built automatically by a background thread on the directory server.

Ideally, indexed attributes are single value with highly unique values that are evenly distributed across the set of instances. Multivalue attributes can be indexed, but the cost to build the index is larger in terms of storage and updating. Even with single-value attributes, keep in mind that the more indexed attributes a class has, the longer it takes to modify or create instances of the class.

Attributes for *attributeSchema* Class Objects

Attributes for the *attributeSchema* class are described in Table 4.1.

Table 4.1 Attributes for the *attributeSchema* Class

Attribute	Syntax	Mandatory	Multi-value	Description
cn	Unicode	Yes	No	Descriptive relative distinguished name for the schema object.
attributeID	Object identifier	Yes	No	Object identifier that uniquely identifies this attribute.
lDAPDisplayName	Unicode	Yes, but filled in automatically	No	Name by which LDAP clients identify this attribute.
schemaIDGUID	String(Octet)	Yes	No	GUID that uniquely identifies this attribute.
mAPIID	Integer	No	No	Integer by which Messaging Application Programming Interface (MAPI) clients identify this attribute.
attributeSecurityGUID	GUID	No	No	GUID by which the security system identifies the property set of this attribute.
attributeSyntax	Object identifier	Yes	No	Syntax object identifier of this attribute.
oMSyntax	Integer	Yes	No	Syntax of this attribute as defined by the XAPIA X/Open Object Model (XOM) specification.
isSingleValued	BOOL	Yes	No	Indicates whether this attribute is a single-value or multivalue attribute. Note that multivalue attributes hold a set of values with no particular order. There is no guarantee that multivalue attributes are ever going to be returned in the order in which they were stored (or in any other order).
extendedCharsAllowed	BOOL	No	No	Indicates whether extended characters are allowed in the value of this attribute. Only applies to attributes of syntax String(teletex).
rangeLower	Integer	No	No	Lower range of values that are allowed for this attribute.[2]

(continued)

Table 4.1 Attributes for the *attributeSchema* Class *(continued)*

Attribute	Syntax	Mandatory	Multi-value	Description
rangeUpper	Integer	No	No	Upper range of values that are allowed for this attribute.[2]
systemFlags	Integer	No	No	Flags that determine specific system operations. Note: this attribute cannot be set or modified. The systemFlags that are relevant to the schema objects are the following: Attribute is required to be a member of the partial set = 0x00000002. Attribute is not replicated = 0x00000001. Attribute is a constructed attribute = 0x00000004.
searchFlags	Integer	No	No	The searchFlags property of each property's *attributeSchema* object defines whether a property is indexed. 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 (should be used in conjunction with 1); 8 = Preserve this attribute on logical deletion (that is, make this attribute available on tombstones).
isMemberof PartialAttributeSet	BOOL	No	No	A Boolean value that defines whether the attribute is replicated to the global catalog (if replicated to the global catalog, it has a value of TRUE, if not, its value is FALSE). For more information, see "Active Directory Replication" in this book.
SystemOnly	BOOL	No	No	System-only attributes are those attributes on which Windows 2000 and Active Directory depend for normal operations. If TRUE, only the system can modify this attribute. No user-defined attribute must ever have the *systemOnly* flag set. *(continued)*

Table 4.1 Attributes for the *attributeSchema* Class *(continued)*

Attribute	Syntax	Mandatory	Multi-value	Description
objectClass	Object identifier	Yes	Yes	Class of this object, which is always *attributeSchema*.
nTSecurityDescriptor	NT-Sec-Des	Yes	No	Security descriptor on the *attributeSchema* object itself.
oMObjectClass	String(Octet)	No	No	For object-syntaxed attributes (OM-syntax = 127), the Basic Encoding Rules (BER) encoded object identifier of the XOM object class. For more information about BER encoding, see RFC 2251.
LinkID	Integer	No	No	Whether a linked attribute or not, an even integer denotes a forward link, an odd integer a back link. A forward link is a pointer to another object in the directory; a back link points back to the first object that has a forward link to it. (For more information about links, see "Active Directory Data Storage" in this book.)

[1]. *Unicode* is a 16-bit character set that contains all of the characters commonly used in information processing.

[2]. When *rangeLower* and *rangeUpper* are defined for attributes that are integers, they define the limits of the value held by the attribute. When they are defined for attributes that are strings, they define the number of characters that can be held in the string.

classSchema Objects

The *classSchema* object specifies the various attributes of the class with which it is associated and, among other things, defines the following constraints of objects that are instances of the class:

- The list of mandatory attributes that must be present on any object that is an instance of this class.
- The list of optional attributes that, in addition to the *mustContain* attributes, can and might be found on an object that is an instance of this class.
- Hierarchy rules that determine the possible parents in the Directory Information Tree of an object that is an instance of the class.

An object can have only attributes that belong to either the *mustContain* or the *mayContain* list for the class.

The *classSchema* object is essentially a template that contains the "rules" for creating objects in an Active Directory class. When a new object is created in a class, the *classSchema* object ensures that this new object has the same properties ("attributes") as all other objects in the class. After an object has been created, the object's class can never be changed.

The *classSchema* object contains, among other things, the following information:

- The LDAP display name of the class.
- The object identifier for the class.
- The GUID for the class.
- The attributes that must be present for an instance of the class.
- Other attributes that can be present for an instance of the class.
- The classes to which the parent of instances of this class may belong.
- The superclass from which this class inherits characteristics.
- Other Auxiliary classes from which this class inherits attributes.
- The type of class (Abstract, Structural, or Auxiliary).
- The default hiding state for the class. If you do not want instances of your class displayed by the end-user user interface, you can define the class as hidden by default.

Categories of Object Classes

The X.500 1993 specification requires that object classes be assigned to one of four categories:

- Structural
- Abstract
- Auxiliary
- 88

Different categories of classes allow for defining structure in the directory. The four categories of classes are applied as follows:

Structural Classes Structural classes are the only classes that can have instances in the directory. That is, you can create directory objects whose class is one of the Structural classes. A Structural class can be used in defining the structure of the directory and is derived from either an Abstract class or another Structural class. A Structural class can include any number of Auxiliary classes in its definition. This type of class is specified by a value of 1 in the *objectClassCategory* attribute.

Abstract Classes Abstract classes are templates that are used only to derive new Structural classes. Abstract classes cannot be instantiated in the directory. This means that no object can belong only to an Abstract class; each object of an Abstract class also belongs to some nonabstract subclass of that class. A new Abstract class can be derived from an existing Abstract class. This type of class is specified by a value of 2 in the *objectClassCategory* attribute. Classes of the abstract category have the sole function of providing attributes for subordinate classes, called subclasses. A subclass contains all mandatory and optional attributes of the class from which it is derived, called its superclass, in addition to those specific to the class itself. Likewise, the subclass of that class contains all attributes of both superclasses, and so forth.

Auxiliary Classes Auxiliary classes are like "include" files; they contain a list of attributes. Adding the Auxiliary class to the definition of a Structural or Abstract class adds the Auxiliary class's attributes to the definition. An Auxiliary class cannot be instantiated in the directory, but new Auxiliary classes can be derived from existing Auxiliary classes. This type of class is specified by a value of 3 in the *objectClassCategory* attribute. For example, the *securityPrincipal* class is an Auxiliary class, and it derives its attributes from the parent abstract class called *top*. Although you cannot create a security principal object in the directory (because Auxiliary classes cannot have instances), you can create an object of the Structural class *user*, which has the *securityPrincipal* class as an auxiliary. The attributes of the *securityPrincipal* class contribute to making the user object recognizable to the system as a security account. Similarly, the *group* class has *securityPrincipal* as an Auxiliary class.

88 Classes Classes defined before 1993 are not required to fall into one of the preceding categories; assigning classes to categories was not required in the X.500 1988 specification. Classes that were defined prior to the X.500 1993 standards, default to the 88 class. This type of class is specified by a value of 0 in the *objectClassCategory* attribute. Do not define new 88 classes.

Note Active Directory does not return an error for 88 classes; it only performs looser semantic checking. For example, an 88 class can be used as an abstract superclass, but it can also be directly instantiated. When you define new schema classes, you need to use one of the X.500 1993 categories.

Inheritance

Inheritance, which is also referred to as derivation, is the ability to build new object classes from existing object classes. The new object is defined as a *subclass* of the parent object. A subclass is a class that inherits from some other class; for example, a subclass inherits structure and content rules from the parent. The parent object becomes a *superclass* of the new object. A superclass is a class from which one or more other classes inherit information. The inherited information includes mandatory and optional attributes (*systemMustContain*, *mustContain*, *systemMayContain*, and *mayContain*) and its parent classes in the directory hierarchy (*systemPossSuperiors* and *possSuperiors*). The diagram in Figure 4.2 illustrates an object class hierarchy.

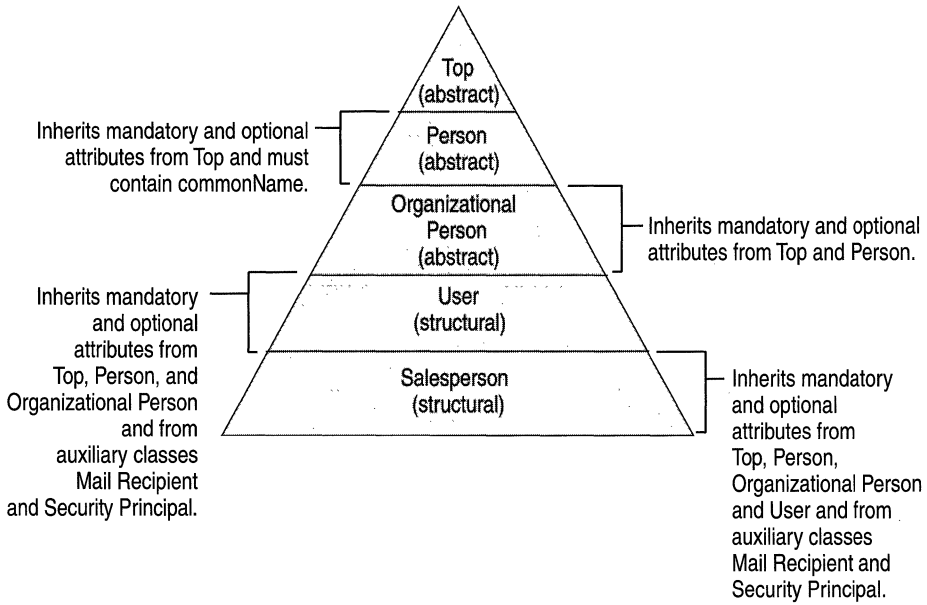


Figure 4.2 Object Class Hierarchy

For example, you can specify a *Salesperson* class that defines information about the salespeople in your company, including specialized information, such as commission rate and travel route. You can specify the *Salesperson* class as *subClassOf* of the *User* class. This would cause the *Salesperson* class to inherit all the mandatory and optional attributes and directory-parent classes of the *User* class after the schema cache is updated. You would not have to define these attributes for each salesperson in your company.

All structural object classes are subclasses, directly or indirectly, of a single abstract object class, which is called *top*. Every object represented in the directory belongs to *top* and, as a result, every entry must have an *objectClass* attribute. When you create a new class, you must specify the superclass: If you are not creating a subclass of an existing class, the new class is a subclass of *top*.

A new class can inherit mandatory and optional attributes from more than one existing class. However, any additional classes must be specified by the *auxiliaryClass* attribute.

Note If you add another attribute later to a class that has subclasses or auxiliary subclasses, the new attribute is automatically added to the subclasses after the schema cache has been updated.

To view a graphical representation of the Active Directory class hierarchy, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

System and Changeable Attribute Pairs

Some aspects of a class-definition object are contained in pairs of attributes, where the value of one of these attributes can be changed by administrators and the other cannot. These attribute pairs are *mustContain/systemMustContain*, *mayContain/systemMayContain*, *possSuperiors/systemPossSuperiors*, and *auxiliaryClass/systemAuxiliaryClass*.

In each of these pairs, the value of the attribute that begins with the word *system* cannot be changed by administrators. This enables Active Directory to protect certain key attributes of certain classes and ensure that the schema stays consistent and usable. System-only properties can only be changed by the directory system agent (DSA). *System-only properties* are those properties on which Windows 2000 and Active Directory depend for normal operations. For example, the *attributeID* and *governsID* attributes in the schema are system-only attributes. The value of the other (nonsystem) attributes in each pair can be changed by administrators.

Mandatory Attributes

The term *mandatory attributes* refers to object attributes for which values must be specified. If you do not specify a value for a mandatory attribute, one of the following happens:

- The attribute takes on a default value.
- The object is not created until you specify a value for the attribute.

Which of the object's attributes are mandatory is determined by the class to which the object belongs.

Some mandatory attributes are inherited. Because every *schemaClass* object belongs to a subclass called *top* in the class hierarchy, each *schemaClass* object inherits the mandatory attributes that belong to *top*. Table 4.2 is a list of the mandatory attributes that every object inherits from *top*. To see a graphical representation of the Active Directory class hierarchy, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Table 4.2 Mandatory Attributes That All *schemaClass* Objects Inherit

Inherited Mandatory Attribute	Default Status
nTSecurityDescriptor	Defaults if not specified. The default value depends on the default security descriptor for the classSchema class.
objectCategory	Defaults to the value of the default object category of the class (which is usually the class itself). Can be changed after the class is created.
objectClass	No default. Administrator must specify the class.

Note You can view an object's mandatory attributes by using the Active Directory Schema snap-in. (The attributes are displayed on the **Attributes** tab in the **Properties** dialog box.) Because some of an object's mandatory attributes are inherited from its parent class, you might need to view the attributes of the parent class in order to identify all of the mandatory attributes of your object. The Active Directory Schema snap-in is an MMC tool that is provided by Windows 2000 to enable administrators to modify the schema by using a graphical interface.

Attributes for *classSchema* Objects

Table 4.3 is a list of the attributes a *classSchema* object can have.

Table 4.3 Attributes of a *classSchema* Object

Attribute	Syntax	Mandatory?	Multi-value?	Description
cn	Unicode	Yes	No	Descriptive relative distinguished name for the schema object.
GovernsID	Object identifier	Yes	No	Object identifier that uniquely identifies this class.
LDAPDisplayName	Unicode	Yes	No	Name by which LDAP clients identify this class.
SchemaIDGUID	String(Octet)	Yes, but defaulted.	No	GUID that uniquely identifies this class.
RDNAttID	Object identifier	No	No	Relative-distinguished-name-type of instances of this class (OU, CN).
SubClassOf	Object identifier	Yes	No1	The class from which this object inherits attributes.
SystemMustContain	Object identifier	No	Yes	The list of mandatory attributes for instances of this class. This list cannot be changed.
MustContain[2]	Object identifier	No	Yes	The mandatory attributes for instances of this class.
SystemMayContain	Object identifier	No	Yes	The optional attributes for instances of this class.
MayContain[2]	Object identifier	No	Yes	The optional attributes for instances of this class.
SystemPossSuperiors[2]	Object identifier	No	Yes	The classes that can be parents of this class in the directory hierarchy. After creation of the class, this property cannot be changed.
PossSuperiors[2]	Object identifier	No	Yes	The classes that can be parents of this class in the directory hierarchy. For an existing *classSchema* object, values can be added to this property but not removed.

(Continued)

Table 4.3 Attributes of a *classSchema* Object *(Continued)*

Attribute	Syntax	Mandatory?	Multi-value?	Description
systemAuxiliaryClass[2]	Object identifier	No	Yes	The Auxiliary classes from which this class inherits its optional (*mayContain*) and mandatory (*mustContain*) attributes. After creation of the class, this property cannot be changed.
AuxiliaryClass[2]	Object identifier	No	Yes	The Auxiliary classes from which this class inherits its optional (*mayContain*) and mandatory (*mustContain*) attributes. A multivalue property that specifies the auxiliary classes that this class inherits from. For an existing *classSchema* object, values can be added to this property but not removed.
DefaultHidingValue	BOOL	No	No	The default hiding state for the class. If you do not want instances of your class displayed in the user interface, you can define the class as hidden.
DefaultSecurity Descriptor	String(Octet)	No	No	The default security descriptor that is assigned to new instances of this class if no security descriptor is specified during creation of the class or is merged into a security descriptor if one is specified.
ObjectClassCategory	Integer	Yes	No	Class types are defined as follows: 88 Class = 0; Structural = 1; Abstract = 2; Auxiliary = 3.
SystemOnly	BOOL	No	No	If TRUE, only the system can create and modify instances of this class.
ObjectClass	Object Identifier	Yes	Yes	This object's class, which is always *classSchema*.
NTSecurityDescriptor	NT-Sec-Desc	Yes	No	Security descriptor on the classSchema object.
DefaultObjectCategory	Distinguished name	Yes	No	The default object category of new instances of this class if none has been specified.

[1] Objects cannot inherit from more than one class by using this attribute. Use the *auxiliaryClass* attribute to define additional parent classes.

[2] Each value is the lDAPDisplayName of a class that is a class object identifier. Note that you must ensure that the classes exist or will exist when the new class is written to the directory. If one of the classes does not exist, the *classSchema* object is not added to the directory.

Note When you look at the attributes in a *classSchema* object's *mustContain* attribute list, you are not seeing the complete set of attributes that must be present for an instance of a class to exist. For example, in the class A, the *classSchema* object B specifies a list of *mustContain* attributes that an instance of A must have through the *systemMustContain* and *mustContain* attributes. However, because mandatory attributes are also inherited, the complete list of attributes for an instance of class A includes the inherited *mustContain* attributes from all classes from which B inherits—that is, all classes in the *subClassOf* and *auxiliaryClass* lists for the *classSchema* object B. The *mayContain* attributes for object B are also defined this way. The *possSuperiors* are defined this way as well, except that *possSuperiors* are inherited only from classes in the *subClassOf* list, not from the classes in the *auxiliaryClass* list.

Syntaxes

The *syntax* for an attribute defines the storage representation, byte ordering, and matching rules for comparisons of property types. Whether the attribute value must be a string, a number, or a unit of time is also defined. Every attribute of every object is associated with exactly one syntax. The syntaxes are not represented as objects in the schema, but they are programmed to be understood by Active Directory. The allowable syntaxes in Active Directory are predefined. You cannot add new syntaxes.

When you define a new attribute, you must specify both the *attributeSyntax* and the *oMSyntax* numbers of the syntax you want for the attribute. The *attributeSyntax* number is an object identifier and *oMSyntax* number is an integer. The *oMSyntax* is defined by the XOM specification. This model provides a relatively fine-grained definition of syntax. For example, there are distinct *oMSyntax* attributes to distinguish among several types of printable strings, according to factors such as the supported character set and whether case is significant. Table 4.4 is a list of the valid syntaxes for attributes in the Active Directory schema.

Table 4.4 Valid Syntaxes for Attributes in the Active Directory Schema

Syntax[1]	*attribute*Syntax	oM Syntax	ASN 1-Encoded Object Identifier	Description
Undefined	2.5.5.0		\x550500	Not a legal syntax.
Object(DN-DN)	2.5.5.1	127	\x550501	The fully qualified name of an object in the directory.
String(Object-Identifier)	2.5.5.2	6	\x550502	The object identifier.
Case-Sensitive String	2.5.5.3	27	\x550503	General String. Differentiates uppercase and lowercase.
CaseIgnoreString(Teletex)	2.5.5.4	20	\x550504	Teletex. Does not differentiate uppercase and lowercase.
String(Printable), String(IA5)	2.5.5.5	19, 22	\x550505	Printable string or IA5-String. Both character sets are case-sensitive.
String(Numeric)	2.5.5.6	18	\x550506	A sequence of digits.
Object(DN-Binary)	2.5.5.7	127	\x550507	A distinguished name plus a binary large object.
Boolean	2.5.5.8	1	\x550508	TRUE or FALSE values.
Integer, Enumeration	2.5.5.9	2, 10	\x550509	A 32-bit number or enumeration.
String(Octet)	2.5.5.10	4	\x55050A	A string of bytes.
String(UTC-Time), String(Generalized-Time)	2.5.5.11	23, 24	\x55050B	UTC Time or Generalized-Time.
String(Unicode)	2.5.5.12	64	\x55050C	Unicode string.
Object(Presentation-Address)	2.5.5.13	127	\x55050D	Presentation address.
Object(DN-String)	2.5.5.14	127	\x55050E	A DN-String plus a Unicode string.
String(NT-Sec-Desc)	2.5.5.15	66	\x55050F	A Microsoft® Windows NT® Security descriptor.
LargeInteger	2.5.5.16	65	\x550510	A 64-bit number.
String(Sid)	2.5.5.17	4	\x550511	Security identifier (SID).

[1]The *oMSyntax* names are specified against the syntax numbers to enable correct choice.

Note A complete syntax specification consists of both the attribute-syntax and the *oMSyntax*. Whenever more than one *oMSyntax* can be used with an attribute-syntax, the correct *oMSyntax* must be used.

Active Directory does not currently enforce character set restrictions for string syntaxes, so if you use attributes with string syntax, use only characters in the standard character set.

Object Identifiers

Object identifiers are unique numeric values that are granted by various issuing authorities to identify data elements, syntaxes, and other parts of distributed applications. Because they are globally unique, object identifiers ensure that the objects that are defined by these issuing authorities 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.

Object identifiers are found in Open Systems Interconnection (OSI) applications, X.500 directories, Simple Network Management Protocol (SNMP), and other applications in which uniqueness is important. Object identifiers are based on a tree structure in which a superior issuing authority allocates a branch of the tree to a subordinate authority, which in turn allocates sub-branches of the tree.

LDAP requires a directory service, like Active Directory, to identify object classes and attributes with an object identifier syntax. The object identifier is the value for the *governsID* attribute in a class-schema object and for the *attributeID* attribute in an *attributeSchema* object. These are required attributes; therefore, object identifiers are necessary when you create new classes or attributes.

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, 1.2.840.113556.1.5.4), the components of which are shown in Table 4.5.

Table 4.5 Components of a Sample Object Identifier (1.2.840.113556.1.5.4)

Numerical Values of the Sample Object Identifier	What the Numerical Values Denote	
1	ISO ("root" authority)	Issued 1.2 to ANSI, which in turn . . .
2	ANSI	Issued 1.2.840 to USA, which in turn . . .
840	USA	Issued 1.2.840.113556 to Microsoft, which . . .
113556	Microsoft	Internally manages several object identifier branches under 1.2.840.113556 that include . . . .
1	Active Directory	A branch called Active Directory that includes . .
5	Classes	A branch called Classes that includes . . . .
4	Builtin-Domain	A class called Builtin-Domain.

Object identifiers ensure that every object is interpreted appropriately—for example, that a telephone number is not mistaken for an employee number. A series of widely used objects and attributes is standardized for use in object identifiers. New object identifiers are issued by standards authorities, and they form a hierarchy below which new object identifiers can be managed internally. An object identifier is represented as a dotted decimal string (for example, 1.2.3.4). Enterprises (and individuals) can obtain a root object identifier from an issuing authority and use it to allocate additional object identifiers internally. For example, Microsoft Corporation has been issued the root object identifier 1.2.840.113556. Microsoft manages further branches from this root internally. One of these branches is used to allocate object identifiers for Active Directory classes, another for Active Directory attributes, and so forth.

Most countries and regions in the world have an identified National Registration Authority (NRA) responsible for issuing object identifiers to enterprises. In the United States, the NRA is the American National Standards Institute (ANSI). The NRA issues root object identifiers. An enterprise can register a name for the object identifier as well. There is a fee associated with registering the root object identifiers and registered names. Contact the NRA for your country or region for details. The International Standards Organization (ISO) recognizes NRAs and maintains a list of contacts on their Web site.

The issuing authority assigns an object identifier space that is a branch of the ISO-International Telecommunications Union (ITU) object identifier tree. Assume that your company is assigned the space 1.2.840.111111. You can extend this space internally as you want (within the constraints of the structure of an object identifier). For example, you can subdivide this space further (by appending dotted decimals to the object identifier root) and assign these subspaces to various divisions within your company. Each division, in turn, can further subdivide the subspace allotted to it. For example, by using the sample object identifier 1.2.840.111111, your company might have the subspace 1.2.840.111111.1.4 for attributes and 1.2.840.111111.1.5 for classes. An internal issuing authority within the company, using an Administrator account, might then allocate object identifiers from this space on request. The *governsID* attribute on every *classSchema* object and the *attributeID* attribute on every *attributeSchema* object are mandatory attributes that contain an object identifier string. In this example, all of your company-created *classSchema* objects have a *governsID* of the form 1.2.840.111111.1.5.*x*, where *x* is a decimal number. Similarly, all of your company-created *attributeSchema* objects have an *attributeID* of the form 1.2.840.111111.1.4.*x*.

Structure and Content Rules

The schema enforces rules that govern both the structure and the content of Active Directory. When you add, delete, or modify objects, validation takes place by using these schema rules to ensure the integrity of the directory. Structure rules define the possible tree structures. When you create a new object, structure rules determine the validity of the object class to which you designate the new object. You cannot create an object that belongs to a nonexistent class. You must first create the new class. Conversely, these rules do not allow you to delete or modify an object that has already been deleted. In Active Directory, the structure rules are completely expressed by the *possSuperiors* and *systemPossSuperiors* attributes that are present on each *classSchema* object. These attributes specify the possible classes that can be parents of an object instance of the class in question. In other words, the *possSuperiors* and *systemPossSuperiors* attribute values determine the object classes and, hence, the location in the Directory Information Tree under which objects of the class in question can be instantiated.

Content rules determine the mandatory and optional attributes of the class instances that are stored in the directory. New objects must contain all of the mandatory attributes that are specified by the *classSchema* object in the schema and can contain any of the optional attributes. In Active Directory, the content rules are completely expressed by the *mustHave, mayHave, mayContain, systemMustContain,* and *systemMayContain* attributes of the schema definitions for each class. In addition, specific marked attributes have additional restrictions imposed by the Security Account Manager (SAM). SAM read-only objects consist of the following:

revision, objectSID, domainReplica, creationTime modifiedCount, modifiedCountAtLastPromotion, nextRID, serverState, samAccountType, isCriticalSystemObject, dbcsPwd, ntPwdHistory,lmPwdHistory, lastLogon, lastLogoff, badPasswordTime, badPwdCount ,logonCount, supplementalCredentials

Below are some other attributes on which SAM enforces special checks:

sAMAccountName. Domain-wide uniqueness, without replication latency, 20-character limit for user objects (not groups).

Member. Membership rules as defined in Windows 2000 groups.

userWorkstations. Must be valid computer names.

primaryGroupID. For a user/computer account, must point to a group and the user /computer account must be a member of the group; the group and the user must be in the same domain. If the computer is a domain controller, the primary group must be the domain controllers group.

LockoutTime. For a user or computer object. Only legal value that can be written is 0 to clear an account.

LockoutPasswordLastset. The system normally writes to it, but two special values can be written 0 and -1 to expire /unexpire a password.

For more information about these attributes, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Schema Cache

All changes made to Active Directory are validated first against the schema. For performance reasons, this validation takes place against a version of the schema that is held in memory on the domain controllers. This "in-memory version," called the *schema cache*, is updated automatically after the on-disk version has been updated. The schema cache provides mapping between attribute identifiers such as a database column identifier or a MAPI identifier and the in-memory structures that describe those attributes. The schema cache also provides lookups for class identifiers to get in-memory structures describing those classes.

When the computer is started, the schema cache is loaded from the underlying database and updated automatically whenever the on-disk version is updated. When changes are made to the schema, the schema cache is automatically updated within five minutes after the first change was applied. During the interval before the schema updates are copied to the schema cache, objects that reference a new or modified class or attribute cannot be added. This behavior keeps the cache consistent, but it can be confusing because changes are not apparent until the cache is updated, even though they were applied on disk.

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; it acts as a trigger or operational attribute. Writing this attribute a cache reload.

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,* used only for administering the directory server itself. It is an artifact attribute that is never defined in the schema and does not require any storage. Generally, when you set an operational attribute, you trigger some action on the server.

Adding the *schemaUpdateNow* attribute causes a schema cache update to start immediately. The call is blocking, which means that if the call returns with no error, the cache is updated and all schema updates are ready to be used. An error return, however, indicates that the cache update is not successful. It is recommended that applications that want to take advantage of this feature be designed to accommodate the blocking write, particularly in giving the user feedback, if the program or script runs interactively.

Important It is recommended that you force an immediate schema cache update only once and only after all required schema updates are finished because cache loads are expensive in terms of memory.

Default Security of Active Directory Objects

The default security descriptor for an Active Directory object is specified in the schema. Essentially there are two segments to the default Active Directory security configuration or default access rights granted.

- Initial security for all objects created while installing Active Directory.
- Default security for objects created after installing Active Directory.

For information about the default security descriptors for Active Directory objects, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. For information about permissions and security descriptors, see "Access Control" in this book.

Note There are special cases where default security is not applied on newly created objects. For more information about these situations, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Default Security of the Domain Directory Partition

The domain directory partition object is derived from the object class *domainDNS*; therefore, the default security is equivalent to the default security for the object class *domainDNS*.

The default security descriptor for the domain directory partition comprises the following:

- Full control permissions to the Domain Administrators group and the System group, and Read permissions to the Authenticated Users group.
- Read property on all properties to the Everyone group. This permission provides backward compatibility for application programming interfaces (APIs).
- Replicating Directory Changes, Replication Synchronize, and Manage Replication Topology permissions to the Enterprise Domain Controllers group. These permissions allow members of the Enterprise Domain Controllers group to manage replication automatically.
- Replicating Directory Changes, Replication Synchronize, and Manage Replication Topology permissions to the Builtin Administrators group. Administrators of individual domain controllers can use these permissions to troubleshoot replication problems.
- Inheritable Full Control to the Enterprise Administrators group. Enterprise Administrators, by definition, have complete control of each domain.

- Inheritable List Contents to the Pre-Windows 2000 Compatible Access group.
- Inheritable Read Property on RAS Information, General Information, Membership, User Account Restrictions, and User Logon on all User Objects permissions to the Pre-Windows 2000 Compatible Access group.
- Inheritable Read on all Group objects.
- Inheritable Auditing successful/failed Writes to the Everyone group.

Activating the auditing policy ensures that writes that are performed on the directory (on any object) are audited immediately without the need for any extra user intervention. Inheritable access control entry (ACE) provides a convenient way of removing auditing policy.

Default Security of the Configuration Directory Partition

The default security descriptor for the configuration directory partition comprises the following:

- Full control permissions to Domain Administrators, and System and Read permissions to the Authenticated Users.
- Replicating Directory Changes, Replication Synchronize, and Manage Replication Topology permissions to the Enterprise Domain Controllers group. These permissions enable domain controllers in the forest to replicate from each other and automatically reconfigure the replication topology on the basis of replication delays and latency for the configuration directory partition.
- Replicating Directory Changes, Replication Synchronize, and Manage Replication Topology permissions to the Builtin Administrators group. These permissions enable administrators from individual domain controllers to synchronize replication and topology management for the configuration directory partition.
- Enable Inheritable Full Control to the Enterprise Administrators group. This permission allows members of the Enterprise Administrators group exclusive control over the Configuration container. The Enable Inheritable Full Control permission is required to control the Configuration container throughout the forest.
- Enable Inheritable Auditing to the Writes by the Everyone group. Activating the auditing policy ensures that writes that are performed on the directory (on any object) are audited immediately without the need for any extra user intervention. Inheritable ACE provides a convenient way of removing auditing policy.

Default Security of the Schema Directory Partition

The default security descriptor for the schema directory partition comprises the following:

- Write property permission on the *fSMORoleOwner* attribute to the Schema Administrators group. This permission enables members of the Schema Administrators group to forcibly transfer the domain controller where schema changes are made.
- Change Schema Master control permission to the Schema Administrators group. This permission enables members of the Schema Administrators group to change (per the Flexible Single-Master Operation [FSMO] protocol) the domain controller where schema changes are made.
- Inheritable Full Control permission designated to the Schema Administrators group. By default, the Schema Administrators group is the only group that has write access to the entire schema container. A schema object does not have any exclusive control over its own security, thus the object inherits its security from the schema container.
- Replicating Directory Changes, Replication Synchronize, and Manage Replication Topology to the Enterprise Domain Controllers group. These permissions enable the members of the Enterprise Domain Controllers group to manage replication of the schema in the forest automatically.
- Replicating Directory Changes, Replication Synchronize, and Manage Replication Topology permissions to the Builtin Administrators group. These permissions enable the administrators per domain controllers to resolve replication issues.
- Read permissions designated to the Authenticated Users group. This permission enables the members of the Authenticated Users group the right to read the schema.
- Audit successful/failed Writes by the Everyone group. Activating the auditing policy ensures that writes that are performed on the directory (on any object) are audited immediately without the need for any extra user intervention. Inheritable ACE provides a convenient way of removing auditing policy.

Default Security of Attributes and Classes

All attributes and classes inherit security from the ACLs on the Schema container. This ensures that the entire schema is consistent in terms of security.

Note The initial security allows only Schema Administrators write access to the Schema container

---

Extending the Schema

When the existing class and attribute definitions in the schema do not meet the needs of your organization, the schema can be extended by adding or modifying schema objects. The Active Directory schema can be extended dynamically. That is, an application can extend the schema with new attributes and classes and use the extensions immediately. Schema updates are accomplished by creating or modifying the schema objects that are stored in the directory. This allows you to make the objects that are meaningful to your organization available throughout the enterprise.

Note As is true for every object in Active Directory, schema objects are protected by access control lists (ACLs), so only authorized users can alter the schema. (For more information about ACLs, see "Access Control" in this book.)

Adding or modifying class or attribute definitions in the schema involves adding or modifying the corresponding *classSchema* object or *attributeSchema* object. The operations that are involved in this process are similar to adding or modifying any object in Active Directory, except that additional checks are performed to ensure that changes do not cause inconsistencies or problems in the schema in the future.

When to Extend the Schema

Modifying the schema is a major change, with implications throughout the directory. It is recommended that you modify the schema only when it is absolutely necessary. Many schema modifications cannot be reversed, so you must make sure that changes are planned and well thought out before they are implemented. Inconsistencies in the schema can cause significant problems that impair or disable Active Directory. These problems might or might not be evident immediately.

Planning for schema modification involves examining the default schema that comes with Active Directory to verify that there is no way to use the existing classes or attributes for your needs. It is then necessary to understand the types of modifications that can be made and, conversely, that cannot be changed. 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. The subclass has all the attributes of the original class and any additional attributes that you specify.
- Creating an entirely new class with any attributes that you want to assign.

You need to extend an existing class when the following conditions apply:

- The existing class needs additional attributes but otherwise meets your needs. For example, you might want to add a *purchasingLimit* attribute to the *User* class and add it to the user object for people who are cost center managers and have purchasing authority.
- You have no need to identify the extended class as a distinct class from the original class.
- You want to use the existing Active Directory Users and Computers console in MMC to manage the extended attributes of the objects. This requires the addition of property pages to the set defined for the object you are extending.

Derive a subclass from an existing class when the following conditions apply:

- The existing class needs additional attributes but otherwise meets your needs.
- You want to identify the extended class as a distinct class from the original class.
- You want to use the existing Active Directory Users and Computers console in MMC to manage the extended attributes of the objects.

How to Extend the Schema

After you have decided that you have to make changes to the schema and you have carefully planned the types of changes you are going to make, you can proceed. Because this is an extremely significant operation, and not without the possibility of causing serious problems, Windows 2000 has three safety features, or interlocks, that control modification of the schema:

- By default, schema modification is disabled on all domain controllers. Use the Active Directory Schema console on a domain controller to permit write access to the schema on that domain controller.

- The schema object is protected by the Windows 2000 security model. Therefore, administrators must be given explicit permissions or be a member of the Schema Administrators group (**Schema Admins** in the user interface) to effect changes to the schema.
- Only one domain controller in the enterprise, the one holding the Schema Master Role, is allowed to write to the schema. This role is one example of an FSMO role.

Installation of Schema Extensions

The recommended practice is to strictly control the schema updates at most customer sites. If a service requires schema extensions, you must be able to install them separately by using one of the following methods

- Extending the schema by using LDIF scripts. This allows customers to update the schema separately and in advance of the rest of the installation.
- Extending the schema programmatically.

In addition to providing a separate installation procedure for schema extensions, it is recommended that the nature of the schema extensions be clearly documented. The documentation needs to contain the following:

- A statement that describes the authority from which your object identifier prefix was obtained.
- The common-name (cn), the LDAP-Display-Name, the object identifier, and the description of each new class and attribute and its expected use. Also answer the following questions:
  - Is the attribute configured for replication to Global Catalog servers?
  - Is the attribute configured for indexing?
  - What are the expected update frequency and expected size of the attribute, which allows the customer to make calculations of replication impact?
  - What are the *rangeLower* and *rangeUpper* values?
- A class hierarchy showing newly created classes.
- If defined, the values for the Default-Security-Descriptor and the NT-Security-Descriptor.

The schema installation program must allow the user to exit the program prior to your making any changes to the schema.

Specify the Schema-ID-GUID

Specify the schemaIDGUID when you create attributes or classes in Active Directory. The schemaIDGUID is a globally unique identifier (GUID) that uniquely identifies all classes and attributes in the schema. Unlike object identifiers, which are issued by a central authority, a special algorithm generates GUIDs. SchemaIDGUIDs are used in ACLs to provide attribute-specific or class-specific privileges.

Naming

When you modify the schema, you must adhere to the following rules with respect to specifying the relative distinguished name attribute (which is common-name [cn]) and the LDAP display name (lDAPDisplayName).

Common-Name (cn)

- Choose a company prefix. This section of the prefix must be the registered DNS domain name of the company and the current year (four digits, separated by a hyphen (-).
- Make the next token in the cn a hyphen (-).
- Choose a product-specific prefix. This section of the name must be unique within your company and a succinct identification of the product and needs to begin with an uppercase letter. The letters in the remainder of the prefix can be uppercase or lowercase as you deem appropriate.
- Make the next token in the cn a hyphen (-).
- Make the next section of the cn the name of the attribute or class separated by hyphens.

LDAP-Display-Name

- Use the Common-Name (cn) as the starting point for the LDAP-Display-Name (lDAPDisplayName).
- Make the first character LDAP-Display-Name lowercase.
- Make the character that immediately follows each hyphen (-) uppercase.
- Remove all hyphens that follow the product-specific section of the prefix except for the hyphen that immediately follows this section.

Table 4.6 illustrates the naming rules as they are applied to the Common-Name (cn) and the LDAP-Display-Name (lDAPDisplayName):

Table 4.6 Naming Rules

Common-Name (cn)	LDAP-Display-Name (lDAPDisplayName)
Microsoft-Com-1999-MQ-Attribute-1	Microsoft-Com-1999-mQAttribute1
Microsoft-Com-1999-EXCHANGE-Attribute-2	Microsoft-Com-1999-exchangeAttribute2

Modifying the Schema

To allow a domain controller to modify the schema, use the Active Directory Schema console in MMC on the selected server.

Note Because of the serious nature of schema modification, the Active Directory Schema MMC snap-in is not listed with the default MMC snap-ins that are provided with Windows 2000 Server. To make it appear in the list of available MMC snap-ins, you must run **Regsvr32** on the dynamic-link library (DLL) (Schmmgmt.dll) from the command prompt.

▶ **To enable schema modification**

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 server** check box, and then click **OK**.

The value of the **The Schema may be modified on this server** 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.

Caution Do not use a registry editor to edit the registry directly unless you have no alternative. The registry editors bypass the standard safeguards provided by administrative tools. These safeguards prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your system. Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. To configure or customize Windows 2000, use the programs in Control Panel or MMC whenever possible.

Schema Administrators Group

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.

Caution 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 Administrators**

1. Open the Active Directory Users and Computers console.
2. Expand the domain for the account by clicking the plus sign (+) next to it.
3. Double-click the **Users** folder.
4. Double-click the **Schema Admins** security group, and then click the **Members** tab.
5. If the account is not listed under **Members**, click **Add**.
6. Select an account from the displayed list, or type the name of the account.
7. Click **Add**, and then click **OK**.

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.

Note To update the schema, the domain controller holding the schema master role must be online.

You can change the domain controller that serves as the schema master at any time according to your needs. This is what is meant by the word "flexible" in FSMO. The current schema master in the enterprise is identified by the value of the *fSMORoleOwner* attribute on the Schema container of the domain. By default, the first domain controller that is installed in the enterprise is the initial schema master.

Although the domain controller that is the current FSMO Role Owner for schema operations is the only one that can make the actual schema modifications, you do not have to be connected to that domain controller when you make schema modifications. If you are connected to a domain controller that does not have that role, it generates a referral to the current FSMO Role Owner to process the modifications.

If you want to do so, you can transfer the role of schema master to another domain controller by using the Active Directory Schema console in MMC.

▶ **To view or change the current schema master by using the Active Directory Schema console in MMC**

1. Open MMC, and install the Active Directory Schema snap-in.
2. Right-click the Active Directory schema, and then click **Operations Master**.

3. The Current Operations Master that is displayed is the schema master.

   To retain the current schema master, click **OK**.

   –Or–

   To change the server that is the current FSMO Role Owner for the schema, click **Change**.

   If the current domain controller (the one that is listed in **Current Focus**) is also the current operations master, you must use the Active Directory Tree console to focus on another domain controller before you can change the operations master. This is because you must be connected to the domain controller that you want to have as the FSMO Role Owner. You cannot direct the connected domain controller to make another domain controller the FSMO Role Owner.

For more information about using the Active Directory Schema console, see "Modifying the Schema" earlier in this chapter.

You can also use the command-line tool Ntdsutil to transfer the Schema FSMO. The tool resides in the \%SystemRoot%\System32 folder. For more information about transferring FSMO roles by using Ntdsutil, see "Managing Flexible Single-Master Operations" in this book.

▶ **To change the schema master by using Ntdsutil**

1. Start Ntdsutil by typing **ntdsutil** at the command prompt. (Note that at any prompt in this tool, you can type a question mark (**?**) to see the list of valid commands for that prompt.)
2. At the Ntdsutil prompt, type:

   roles

3. At the **fsmo maintenance** prompt, type:

   connections

4. To display the current connection information, at the server connections prompt, type:

   info

   If necessary, type the appropriate command to connect to the server that is to become the schema master. (Use the **?** command to see a list of valid commands.)

5. To return to the **fsmo maintenance** prompt, type:

   quit

6. To do a graceful transfer of the Schema FSMO, type:

   transfer schema master

You can also perform the schema master role transfer in a program. Before a program can make changes to the schema, it must check explicitly whether the domain controller is the current schema master and, if it is not, explicitly request the transfer operation.

To understand the transfer process, consider a scenario in which computer A is the current FSMO Role Owner and computer B must perform some schema updates. To request an FSMO Role Owner transfer from computer A, a program must add the operational attribute *becomeSchemaMaster* with value of 1 to the rootDSE (that is, to the object with a blank distinguished name) on computer B. It is an operational attribute that is never defined in the schema and does not require any storage. Generally, when you set an operational attribute, you trigger some immediate action on the server.

In this case, the action taken by the server (computer B) is its sending out a request to computer A for a role transfer. Computer A, upon receiving such a request, changes the *fSMORoleOwner* attribute on its Schema container to the name of computer B and sends this new attribute value back to computer B. It also sends back any schema changes that were implemented on computer A but were not yet incorporated by computer B. (This kind of discrepancy is possible as a result of replication latencies.) Computer B, upon receiving the reply from computer A, applies all changes that were sent back from computer A and, in the process, becomes the current schema master.

Note Computer B, the new schema master, now has all previous schema updates in the enterprise and, hence, the latest version of the schema.

If the old schema master is unavailable or has crashed, you can forcibly transfer (*seize*) the schema FSMO so that a new domain controller can make schema changes. However, it is recommended that you take this step only as a last resort. When the schema master is forcibly transferred to a new domain controller, recent schema changes that were made at the old schema master might not be propagated to the new schema master and might be lost. The transfer also can result in conflicting updates at other domain controllers in the forest, which might require an extensive offline cleanup of the directory.

Caution Seizing the schema master is a drastic step that you must consider only when the current schema master is no longer able to function and is never going to be available again. Before you seize the current schema master, remove it from the network. Verify that the domain controller that seizes the role is fully up-to-date with respect to updates performed on the previous role owner.

▶ **To seize the schema master by using Ntdsutil**

1. Start Ntdsutil by typing **ntdsutil** at the command prompt. (Note that at any prompt in this tool, you can type a question mark (**?**) to see the list of valid commands for that prompt.)
2. At the Ntdsutil prompt, type:

   roles
3. At the fsmo maintenance prompt, type:

   connections
4. To display the current connection information, at the server connections prompt, type:

   info

   If necessary, type the appropriate command to connect to the server that is to become the schema master. (Use the **?** command to see a list of valid commands.)
5. To return to the fsmo maintenance prompt, type:

   quit
6. To perform a forced transfer (*seizure*) of the schema master, type:

   seize schema master

For more information about FSMOs, see "Managing Flexible Single-Master Operations" in this book.

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. Target your update at the FSMO Role Owner. Bind to the schema on the domain controller that is the schema master. Avoid unnecessarily changing the schema master role between domain controllers. Only one domain controller is allowed to perform critical operations like updating the schema at any one time. This domain controller is known as the FSMO Role Owner. If you have more than one Windows 2000 server on your network, your current server might not be the FSMO Role Owner. You have to ensure that you target your update at the FSMO Role Owner.
2. Ensure that you have sufficient administrative privileges to perform the schema update. Check the *allowedChildClassesEffective* property of the Schema container to see if you can create attributes or classes. If *attributeSchema* and *classSchema* are not values in that property, you do not have sufficient rights to add attributes or classes to the schema. Only members of the Schema Administrators group are allowed to alter the contents of the schema. You must ensure that your user account is a member of this group. (The Administrator account is automatically a member of the Schema Administrators group.)
3. Create the registry entry that allows write access to the schema. By default, access to the schema is read-only. This entry, known as the safety interlock, can be found in the registry in HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS\Parameters**Schema Update Allowed**. This entry must exist and its value must be nonzero for schema updates to take place. Check that the safety interlock is engaged before removing it. Note the value that you found for this entry, and make sure to leave the value in the same state afterward. Note that you only have to create the safety interlock on the server that holds the FSMO role.
4. Add your new attributes.
5. Add your new classes.
6. Add attributes to classes. Any new attributes need to be referenced by object identifier because their names are not going to be present in the cache yet. Unless you trigger a schema cache reload after you add new attributes, an attempt to use an attribute by name is going to fail.
7. Each domain controller updates its schema cache five minutes after a schema change. If the extensions are going to be used within five minutes, you must trigger a cache reload.

8. If you had to create the safety interlock before you added your new classes or attributes, it is recommended that you re-apply the safety interlock again after you add them.
9. If you are installing a schema extension by programmatic means (script or ADSI), you must make sure that the extension is provided as a separately installable routine. This means that you must be able to do it separately from the application installation process.
10. Before you create a program to perform a schema extension, see the Microsoft Platform SDK link on the Web resources page at http://windows.microsoft.com/windows2000/reskit/webresources. Follow the links to "Active Directory Programmer's Guide" and then to "Schema Extensibility."

Note A cache update is not necessary if the schema extensions are not to be used immediately. Depending on system load, the extensions appear in the schema cache in approximately five minutes.

Adding and Modifying Schema Objects

Because schema objects are another kind of directory object, you can use the same methods that you would use to add or modify any directory object. Windows 2000 provides an administrative tool called Active Directory Schema that provides a straightforward user interface, and, of course, you have the option of making changes to the schema programmatically.

Adding an Attribute

It is recommended that you try to use existing attributes wherever possible. If you decide that you need to create a new attribute, however, you must adhere to the following guidelines:

- Use *cn* as the *name* (relative distinguished name) attribute; this is the default for most classes, including those derived directly from *top*. Because *cn* is an indexed attribute, it allows an efficient search for your object by name.
- Large multivalue attributes are costly to store and retrieve; it is recommended that you avoid using them. Active Directory implements an LDAP control to allow an incremental read of large multivalued attributes, but not all LDAP clients know how to use this control.
- Remember that attributes are "flat," which means that there is no implied substructure to an attribute. All attributes in a specific class must relate directly to instances of that class. This is also good data normalization practice.

To add a new attribute to the schema, you must create a new attribute object. First create the Active Directory safety interlocks as described in "How to Extend 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 syntax of the attribute.
4. Decide whether the attribute needs to be a single-value or multivalue attribute.
5. Decide whether and how the attribute needs to be indexed.
6. Decide whether the attribute needs to be 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 4.7 and Table 4.8.

Table 4.7 Mandatory Attributes for New Attribute-Definition Objects

Mandatory *Attributes*	Default Status
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 that are 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.
lDAPDisplayName	Defaults from the common name if not specified by the administrator.

Table 4.8 Optional Attributes for New Attribute-Definition Objects

Optional Attributes	Default Status
rangeLower	No default. The administrator must specify a value.
rangeUpper	No default. The administrator must specify a value.
isMemberOfPartial ReplicaSet	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).

As an example, suppose you want to add a new attribute called *userName*. Each instance of a *userName* attribute stores exactly one Unicode string of at least one character and not more than 1,000 characters. In this case, you would add the following attribute definition:

- *cn* = *userName*
- *objectClass* = *attributeSchema*
- *attributeID* = 1.2.567.8901234.5.6.879 (Valid object identifier value)
- *attributeSyntax* = 2.5.5.12 (Syntax value for Unicode string)
- *oMSyntax* = 64 (Syntax value for Unicode string)
- *isSingleValued* = TRUE (The intention is to store exactly one value.)
- *rangeLower* = 1 (Minimum length of the string)
- *rangeUpper* = 1000 (Maximum length of the string)

Modifying an Attribute

To modify an attribute, modify the existing attribute-definition object that represents the class. For reasons of consistency and security, some attributes of each attribute-definition object are designated as system-only. You cannot modify system-only attributes of an attribute object, not even for new classes that you originally created. System-only attributes are designated by having the *systemOnly* attribute of the attribute set to TRUE.

The following attributes of an attribute-definition object are *systemOnly* and, thus, 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, as described in "How to Extend the Schema" earlier in this chapter, 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 4.9 and Table 4.10. If you do not define values for some of these attributes, they are given default values.

Table 4.9 Mandatory Attributes for New Class-Definition Objects

Attribute	Default Status
cn	No default. Administrator must specify a name.
objectClassCategory	Defaults to 88 class because it is assumed to be a class with no category. Other options are Structural, Abstract, or Auxiliary.
governsID	No default. Administrator must specify an object identifier string.
possSuperiors	No default. Administrator must specify the structural class or classes that are legal parents of instances of this class.
subClassOf	No default. Administrator must specify a value.
schemaIDGUID	Defaults if not specified. The default value is automatically generated by the system.
nTSecurityDescriptor	Defaults if not specified. The default value depends on the *default SecurityDescriptor* of the *classSchema* class.
lDAPDisplayName	Defaults from the common name if not specified.

Table 4.10 Optional Attributes for New Class-Definition Objects

Optional	Default Status
defaultSecurityDescriptor	If there is no default security descriptor 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, to make the new class actually useful, you probably also want to define some attributes in *mustContain*, *mayContain*, and *possSuperiors*. Any attributes you specify when you add a new class must already exist. So, if you want to add a new class with new attributes, you must add the new attributes to the schema first.

When you add a new class, the object identifier specified in *governsID* must be unique, not only in your enterprise but also globally.

Note The system imposes rules that restrict the addition of schema objects.

Suppose you want to add a new class "*Friend*" to store information about a friend. Any *Friend* object must contain the name of the friend and might also contain her address or phone number. And because a friend is a person, you want objects of the *Friend* class to have the same mandatory attributes, optional attributes, and directory superiors as the *Person* class you have already defined. In this case, you add the following class definition:

- *cn* = *Friend*
- *objectClass* = *classSchema*
- *subClassOf* = *Person*
- *governsID* = 1.2.345.678901.2.3.45 (valid object identifier value)
- *mustContain* = givenName, sn
- *mayContain* = Address, phone-number

Modifying a Class

To modify a class, modify the existing class-definition object that represents the class. Some attributes of each class are designated as system-only, for consistency and security reasons. You cannot modify system-only attributes of a class-definition object, not even for new classes that you originally created. System-only attributes are designated by having the *systemOnly* attribute of the attribute set to TRUE.

The following attributes of a class-definition object are system-only attributes and, thus, cannot be modified:

- *governsID*
- *schemaIDGUID*
- *rDNAttID*
- *subClassOf*
- *systemMustContain*
- *systemMayContain*
- *systemPossSuperiors*
- *systemAuxiliaryClass*
- *objectClassCategory*
- *systemOnly*
- *objectClass*
- *instanceType*

System Checks and Restrictions Imposed on Schema Additions and Modifications

When you try to add or modify a class or attribute, Active Directory performs some checks 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
- 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 of the following tests as well as the normal extension checks.

- The value of *governsID* must be unique.
- All attributes that are defined in the *systemMayContain*, *mayContain*, *systemMustContain,* and *mustContain* lists must already exist.
- All classes that are defined in the *subClassOf*, *systemAuxiliaryClass*, *auxiliaryClass*, *systemPossSuperiors*, and *possSuperiors* lists must already exist.
- All classes in the s*ystemAuxiliaryClass* and *auxiliaryClass* lists must have either 88 class or Auxiliary class specified as their *objectClassCategory*.
- All classes in the *systemPossSuperiors* and *possSuperiors* lists must have either 88 class or Structural class specified as their *objectClassCategory*.
- Classes in the *subClassOf* list must follow certain X.500 specifications for inheritance hierarchies. These specifications are that Abstract classes can inherit only from other Abstract classes, Auxiliary classes cannot inherit from Structural classes, and Structural classes cannot inherit from Auxiliary classes.
- The attribute specified in the *rDNAttID* attribute must have Unicode-string as its syntax and be single-valued.

For attribute changes, the system also checks the following:

- The value of *attributeID* must be unique.
- The value of *mAPIID*, if any, must be unique.

- If *rangeLower* and *rangeUpper* are present, *rangeLower* must be smaller than *rangeUpper*.
- The values of *attributeSyntax* and *oMSyntax* must match, as shown in Table 4.11.
- If the attribute is object-syntaxed (*oMSyntax*=127), it must have the correct *oMObjectClass*, as shown in Table 4. 12.
- The *linkID*, if any, must be unique. In addition, a back link must have a corresponding forward link. (For more information about links, see "Active Directory Data Storage" in this book.)

Note A complete syntax specification consists of both the *attributeSyntax* and the *oMSyntax*. Hence, whenever more than one *oMSyntax* can be used with an *attributeSyntax*, the correct *oMSyntax* must be used.

Table 4.11 Values of *attributeSyntax* and Corresponding Syntaxes

***attributeSyntax* Value**[1]	**Matching *oMSyntax***
2.5.5.1	127 [Object(DN-Binary)]
2.5.5.2	6 [String(Object-Identifier)]
2.5.5.3	27 [String(Case sensitive)]
2.5.5.4	20 [String(Case insensitive)]
2.5.5.5	19 [String(Printable)], 22 [String(IA5)]
2.5.5.6	18 [String(Numeric)]
2.5.5.7	127 [Object(ORName)] or [Object(DNBinary)]. Distinction is oMObjectClass value.
2.5.5.8	1 [Boolean]
2.5.5.9	2 [Integer], 10 [Enumeration]
2.5.5.10	4 [String(Octet)]
2.5.5.11	23 [String(UTC-Time)], 24 [String(Generalized-Time)]
2.5.5.12	64 [String(Unicode)
2.5.5.13	127 [Object(Presentation-Address)]
2.5.5.14	127 [Object(Access-Point)] or [Object(DN-String)]. Distinction is oMObjectClass value
2.5.5.15	66 [String(NT-Sec-Desc)]
2.5.5.16	65 [LargeInteger)]
2.5.5.17	4 [String(Sid)]

[1]The *oMSyntax* names are specified with the syntax numbers to enable the correct choice.

For attributes with *oMSyntax*=127, the *oMObjectClass* also must be correctly specified according to the *attributeSyntax*. For attributes with any other *oMSyntax* value, it is not relevant and need not be specified. Because an *oMObjectClass*, being a binary value, is somewhat inconvenient to specify and because in most cases there is a one-to-one mapping between the *attributeSyntax* and *oMObjectClass*, the value defaults if none is specified by the user. There are a couple of cases where the mapping is not one-to-one, however, and the value defaults to the more common value. Table 4.12 is a list of the *oMObjectClass* values that correspond to the different *attributeSyntax* values for attributes with *oMSyntax*=127.

Table 4.12 Values of *attributeSyntax* and Corresponding *oMObjectClass* Values

***attribute*Syntax**	***oMObjectClass Values*[1]**
2.5.5.1	\x2B0C0287731C00854A [Object(DS-DN)].
2.5.5.7	\x56060102050B1D [Object(OR-Name)] or \x2A864886F7140101010B [Object(DN-Binary)].
	Defaults to Object(OR-Name) if none specified by the user.
2.5.5.13	\x2B0C0287731C00855C [Object(Presentation-Address)].
2.5.5.14	\x2B0C0287731C00853E [Object(Access-Point)] or \x2A864886F7140101010C [Object(DN-String)].
	Defaulted to Object(Access-Point) if none specified by the user.

[1]The syntax names are specified in brackets for easy reference.

Safety Checks

The purpose of the safety checks is to reduce the possibility of schema updates by one user or application breaking another application. These checks are necessary because multiple applications might share a schema definition.

When you modify existing schema objects, the modifications are subject to certain restrictions enforced by Active Directory. In some cases, these restrictions are determined according to whether the objects are part of the original schema or whether they have been added after the original installation. So the schema objects are really divided into two categories:

- Category 1 objects
- Category 2 objects

Category 1 objects are the default base schema objects that are included with Windows 2000 in the base schema. *Category 2 objects* are objects that are added subsequently to the schema by administrators or applications. You can determine the category in which an object is located by looking at the second bit (starting at the least significant bit) in the *systemFlags* attribute. If the bit is set, it has the value FLAG_SCHEMA_BASE_OBJECT, which indicates that the object is part of the base schema, that is, category 1. If this bit is not set or the attribute is not present, the object is category 2.

The following restrictions apply to both category 1 and category 2 schema objects:

- You cannot add a new *mustContain* attribute to a class either directly or through inheritance by adding an auxiliary class.
- You cannot add or delete any *mustContain* attribute of a class either directly or through inheritance.

The following restrictions apply to category 1 schema objects:

- You cannot change the *rangeLower* and *rangeUpper* of an attribute.
- You cannot change the *atributeSecurityGUID* of an attribute.
- You cannot deactivate a class or an attribute (make it defunct).
- You cannot change the *lDAPDisplayName* of a class or an attribute.
- You cannot rename a class or an attribute.
- You cannot change the *defaultObjectCategory* of a class.
- You cannot change the *objectClassCategory* of instances of a class.

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.

You might want to delete schema classes or attributes that are not needed in your organization. However, deleting schema objects raises some serious issues. For instance, what would happen to any other schema objects that use the class or attribute that you have deleted? Because doing an enterprise-wide check and cleanup might prove very time-consuming and costly, Active Directory does not support the actual deletion of schema objects. Rather it provides a mechanism for deactivating schema objects, also referred to as making them defunct. When you deactivate a schema object, you make it unusable for most purposes, and you get most of the benefits of deletion.

A class or an attribute can be deactivated by setting the Boolean attribute *isDefunct* to TRUE on the schema object. At any point in time, there are a number of ways to identify the defunct schema objects in the system. Programmatically, the user can search for all schema objects that have the attribute *isDefunct* set to TRUE (or if a particular schema object has *isDefunct* set to TRUE, to check whether the object is defunct). You can also use the Search function of the Ldp tool to search the schema with a filter set to (*isDefunct*=TRUE). For more information about the Ldp tool, see "Active Directory Diagnostics, Troubleshooting, and Recovery" in this book.

Note There is currently no method in the user interface for viewing defunct schema objects. To do this, you can use only one of the methods described in the preceding paragraph.

As with additions or modifications of classes or attributes, there are some special validation checks performed when a class or an attribute is made defunct. This is to ensure the consistency of the schema. In particular, on an attempt to make a class defunct, Active Directory verifies that the class is not used in the *subClassOf*, *auxiliaryClass*, or *possSuperiors* list of any existing nondefunct class. Similarly, on an attempt to make an attribute defunct, Active Directory checks that the attribute is not used in the *mustContain* or *mayContain* of any existing nondefunct class.

A defunct schema object can be resurrected, that is, made nondefunct again, by either removing the *isDefunct* attribute from the object or by setting the value of the *isDefunct* attribute to FALSE. This can also be done easily by using the Active Directory Schema console. Because making a defunct schema object nondefunct is similar to adding a new schema object as far as subsequent schema updates go, Active Directory performs the same validation checks here as it does on the addition of a new schema object.

▶ **To reactivate a class or attribute by using the Active Directory Schema console**

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**

1. Open ADSI Edit.
2. Right-click **ADSI Edit**, and then click **Connect to**.

   The **Connection** dialog box is displayed, as shown in Figure 4.3.

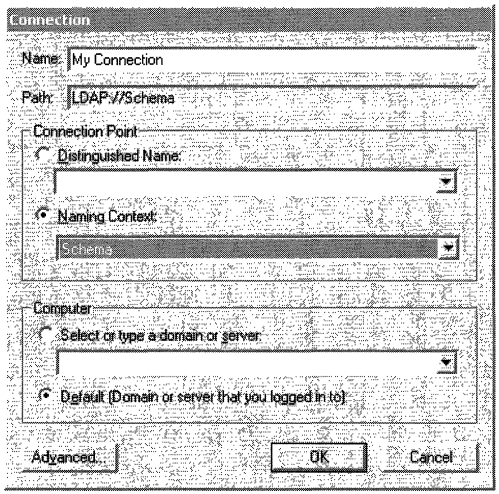


Figure 4.3 Connection Dialog Box

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**.

   The **Schema** folder is displayed.
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 in Figure 4.4), type:

   FALSE

10. Click **Set**, and then click **OK**.

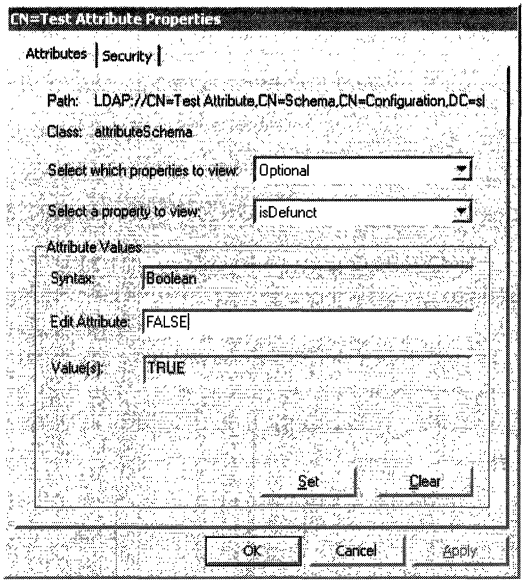


Figure 4.4 Test Attribute Properties Dialog Box

A schema object can be reactivated at any time. The only restriction imposed is that in any such modification, the *isDefunct* attribute is the only attribute present in the modify call. This is done to achieve clean semantics.

The only modification that is allowed on a defunct class or attribute is to modify the *isDefunct* attribute on it to make the class or attribute active again if this is necessary. No other modifications are allowed on a defunct class or attribute. The assumption is that because the object has been deactivated, it is not going to be used for any new modifications; so there is no need to modify it.

Disabling Existing Classes and Attributes

Disabling schema classes and attributes is subject to the following restrictions:

- You cannot disable a category 1 class or attribute.
- You cannot disable an attribute that is a member of a class that is not also disabled. This is because an attribute might be a "must have" for the (not disabled) class and disabling the attribute prevents new instances of the class from being created.

To disable an attribute, set the *isDefunct* attribute of its *attributeSchema* object to TRUE. When an attribute is disabled, new instances of the attribute can no longer be created. To re-enable the attribute, set the *isDefunct* attribute to FALSE.

To disable a class, set the *isDefunct* attribute of its *classSchema* object to TRUE. When a class is disabled, new instances of the class can no longer be created. To re-enable the class, set the *isDefunct* attribute to FALSE.

Effect of Deactivating a Schema Object on All Objects

After a class A is made defunct, any subsequent addition or modification of instances of A fails as if A has been deleted from the system; that is, the same error codes are returned as if A never existed at all. For example, creating a new instance of A fails and trying to modify or rename an existing instance of A fails. Similarly, if an attribute B is made defunct, B is treated as nonexistent for new object creations and attempts to modify (add or replace) the value of B in an existing object fail.

However, any search or deletion in an object behaves as if no schema objects have been made defunct, the only exception being that schema objects are not allowed to be deleted. So in the preceding example, the user still is able to search for all existing instances of A and delete them if necessary. Similarly, the user can search for all instances that have a value for the attribute B and delete B from such an existing object. This behavior is retained to allow the user to clean up if necessary after a schema object is made defunct. For example, the administrator can decide that a class is not needed anymore and make it defunct so that no one can use it for any modifications. The existing instances of the class can then be cleaned up by searching for all instances and deleting them. Active Directory does not perform any cleanup after a schema object is made defunct.

Similarly, an attribute can be made defunct, and all its uses can be cleaned up. Note that you can delete only the entire attribute from the object, not the values of the attribute. For example, in the preceding example, if B is a multivalue attribute and an object had more than one value for B, trying to delete a value of B from the object fails. This behavior is enforced because there is no reason not to delete the attribute totally when cleaning up a defunct object.

Effects of Deactivating a Schema Object on Schema Updates

In addition to the effects on the instances of the schema object, there are some additional effects on subsequent schema updates when you make a schema object defunct. The additional effects arise mostly because schema updates are subject to special validation checks to which nonschema object updates are not subjected. If a class A or attribute A is made defunct, subsequent schema updates show the following behaviors:

- No modifications are allowed on defunct classes or attributes. The only exception that is allowed is to modify the *isDefunct* attribute on a defunct class A to make the class active again if required. The assumption is that because the class or attribute is made defunct, it is not used for any new modifications. So there is not any need to modify it, except to make it active again if the administrator decides later that it is needed.
- Validation checks that are performed when you add a new class or attribute or modify an existing nondefunct class or attribute treat A as nonexistent. For example, if A is an attribute, trying to modify an existing nondefunct class B by adding *mayContain*=A fails because the validation checks that are performed at schema modification time fail as if A did not exist. Or if A is a class, trying to add a new class with *subClassOf*=A fails because A is treated as nonexistent by the validation checks performed during the addition of the class. The exception is when you try to add or modify a class or attribute to have the same distinguished name, object identifier, *lDAPDisplayName*, *mAPIID*, or *schemaIDGUID* as the defunct class A or attribute A; the operation fails. In this case, A is treated as a nondefunct schema object to ensure that schema consistency is not violated.

This ability to make schema objects defunct can be very useful in different ways in production environments. Schema objects that are no longer needed can be cleaned up by making them defunct. Then existing instances of those classes or attributes can be deleted if desired. At the same time, if the same schema object is found to be of use later, it can be brought back quickly by modifying the object by removing the *isDefunct* attribute on it. This also protects against the accidental removal of a schema object by mistake (by making it defunct). The operation can be reversed easily with no side effects. Note that because Active Directory does not do any cleanup after a schema object is made defunct, all instances of the schema object that was made defunct by mistake remain and become valid, normal objects when the defunct schema object is made active again.

Issues Related to Modifying the Schema

When you modify the schema, you must be aware of the implications and of the potential problems that can arise. There are three main issues involved with modifying the schema: replication, concurrency control, and handling invalid object instances.

Replication

Because the schema is replicated across all domain controllers in the forest, a schema update that is performed at one domain controller is guaranteed to be propagated throughout the forest. This guarantees a schema that is consistent forest-wide. However, because of replication latencies, there can be temporary inconsistencies.

For example, consider that a new class A is created at server X, and then an instance of this class (B) is created at the same server (X). However, when the changes are replicated to another server Y, the object B is replicated out before the *classSchema* object A. When the change arrives at server Y, the replication of B fails because Y's copy of the schema still does not contain the *classSchema* object A. Hence, Y does not know about the existence of A.

Active Directory solves this problem in such scenarios by explicitly replicating the Schema container from the originating server when such failures occur. Additionally, the replication of the Schema container triggers an immediate schema cache update on the target server. Active Directory then re-replicates the object that failed. In the example, re-replication brings in *classdefinition* object A and also puts it into the schema cache of Y. Retrying the replication of B now succeeds.

Concurrency Control

Active Directory must ensure that different program threads do not perform simultaneous, conflicting schema updates (such as when one thread is deleting an attribute and another is adding it to the *mayContain* list of a class).

To ensure this, any thread that attempts to perform a schema update also automatically writes a special attribute on the Schema container as part of the transaction. (Active Directory automatically causes the thread to write the attribute you do not have to do so in your program code.) Only one thread can write this attribute at any one time. This method guarantees schema consistency, but it does not guarantee which of the updates is successful. You must be aware of this when schema updates are made in a batch (such as in the case of the installation of directory-enabled applications).

For example, consider a scenario in which two Active Directory–aware programs, A and B, are being installed simultaneously, each of which creates several new schema objects. Because Active Directory creates one thread per object update, it is possible that some of the objects in program A and some of the objects in program B get created (if the internal threads do not overlap), and then one of the installations fails (because a thread for a schema object creation for program A overlaps with a thread for a schema object creation for program B).

Assume that program A fails. Now running A from scratch again does not work because some of the objects that program A created are already in the schema; trying to re-create them in the second run (existing objects) returns an error. Therefore, it is recommended that programs that modify the schema not be run concurrently, unless provisions are made in the program to first check if the schema update that is about to be made has already been made and then proceed accordingly.

Handling Invalid Object Instances

Schema update can make an existing instance of an object invalid. For example, suppose object X is an instance of class Y. Class Y has an attribute, Z, in its *mayContain* list. Therefore, because object X is an instance of class Y, object X can have this attribute defined on it. Assume that X does indeed have this attribute currently defined in it. Now a schema update is performed that modifies class Y by deactivating attribute Z from its *mayContain* list. Note that this change makes the instance of object X invalid because X now has an attribute, Z, that it is not allowed to have according to the class definition of Y (of which object X is an instance). Active Directory allows the now-invalid objects to remain in the directory and ensures that they do not cause any problems in the rest of the schema. Active Directory does not automatically clean up invalid objects, but invalid objects and attributes appear in searches and can be deactivated manually.

Methods for Extending the Schema

Windows 2000 gives you some choices regarding the way you accomplish schema extension. 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 large numbers of objects (such as users, contacts, groups, servers, and printers) in one operation. By using these tools, it is possible to export Active Directory data to other applications and services and to import information from other sources into Active Directory. These tools are installed automatically on all Windows 2000 servers. Or you can perform schema extension programmatically by using ADSI Edit. You can also extend the schema from the user interface with the Active Directory Schema console.

LDAP Data Interchange Format

The LDAP Data Interchange Format (LDIF) (file) format has a command-line utility called "LDIFDE" 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 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 that are 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.

Using the LDIFDE Tool

The LDIFDE tool is executed from the command prompt. At the prompt, type the command **LDIFDE** and the appropriate parameters in the following form:

LDIFDE [**-i**] [**-f**] [**-s**] [**-c**] [**-v**] [**-t**] [**-d**] [**-r**] [**-p**] [**-l**] [**-o**] [**-m**] [**-n**] [**-j**] [**-g**] [**-k**] [**-a**] [**-b**][**-?**][**-u**][-**y**]

Note A hyphen (-) is required before each parameter.

Table 4.13, Table 4.14, Table 4.15, and Table 4.16 contain descriptions of all of the parameters.

Table 4.13 LDIFDE Tool Basic Parameters

Basic Parameters	Value(s) to Specify	Description
-i	mode	Specifies import mode. If this parameter is not specified, the default mode for LDIFDE (and CSVDE) is export.
-f	filename	Identifies the import or export file name.
-s	server name	Specifies the domain controller to perform the import or export operation. If this parameter is not specified, the operation communicates with the domain controller of the domain to which the user is currently logged on.
-c	from distinguished name (string1) to distinguished name (string2)	Replaces all occurrences of string1 with string2. This usually is used when you are importing data from one domain to another and the distinguished name of the export domain has to be replaced with that of the import domain. This parameter is designed to support the import of data when the receiving domain name is different than the exporting domain name.
-t	port number	Specifies a port number. The default LDAP port is 389. (The Global Catalog port is 3268.)
-v	verbose mode	Sets verbose mode, which provides more detailed status description of the import/export operation. If this parameter is not specified, the default is nonverbose mode.
-?	Help	Use to display Help.

Table 4.14 LDIFDE Tool Export-specific Parameters

Export-specific Parameters	Value(s) to Specify	Description
-d	base distinguished name	Sets the distinguished name of the search base for data export. If this parameter is not specified, it defaults to the root of the domain.
-r	LDAP filter	Creates an LDAP search filter for data export. For example, to export all users with your surname, the following filter could be used: -r "(&(objectClass=user)(sn=yoursurname))". Note that the default is (objectClass=*). For more information about LDAP search filters, see "Name Resolution in Active Directory" in this book.
-p	scope	Sets the search scope. Values are: Base, OneLevel, or SubTree. If not specified, the default is SubTree. For more information about the search scope, see "Name Resolution in Active Directory" in this book.
-l	LDAP attribute list	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, the following attribute list would be specified: -l "distinguishedName, cn, givenName, sn, telephone". (Note: Quotation marks around the list of attributes is optional.) *(Continued)*

Table 4.24 LDIFDE Tool Export-specific Parameters *(Continued)*

Export-specific Parameters	Value(s) to Specify	Description
-o	attributes in results	Omits a list of attributes from the results of an export query. This is used when exporting objects from Active Directory and then importing them into another LDAP-compliant directory. Some attributes might not be supported in the directory receiving the objects. For example, to omit *whenCreated* and *whenChanged*, the following would be specified: -o "whenChanged, whenCreated". This parameter omits the attributes from the results. If not specified, all attributes are included (Note: Quotation marks around the list of attributes is optional.)
-m	Active Directory attributes	Omits attributes that apply to only Active Directory objects such as *objectGUID* (globally unique identifier), *objectSID* (security identifier), *pwdLastSet* (password last set), and *samAccountType*. By default, this parameter is disabled. Its primary purpose is to export entries in preparation of re-importing them into Active Directory. This also activates the linked attribute option, which appends to the end of the file the values for attributes that are linked to the current object. For example, a parent object has linked attributes to a child object, and those entries are placed at the end of the file. (Note: Some attributes are read-only for Active Directory and, thus, cannot be re-imported. The -m option strips these attributes out at the time of export to prepare the file for re-importing.)
-n	binary values	Specifies not to export binary values. By default, this parameter is disabled.

(Continued)

Table 4.34 LDIFDE Tool Export-specific Parameters *(Continued)*

Export-specific Parameters	Value(s) to Specify	Description
-j	directory path	Sets the log file location. The default is the current directory.
-g	paged searches	Specifies not to perform paged searches. If not specified, it performs paged searches. Note that some servers might not support the paged search control.
-u	Unicode	Enables Unicode support. When this parameter is used during an export operation, a Unicode file is generated. When the parameter is used during an import operation, the tool would expect a Unicode file as input.
-y		Enables lazy commit to the directory.

Table 4.15 LDIFDE Tool Import-specific Parameters

Import-specific Parameters	Value(s) to Specify	Description
-k	action if errors are encountered	Skips errors during the import operation and continues processing. If not specified, the import operation stops if it encounters the following errors: LDAP_ALREADY_EXISTS LDAP_CONSTRAINT_VIOLATION LDAP_ATTRIBUTE_OR_VALUE_EXISTS LDAP_NO_SUCH_OBJECT ERROR_MEMBER_IN_ALIAS It will also skip objects with no attributes. For more information about these errors, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Table 4.16 LDIFDE Tool Credentials Parameters

Credentials Parameters	Value(s) to Specify	Description
-a	user distinguished name password OR *	Sets the command to run by using the supplied user distinguished name and password. The default is to run by using the credentials of the currently logged on user. For example, -a "cn=yourname,dc=yourcompany,dc=com password". * = option to hide password
-b	username domain password OR *	Sets the command to run as the username domain password. The default is to run using the credentials of the currently logged on user. * = option to hide password

Note Make sure that all required attributes exist when you create or modify objects. For example, the required attributes for creating a user are *distinguishedName*, *objectClass*, and *samAccountName*.

Exporting and Re-Importing Objects

Linked attributes contain information about the links to a current object. During a normal export session, a parent object might be exported before its child object. On the re-import operation, if the parent object has been added before the child object, the operation fails because the child object is not yet in the directory.

However, when the -m parameter is used to export objects and re-import them into Active Directory, all entries that contain a linked attribute are appended to the end of the file. Moreover, the linked addition is separated from the main object creation call so that the failure in membership addition does not cause the object creation to fail. The linked attribute is appended to the end of the file.

Read-only Properties on Objects

Active Directory has Security Accounts Manager (SAM) properties that are read-only because they are set by the system at the time the object is created. When the -m parameter is used to export objects and re-import them into Active Directory, all of the SAM attributes are ignored during the export operation. In that way, when the entries are re-imported into Active Directory, they succeed because they do not contain any SAM information.

Example of an LDIF Import File

In the following example of an LDIF import file format, you also can see how to add a *user* object to the myDomain.microsoft.com domain:

```
dn: CN=sampleUser,CN=Users,DC=myDomain,DC=microsoft,DC=com
changetype: add
cn: sampleUser
description: Example of an Imported User using LDIFDE
objectClass: user
sAMAccountName: sampleUser
```

The following is an example of the command that is used to import the file in the preceding example:

```
ldifde -i -f import.ldf -v
```

Manipulating Data in an LDIF Export File

The preferred method of manipulating the distinguished name (*distinguishedName*) during an LDIFDE export operation is to use the –c parameter. For example, by using this parameter in conjunction with the –m parameter, you can import a large group of users from one domain into another domain.

Note You must use a text editor to make substantial changes to attribute values in your export file prior to import.

Comma-Separated Value File Format

The bulk import and export of data to and from Active Directory can be performed by using files that store data in the Microsoft comma-separated value (CSV) file format, also known as a .csv file. The CSV file format is supported by many other applications, such as Microsoft® Excel, that can read and save data in the CSV file format. Also, Microsoft® Exchange Server administration tools can import and export data by using the CSV format. The CSV format has a command-line utility called "CSVDE" that allows you *only* to add new objects. It can be run on a Windows 2000–based server or copied to a Windows 2000–based workstation.

The CSV format consists of a simple text file with one or more lines of data where each value is separated by a comma. The text file contains entries where the initial entry is a comma-separated list of attribute names. Each subsequent entry in the text file represents a single object in the directory. Attribute values are delimited by commas.

Using the CSVDE Tool

The CSVDE tool is executed from the command prompt. At the command prompt, type the command **CSVDE**. The parameters that are used for the CSVDE tool are the same as those that are used for the LDIFDE tool. However, unlike the LDIFDE tool, CSVDE creates files that can be read from applications other than LDAP servers. For example, if you want to view all Active Directory users in a Excel report, CSVDE is used to export the directory data into the .csv file format, which could then be read by Excel.

The CSVDE tool is executed from the command prompt. At the prompt, type the command **CSVDE** and the appropriate parameters in the following form:

CSVDE **[-i] [-f] [-s] [-c] [-v] [-t] [-d] [-r] [-p] [-l] [-o] [-m] [-n] [-e] [-j] [-g] [-k] [-a] [-b] [-?][-u]**

The descriptions of these parameters are contained in Tables 4.13, 4.14, 4.15, and 4.16. As for the LDIFDE tool, the default mode for CSVDE is Export, unless otherwise specified by using the -i parameter for the import mode.

Note CSVDE cannot be used to modify or delete objects. It can be used only to *add* directory objects. A hyphen (-) is required before each parameter.

Viewing Data in the .csv File

When you view data in the .csv file, the values for multivalue attributes are expressed as a single value that is internally delimited by a second user-definable delimiting character (by default, $). Attribute values are listed left to right in the order in which the attribute names are listed in the initial entry. Values are positional, and every entry must account for each attribute listing in the initial entry. The attribute names must be in the same order as the data in any line that follows the first line, as shown in the following example:

```
dn,cn,firstName,surname,description,objectClass,sAMAccountname
"cn=John Smith,cn=Users,dc=myDomain,dc=microsoft,dc=com",John
Smith,John,Smith,Manager,user,jsmith
"cn=Jane Smith,cn=Users,dc=myDomain,dc=microsoft,dc=com",Jane
Smith,Jane,Smith,President,user,janes
```

Each object stands alone and does not need the context of another object to be listed in the .csv file, which simplifies the reading and writing of files and allows objects from different classes to be contained in a single file.

Another example shows the .csv file format and lists the header, which contains the LDAP display names of the properties ("attributes")—distinguished name, object class, common name, given name, surname, telephone number, street address, locale, country/region, and sAMAccountName.

```
dn,objectClass,cn,givenName,sn,telephoneNumber,
street,l,c,sAMAccountName
"cn=James Smith,cn=Users,dc=myDomain,dc=microsoft,dc=com",user,James
Smith,James,Smith, ,203-223-2233, 15 Woodbine St.,Fenwick,US,jsmith
```

All data values are represented as strings. Numeric values are represented by numeric strings; binary values are represented by hexadecimal strings. Hexadecimal strings start with the character "x," followed by a single quotation mark ('), then the hexadecimal string, and, finally, another single quotation mark ('). The following is an example of a hexadecimal string:

```
X'0105000000000005150000079e3fc535729024c235f636bf5010000'
```

Syntax information is stored in the schema of the destination directory. Programs that accept imported .csv files determine how to process the values by using the schema in the target directory.

A missing or unsupported attribute value has an empty position in the string. For example, if the third attribute value for an entry is missing, it would be expressed as follows:

```
firstvalue,secondvalue,,fourthvalue
```

Multivalue attributes are separated by semicolons (;). For example if there are three attributes and the second one is a multivalue attribute, it would be expressed as follows:

```
1stvalue,2ndvalue1;2ndvalue2;2ndvalue3,3rd value
```

Reserved characters that appear in string properties are represented through an escape mechanism. The following are reserved characters:

- Backslash (\)
- Semicolon (;)
- Special character for hexadecimal representation (x')

The escape mechanism uses a backslash (\) before a reserved character as an escape character. If a value contains a backslash, the backslash in the value also must be preceded by the escape character—that is, by another backslash (for example, \\). The semicolon (;) character is used to delimit multivalues. If the value itself contains a semicolon, the semicolon in the value must be preceded by the escape character (for example, \;). The hexadecimal prefix (x') character, if used in a value, must also be preceded by an escape character (for example, \x').

There are two other characters that must be handled in a special way. They are the comma (,) and the double quotation marks (") characters. The comma (,) is treated as a special character in the CSV format because it is used to separate values. If the value contains a comma (,), the format specifies that the comma has to be enclosed by double quotation marks (for example, value1,value2 are represented as "value1,value2"). The double quotation marks (") character is used to contain values if the values contain commas (,). When a value contains a pair of double quotation marks as well as a comma, the quotation marks in the value have to be enclosed with another set of double quotation marks, as follows:

```
"value1","value2" is represented as ""value1"",""value2""
value1"value2 is represented as "value1""value2"
```

The following CSV file shows an example of adding an organizational unit, followed by a user, and a computer:

```
dn,cn,givenName,sn,description,objectClass,sAMAccountname
"ou=sampleOU,dc=myDomain,dc=microsoft,dc=com",,,,Sample
Organizational Unit,organizationalUnit,
"cn=John Smith,ou=sampleOU,dc=myDomain,dc=microsoft,dc=com",John
Smith,John,Smith,Sample User,user,jsmith
"cn=sampleComputer,ou=sampleOU,dc=myDomain,dc=microsoft,dc=com",sampl
eComputer,,,Sample Computer,computer,computer1
```

Note Both ANSI text and UNICODE are supported.

Using LDIFDE and CSVDE to Modify the Schema

LDIFDE and CSVDE use files that contain directory data in the appropriate format. These files can be imported or exported to LDAP-based directory servers as a means of populating a directory or modifying objects in a directory. Because the Active Directory schema exists as a collection of directory objects, either of these tools can be used to extend the schema.

Note At present, CSVDE can be used only for additions to the directory, not for modifications to the directory.

LDIF File Format

The LDIF file format can be used to perform batch operations on directories that conform to LDAP standards. It is suitable for additions to the directory as well as modifications and deletions of directory objects. A *record* in an LDIF file consists of a sequence of lines that either describe a directory entry or a set of changes to a single directory. This format can be used for all LDAP operations.

The preferred method when modifying the schema is to use the Active Directory Schema console to edit the schema on a practice system that is isolated from your real enterprise. You can use the LDIFDE export to produce a script, which you can then run against your live system. The following example represents the contents of a sample LDIF file that can be used to add a new attribute to Active Directory.

```
dn: CN=New-Attribute-Name,CN=Schema,CN=Configuration,
DC=microsoft,DC=com
changetype: add
objectClass: attributeSchema
ldapDisplayName: newAttributeName
adminDisplayName: New-Attribute-Name
adminDescription: New-Attribute-Name
attributeId: 1.2.840.113556.1.4.8000.1 <- the id has to be unique
attributeSyntax: 2.5.5.12
omSyntax: 64
isSingleValued: TRUE
systemOnly: FALSE
searchFlags: 0
showInAdvancedViewOnly: TRUE
```

The following example shows an LDIF file that can be used to force a schema cache update.

```
dn:
changeType: modify
add: schemaUpdateNow
schemaUpdateNow: 1
```

For more information about the LDIF format and using LDIFDE, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

CSV File Format

The CSV file format is a simple format whose primary benefit is ease of use. In the CSV file format, each line represents a discrete object in the directory, with the object's attributes separated by commas. The first line of the file always contains all of the attribute names. Each subsequent line represents a different entry in the directory. Values for multivalue attributes can also be specified and are delimited by the semicolon (;) character.

Because this format is compatible with the Microsoft Excel CSV format, you can dump directory information to an Excel spreadsheet or import data from a spreadsheet into Active Directory. This format can be used only for additions to the directory. The following example represents the contents of a CSV file that can be used to add a user to Active Directory:

```
dn,objectClass,cn,sn,givenName,telephoneNumber,street,l,c,sAMAccountName
"CN=John Doe,DC=myDomain,DC=microsoft,DC=com",
User,John Doe,Doe,John, 555-456-7890,123 Magnolia Ave.,Redmond,US,jdoe
```

Both the CSVDE and LDIFDE tools have command line help that can be viewed by typing the command name at an MS-DOS® prompt. Because both of these tools allow data to be imported and exported, there are a number of different uses for them.

Migration to Active Directory By using either of these tools, users of other directory services can import data to Active Directory. This works for any directory that is LDAP-compatible as long as the attribute names match.

Publishing Information from the Directory You can use either of these tools to export directory data to another application that can read either the LDIF or CSV format. You can also export to other LDAP-compatible directory services, provided there are matching attribute names.

Adding Resources to the Directory In addition to the Active Directory Users and Computer console and ADSI Edit, administrators can choose to use one of these tools to add objects to a directory. These tools lie somewhere between the other options in terms of ease of use and flexibility. Because the schema is represented in Active Directory as directory objects, you can use LDIFDE or CSVDE to extend the schema with new or modified schema objects. In fact, if your application requires schema modifications, the best way to accomplish this is to distribute an LDIF or CSV file with the application that is to be imported to the schema.

Figure 4.5 illustrates one way in which LDIF can be used to extend Active Directory.

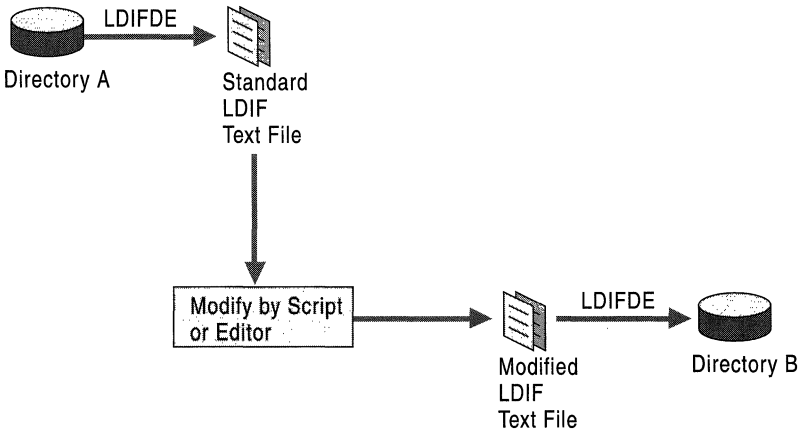


Figure 4.5 Extending Active Directory with LDIF

Using Active Directory Service Interfaces and Visual Basic Scripts

Although one potential benefit of using an LDIF or CSV file is that the administrator can look at it to see what it does, consider the merits of extending the schema programmatically:

- A programmatic extension is invariant; it is a Windows executable file. The binary cannot be tampered with, unlike an LDIF or a CSV file, either of which can be modified inadvertently or maliciously.
- Programs can detect and recover from errors and provide intelligent feedback.
- Programs handle Unicode without resorting to Base64 encoding. (Unicode is a 16-bit character set that contains all of the characters commonly used in information processing.)
- Programs can use the Windows Installer (.msi) setup APIs.
- Programs can be signed to prove their authenticity.

Active Directory provides a set of interfaces that enable you to gain access to directory objects, including schema objects, programmatically. ADSI defines a directory service model and a set of COM interfaces that you can easily use with a variety of programming languages. ADSI conforms to the Component Object Model and supports standard COM features.

By using Microsoft® Visual Basic® Script and ADSI, you can write scripts easily to accomplish various directory modifications, including extending the schema.

These are the specific ADSI interfaces to use when you extend the schema:

IADsContainer Use **IADsContainer::Create** to create new *classSchema* and *attributeSchema* objects.

IADs Use **IADs::Get (or GetEx)** to read the attributes of *classSchema* and *attributeSchema* objects. Use **IADs::Put** (or **PutEx**) to set the attributes of *classSchema* and *attributeSchema* objects. PutEx is particularly useful in manipulating the lists of *mustContain* and *mightContain* attributes because it is designed specifically for handling multivalue attributes.

The code in the example that follows represents a script that you can use to add a user to Active Directory.

```
Dim oDomain
Dim oUser

Set oDomain=GetObject("LDAP://OU=Marketing,DC=antipodes,DC=com")
Set oUser = oDomain.Create("user","cn=John Smith")
oUser.Put "samAccountName","JSmith"
oUser.Put "givenName","John"
oUser.Put "sn","Smith"
oUser.Put "userPrincipalName","jsmith@antipodes.com"
oUser.SetInfo
MsgBox "User created " & oUser.Name
Set oDomain = Nothing
MsgBox "Finished"
WScript.Quit
```

Note For more information on ADSI and ADSI interfaces, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Using the Active Directory Schema Console

The Active Directory Schema console allows members of the Schema Administrators group to manage the schema through a graphical interface. With it, you can create and modify classes and attributes and also specify what attributes are indexed and what attributes are replicated to the Global Catalog. After you start the Active Directory Schema console, the first thing that you must do is to make sure that the tool is focused on the schema master for your enterprise.

Note The Schema Management snap-in is not one of the default MMC snap-ins that is provided with Windows 2000 Server. To make it appear in the list of available snap-ins, you must install the admin tools package (Adminpak.msi). To register the Schema Management snap-in, open your *%systemroot%*\System32 folder and run **Regsvr32** Schmmgmt.dll from the command prompt or from the **Run** command on the **Start** menu.

▶ **To view or change the current schema master by using the Active Directory Schema console**

1. Open MMC, and then install the Active Directory Schema snap-in.
2. Right-click **Active Directory Schema**, and then click **Operations Master**.
3. The **Current Operations Master** that is displayed is the schema master. Click **OK** to leave it as is.

   –Or–

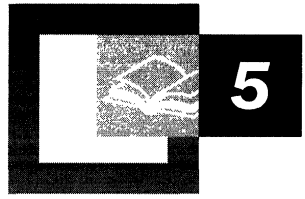
   Click **Change** to change the server that is the current FSMO Role Owner for the schema. If the current domain controller (the one that is listed in **Current Focus**) is also the current operations master, you must use the Active Directory Tree console in MMC to focus on another domain controller before you can change the operations master.

After you have verified that the tool is focused on the current schema master, you can use it to add, modify, or deactivate attributes and classes. Remember that the account you are using must be a member of the Schema Administrators group and that the server must be set to allow schema modifications.

Note Schema objects that are part of the default base schema cannot be deactivated.

CHAPTER 5

Service Publication in Active Directory



Active Directory™, the directory service included with Microsoft® Windows® 2000, is designed as a distributed network data store for information about computers, users, services, and applications. Directory-enabled services and applications can publish globally useful information, such as service availability and properties, in Active Directory. Active Directory management and user interfaces enable administrators and client processes to find and connect to the directory-enabled service as needed.

Anyone responsible for making resources available throughout a network needs to know the concepts and procedures involved with publishing services of Active Directory. Knowledge of the architecture of Active Directory service publication and of the base service publications provided by Active Directory is necessary for understanding the technology of service publication as it is used to manage services in a distributed network.

General knowledge of the architecture of Active Directory is required to understand service publication.

In This Chapter

Introduction to Service Publication

A service is a server process that performs a specific system function and often provides an application programming interface (API) for other processes to call. A server process provides one or more threads that accepts requests from client processes. It implements a set of services that it makes available to clients running one or more computers in a distributed network. Although Windows 2000 services are enabled for remote procedure call (RPC), meaning that their API routines can be called from remote computers, they do not require RPC.

Service publication is the creation, storage, and maintenance of information in the Active Directory data store. Network clients and network administrators can use the information stored in Active Directory to find, connect to, and manage a service. In addition, Active Directory enables clients and administrators to view the distributed network as a collection of services rather than as a collection of individual computers.

Types of Service Information

Active Directory can store information that includes data describing the methods of service instantiation and client binding, a critical part of which is binding information. Binding information is information a client uses to connect, or bind, to a service that is available on a network. Binding information encompasses a wide range of possible types of data.

Directory-enabled services can store in Active Directory the type of information that is listed in Table 5.1.

Table 5.1 Types of Service Information That Directory-Enabled Services Can Store

Service Information in Active Directory	Description
client bindings	Service name and connection methods used by the clients of a service to access the service.
administrative bindings	Service name and connection methods used by administrative programs to connect to a service for administrative operations. A single binding can be used for both client and administrative functions.
configuration data	Persistent configuration data about a service can be stored in the directory to use the security and replication capabilities of Active Directory. For example, a database service could store its default configuration for database servers in Active Directory. When a new instance of the database service is installed, the configuration information can be dynamically accessed to simplify installation and ensure consistent configuration.
other service-specific data	Service-specific extensions to the Active Directory schema and object classes that are useful client or administrative information.

Service Objects

An object is a distinct, named set of attributes that represents, for example, a user, a printer, or a server application. Active Directory objects represent those things that customers say are the most heavily accessed directory service objects that are found in network environments. Active Directory includes specifications for the following types of objects: user, group, directory service container, print management, schema, and service management.

For systems running Microsoft® Windows® NT version 4.0 and earlier, client processes and administrative programs require the computer name or the Internet Protocol (IP) address of the computer on which the service exists. The name or address is needed to find and connect to a service. In contrast, Active Directory enables client processes and administrative programs to connect to a service by using the *keywords* attribute, which enables the client to find the dnsHostName.

For more information on keywords, see "Finding and Viewing Service Information in Active Directory" later in this chapter. For more information on dnsHostName see "Name Resolution In Active Directory" in this book.

Active Directory enables you to find and connect to a directory-enabled service by using attributes other than the computer name or the IP address of the computer on which the service exists. Active Directory uses other object attributes, such as the service display name, commonly called the "friendly" name, to find services like Dynamic Host Configuration Protocol (DHCP). The structure of the Active Directory database and specific service objects are described in more detail in "Directory Infrastructure for Service Publication" later in this chapter.

Service Bindings

To publish a service in Active Directory, a directory-enabled service must store, as a minimum requirement, its binding information. Service bindings are the information a client uses to connect, or bind, to an instance of a given service. The information needed to bind to a service includes the service name and its location. For example, a World Wide Web browser binds to a Web server by using a Uniform Resource Locator (URL).

Table 5.2 lists examples of service bindings.

Table 5.2 Examples of Service Bindings

Service	Binding
File Service	Universal Naming Convention (UNC) name for a share
	Example: \\MyServer\MyshareName
Web Service	URL
	Example: http://www.reskit.com
RPC Service	RPC binding, encoded information used to connect to the RPC server. RPC bindings can be converted to and from strings with the RPC APIs
	Example: ncacn_ip_tcp:server.microsoft.com

Service Instantiation

A specific service can publish itself one or more times in Active Directory. Each instance of the service running on a computer or computers on the network can create connection point objects in Active Directory. A connection point represents one or more instances of a service that is available in a network.

For example, if a service such as Microsoft Certificate Services for Windows 2000 is installed and running on two computers on the network, there can be two connection point objects—one for the instance of the service on each computer. Similarly, a service with multiple instances installed on a single computer can create separate connection point objects for each instance. It is also possible for multiple instances of a replicated service to publish themselves using a single connection point in Active Directory. In this case, the connection point contains information that enables a client to select and bind to a replica.

Note The *possSuperiors* attribute and *systemPossSuperiors* attribute of each Active Directory object class contain a list of the object classes that can contain instances of the object—that is, the types of containers in which you can create the object. After creation of the class, the *systemPossSuperiors* attribute cannot be modified, and for an existing classSchema object, values can be added to the possSuperiors attribute, but they cannot be removed. For example, you can create an instance of the **serviceConnectionPoint** object class as a child of a **Computer**, **Container**, or **Organization-Unit** object. For more information on the *possSuperiors* attribute and *systemPossSuperiors* attribute, see "Active Directory Schema" in this book.

Directory Infrastructure for Service Publication

The objects stored in Active Directory are arranged in a hierarchy. Active Directory is shipped with a pre-configured hierarchy that contains the information required to install and run Windows 2000 and the directory service itself. The default structure includes containers intended specifically for use in service publication.

Active Directory has a default structure. Figure 5.1 illustrates this structure.

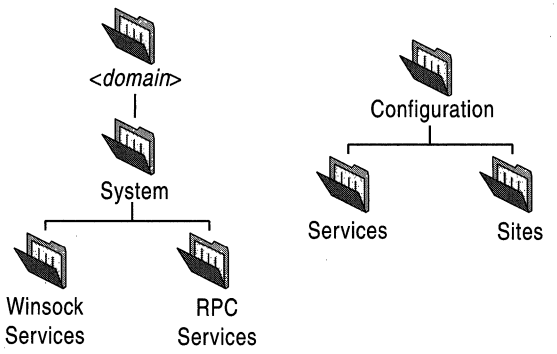


Figure 5.1 Default Active Directory Structure for Service Publication

The *domain hierarchy* stores all of the objects (for example, users, computers, and printers) related to a particular domain, and it is replicated to all domain controllers for that domain. The *configuration hierarchy* stores all of the objects related to a complete *forest* of domains. The configuration hierarchy is replicated to *every* domain controller in the forest of domains. The Services container and Sites container are immediate children of the configuration directory partition.

Connection Points

A **Connection Point** object represents one or more instances of a service that is available in a network. The Active Directory schema defines a variety of object classes for use in service publication. All objects representing resources that accept connections are derived from the object class **Connection Point**.

Figure 5.2 shows the hierarchy of the **Connection Point** class.

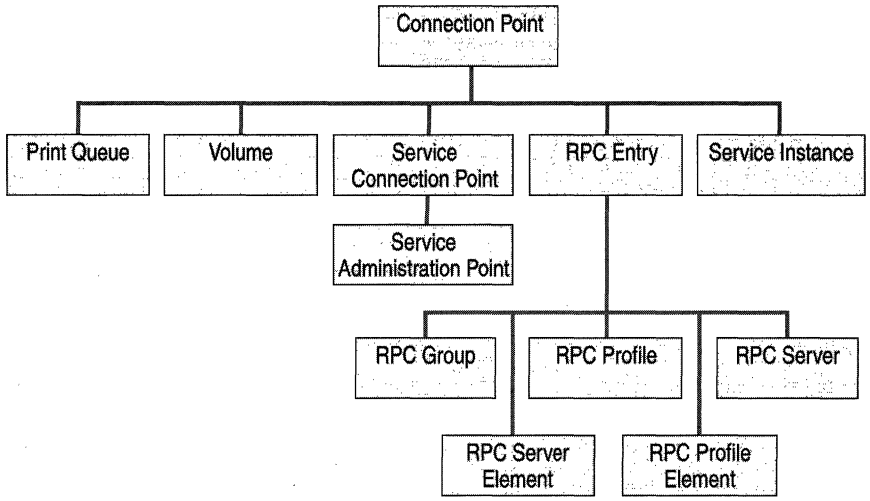


Figure 5.2 Connection Point Class Hierarchy

Some examples of objects that are derived from **Connection Point** and that accept connections are as follows:

Print Queue and Volume The **Print Queue** and **Volume** objects are the standard connection points used by print and file services respectively.

RPC Entry Points Windows 2000 supports two sets of networking APIs that provide transport independence: the Microsoft® Windows® Sockets interface and the RPC interface. RPC provides a mechanism for calling functions in other processes, even processes running on remote computers on a network. RPC provides a threading model, provides an endpoint (socket/pipe/port) mapping service, and connects to a name service. Services that currently publish themselves by using the RPC Name Service (RpcNs) APIs are published in Active Directory using the **RPC-Entry** object and other RPC object classes.

Service Instance Windows Sockets services that publish themselves using the registration and resolution (RnR) APIs are published in Active Directory using the **Service-Instance** object class.

Service Connection Point Service Connection Point is used by services that need to explicitly publish themselves in Active Directory, rather than use an existing abstraction, such as RPC Name Service or RnR.

A service that uses Service-Connection-Point needs to provide an abstraction layer to hide service location details from client applications. This abstraction can be implemented as a dynamic-link library (DLL), or as part of the client application. The abstraction queries Active Directory for a connection point object representing the service requested by the client application and uses the binding information from that object to connect the client application to the service.

A client application queries Active Directory for connection point objects that represent the services that the application wants to use. The client then selects one of those objects and uses the binding information from that object to connect to the service. In the case of RpcNs and RnR, the query is performed by the abstraction layer. In the case of the **serviceConnectionPoints** class, the query is performed directly by the client.

Note COM-based services do not use **Connection Point** objects to advertise themselves. These services are published in the *class store.* The Windows 2000 class store is a directory-based repository for all applications, interfaces, and APIs that provides for the publishing and assigning of applications.

For information about COM-based services and Windows 2000 Class Store, see the MSDN™ link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Where to Publish

To know where to publish services in Active Directory, you first must understand and observe the following guidelines:

- A service needs to publish its information in the domain hierarchy, and *never* in the configuration hierarchy.
- A service needs to create service-specific objects in the same domain as the computer running the service, and in locations that are convenient for the administration of the service and maintenance of those objects. The recommended location is the computer object on which the service is installed.

If a client is able to access a computer running a service, that client should be able to access a replica of the domain containing that computer.

Note The client can rarely identify the computer on which the service is running. The client normally finds the service from the *keyword* attribute The *keyword* attribute may contain the GUID that is specific to the vendor and application. However, the service that publishes the information must publish the *keywords* attribute.
For more information on the *keywords* attribute, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Therefore, it is sufficient for a service to publish information to the domain that contains the computer running the service. Publishing service information to the configuration hierarchy does not make the service more available or easier to access. It does, however, cause extra replication traffic. To maximize efficiency, do not publish service information to the configuration partition.

Clients find service objects by querying the directory. The client can scope the query to its domain, or it can search the entire forest using the Global Catalog. In either case, the specific location of the service object is irrelevant to the client. Therefore, the client does not influence the location to which to publish service objects. Instead, the key consideration is the ease with which the service can be administered.

In the domain hierarchy of Active Directory, service-related objects occur in three places, which are the following containers:

- Computer
- Organizational-Unit
- System or one of its children

Note In Active Directory (and many other directory services), *any* object can be named as a valid parent, defining it as a container. This allows a directory to have an intuitive structure. For example, objects that relate to exactly one computer should appear in the directory as children of that computer, where an administrator expects to find them.

Computer Object

When you publish services for a computer, create service-specific objects as children of the **Computer** object where the service is installed.

For ease of installation, many services use a "default" location in which to create their objects, but most services allow their objects to be moved after creation. The **Computer** object where the service is installed is a convenient and intuitive location for these objects, because when a **Computer** object is moved to a different location, Active Directory also moves the service objects to the new location. Also, when a **Computer** object is deleted or joins another domain, the service objects are also deteled. Therefore, there is less likeliness of having an *orphan* service. A process running on a computer that is a domain member can always find the computer object in the directory

---

Note Only computers that are domain members have valid **Computer** objects in the directory.

Organizational Unit Container Hierarchy

Active Directory allows you to create a container hierarchy that meets the needs of your organization. The object class for building these hierarchies is **organizationalUnit**. It is a general purpose container that you can use to group together most object classes, including computer objects, for administrative purposes. You might want to also group service-specific objects together for administrative purposes. Active Directory enables you to set permissions on all of the objects in a container or a subtree with a single ACL. Windows 2000 uses a permissions model that allows you to apply very fine-grained access control to objects in the directory and to delegate administrative authority to other users.

All of the standard service-related object classes are valid children of the **organizationalUnit** object class. This means that a service does not require its objects to reside at a fixed location in the directory.

For sample code that creates a service object, stores its objectGUID, and locates the service object using the objectGUID, see the MSDN link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Users and Computers Containers

When you upgrade to Windows 2000 from Windows NT, the upgrade process places users, groups, and computer accounts in the containers for users and computers. User and group objects are stored in the Users container, and computer objects are stored in the Computers container. Windows NT and APIs (such as the Net API) that create users, groups, and computer accounts, use tools that create these objects in these containers automatically.

Do not move service-specific objects directly into these containers, or create new containers in these containers. However, services can publish objects as children of a computer object, whether or not the computer object is stored in the Computers container. Create service connection points under the computer object on which the service is installed.

System Container

The System container stores operational information, organized by domain. This information includes the default local security policy, file link tracking, network meetings, and containers for connection points to RPC and Windows Sockets (Winsock). By default, the System container is hidden, and it provides a convenient place for storing objects that are of interest to administrators but not to users.

Services that use the Winsock RnR or RpcNs APIs to advertise themselves automatically create the proper objects in the WinsockServices and RpcServices containers. *Do not explicitly create or move objects in these containers.*

Services that create service-specific objects in the System container must perform the following operations:

- Create a subcontainer of the object class container, which is an immediate child of the System container, and give it a name that clearly relates it to the service.
- Publish the service-related objects in the System container. For example, Microsoft® NetMeeting® uses the Meetings container to publish network meeting objects. When multiple products from a single vendor publish objects in this manner, the containers for each service need to be created as children of a container (an object of the class container), and the container needs to be given a name that clearly relates to the particular vendor. The vendor-specific container must be created as a child of the System container.

You need to publish services in a container beneath the System container when your service, like NetMeeting, is not strongly tied to a single computer.

Publishing Services in Active Directory

Observe the following principles when publishing services:

- Create connection point objects when you install services so that a user with sufficient privileges is able to install the published service using that connection point.
- Restrict the service to the update at run time of existing connection point objects.

Regardless of the publishing method you choose, you must understand and allow for access restrictions on the directory in your installation procedure and at run time. The RpcNs and RnR APIs create connection points in the RpcServices and WinsockServices containers, respectively. These are located in the System container in each domain. **Service-Connection-Point** objects need to be created as children of the **Computer** object for the computer where the service is installed.

To create an object of any kind, the process that creates the object must have permission to create a child for the class and container in which the object is to be created. To remove an object, the process that deletes the object must have permission to delete a child for the class and container in which the object is to be deleted. Or, it needs to have permission to delete the object itself. To update a connection point, the process performing the operation must have permission to write to the properties of the object that need to be updated.

These permissions might not be available to a service account or computer account to create objects during installation of a service, but an administrator, or anyone with the necessary permission, can perform the installation. When installing a published service, set the protection on any connection point object to allow update by the service at run time. This allows the service to run with a minimally privileged account and still be able to maintain its connection point.

For more information about the publishing of services,
see the MSDN link on the Web Resources page at
http://windows.microsoft.com/windows2000/reskit/webresources.

The Service Connection Point (SCP) class is a base object class for use in publishing and locating services. SCP is intended for use in publishing client bindings. The properties defined for this class are sufficient for a service to publish the information that a client requires to be able to bind to an instance of the service. The clients of the service must have prior knowledge of how to interpret and use the binding attributes; Active Directory does not define usage. Services that need to publish additional information about themselves can extend the schema by *subclassing* the Service-Connection-Point, giving the subclass a distinct name that makes it easily recognizable. For more information about extending the schema, see "The Active Directory Schema" in this book.

A setup program (or installation option for the executable routine of the service) installs the service and creates a service account, if necessary, and the service connection point. At service startup, the service itself checks to ensure that the service connection point contains the correct binding information.

Multiple instances of a service can exist in a network. Each instance can have different capabilities. For example, different database servers might contain entirely different data, although they are all of the same kind of service, or *service class*.

In addition, services can be replicated. A replicated service has multiple instances that have identical capabilities. A client can connect to any copy of a replicated service and receive identical service. Active Directory is an example of a replicated service for which all domain controllers for a given domain store identical data and provide identical services. When you create a service connection point for your service, you need to consider by what method clients will locate it. If you will have multiple instances of the service, you must consider how the clients will distinguish the instance with the desired capabilities from similar instances with different capabilities.

The serviceConnectionPoint Attributes

The attributes defined for the serviceConnectionPoint class are sufficient for a service to publish the information that a client needs to be able to bind to an instance of the service.

The serviceConnectionPoint object has an attribute, the *keywords* multivalued attribute, that contains string values—keywords—that enable client applications to find service connection point objects for a specific type of service. This attribute is indexed and is replicated to the Global Catalog server. The service that publishes the service connection point needs to add each keyword as an individual element in the multivalued attribute.

For information about keywords and publishing your service connection points, see the MSDN link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Publishing with the RPC Name Service (RpcNs)

RPC services use the RpcNs API to publish themselves in a namespace; the Windows 2000 RpcNs APIs publish the RPC entries in Active Directory. Services create RPC bindings and publish them in the Active Directory namespace as named RPC Server entries with attributes. The attributes include the unique interface ID and a GUID that are known to clients. Clients can then search for RPC servers that offer the desired interface, import the binding, and connect to the server.

For information about RPC Name Service and Active Directory, see "Windows 2000 RPC Name Service and Integration with Active Directory" later in this chapter.

Publishing with Windows Sockets Registration and Resolution (RnR)

Windows Sockets services can use the RnR APIs to publish services and to locate those services. RnR publication occurs in two steps. The first step installs a service class that associates a GUID with a name for the service. The service class can hold service-specific configuration information. Services can then publish themselves as instances of the service class. After the services are published, clients can query Active Directory for instances of a given class using the RnR APIs, and select an instance to which to bind. When a class is no longer useful, it can be removed.

Finding and Viewing Service Information In Active Directory

The Services container is an immediate child of the Configuration container. If your service-related objects are direct instances of the **serviceConnectionPoint** class, locate published services using ADSI to search for any object where objectCategory is equal to serviceConnectionPoint and objectClass is equal to serviceConnectionPoint. The *keywords* attribute contains the vendor-specific and application-specific GUID. To fill in the object category, query using objectCategory rather than ObjectClass.

If your service-related objects are instances of a class derived from **serviceConnectionPoint**, then use ADSI to search for any object where objectCategory is equal to serviceConnectionPoint or objectClass is equal to your service connection point.

Note Searching by object category is recommended because categories are indexed, making searches faster than searches on non-indexed values. However, objectClass is not indexed.

For binding, clients need to use the information stored in the *serviceBindingInformation* attribute and, potentially, the *serviceDNSName* and *serviceDNSNameType* attributes.

Windows 2000 RPC Name Service and Integration with Active Directory

Client processes use Name Service APIs to find RPC servers that are *exported.* An RPC server exports itself by using RpcNs APIs. To connect to an RPC server, a client needs a compatible binding that offers the interface and version desired and has the proper protocol sequence. It finds compatible bindings by calling the RpcNs API. RPC Name Service provides the binding required by the RPC clients to use RPC and communicate with RPC servers.

Although the RpcNs APIs do not provide for ACLs, ACLs are supported by Active Directory and can be associated with Name Service entries. The Windows 2000 implementation of RPC Name Service uses Active Directory ACL enforcement and prevents unauthorized exports that need to be instantiated. It also makes sure that clients can import based on the ACLs.

The mechanisms for associating ACLs use *out-of-band* data. Out-of-band data is data that travels outside the normal flow of data from one process to another. For example, stream sockets are very useful for sending a stream of bytes from one process to another. The data that flows in this manner is called *inline* data. However, applications at either end occasionally need to communicate with each other without interrupting the regular flow of inline data. Such communication can be achieved by sending out-of-band data, which is sent at a higher priority to the receiver on the same socket as that used for the inline transfer. The advantage of using out-of-band data is that the data goes from sender to receiver directly, without any wait at the end of the incoming data stream. Some methods of configuring ACLs to facilitate the transmission of out-of-band data by processes are as follows:

- The RpcServices container is created when a domain controller is installed. By default, all RPC Name Service entries created under RpcServices inherit the ACLs of this container object. By associating a default ACL with this container, the administrator can ensure that the ACL is applied across that domain for all RPC Name Service entries.
- The Active Directory Users and Computers snap-in allows you to set ACLs on arbitrary Active Directory objects.
- The RPC Locator impersonates the security identifier (SID) of the child process while making directory service calls to enforce ACL verification. The export succeeds locally, but is not persistent. It is visible only if you use it in a manner that is compatible with Windows NT 4.0, such as the broadcast method discussed in the following section.

Note ACL failures on RPC Name Server lookups are treated as if the entry did not exist.

Windows 2000 RPC Name Service Process

All entries are instantiated as Active Directory objects. In every Windows 2000 domain there is a container object that is the root of the RPC Name Service. The distinguished name is as follows:

CN=RpcServices,CN=System,CN=Configuration,DC=<*domain name*>

Figure 5.3 illustrates the role of the Windows 2000 RPC Name Service with the server computer, the client computer, and Active Directory.

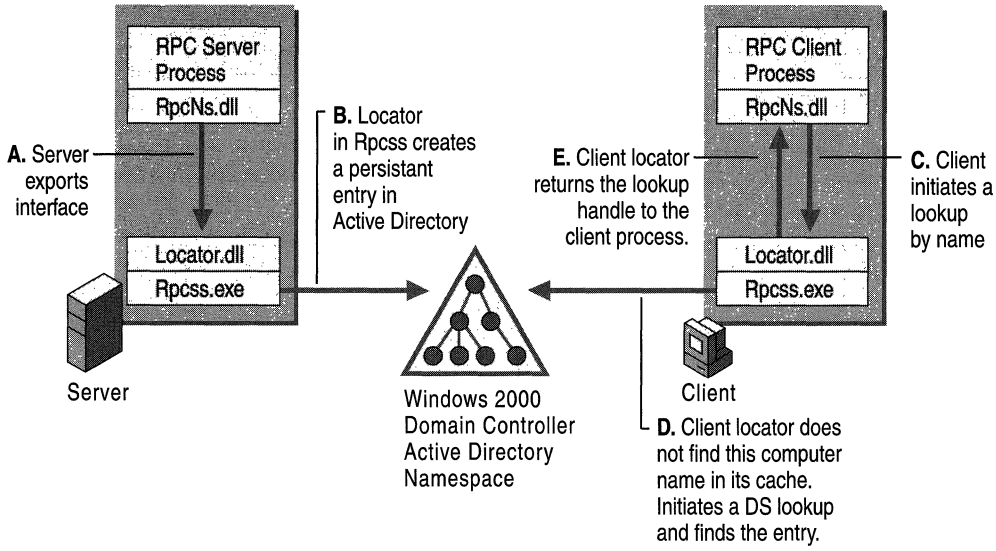


Figure 5.3 Windows 2000 RPC Name Service

Optional Broadcast

The Windows 2000 RPC Locator looks in Active Directory if the entry is not found first in the memory cache.

If the lookup does not succeed, the RPC Locator reverts to the broadcast method used in Windows NT 4.0, notifying all nodes on the network of its existence. Broadcasting is also important for ensuring interoperability between Windows 2000 and Windows NT 4.0, because Windows NT 4.0 computers do not interact with Active Directory and can only be found through the broadcast method. If the exporting server is running Windows 2000 and the domain controller with which it is communicating is running Windows NT 4.0, the export does not persist in Active Directory.

Therefore, the broadcast method is necessary. In a Windows 2000 native-mode environment that is running only Windows 2000–based domain controllers, sending broadcasts for every unsuccessful attempt is not recommended, and you need to disable the broadcast method.

Setting the *BitFlags* value equal to 1 in the *NameServiceFlags* attribute on the RpcServices container enables RPC Locator to be compatible with Windows NT 4.0 in the domain. The broadcast method is enabled by default.

Note Windows 2000–based computers do not initiate a broadcast lookup if Active Directory Locator is enabled and the Broadcast is disabled.

Configuration from the Client Side

If the domain contains computers that are running Microsoft® Windows® NT version 3.51 or later, RPC Name Service lookups must be configured. For RPC Name Service lookups on Windows 2000, use the Active Directory Users and Computers snap-in. Also, for compatibility with Windows NT 4.0, NetBIOS is used for broadcasts.

Enable RPC Name Service lookups from Active Directory Users and Computers snap-in

From the Active Directory Users and Computers snap-in, access the **View** menu and select **Advanced Features**. Under the System container is another container listed as RpcServices. Open the RpcServices container, and right-click **Properties**. The **RpcServices** property page is displayed and contains the **Enable RPC Name Service lookups for pre-Windows 2000 computers** checkbox. By default, RPC name service lookups for computers running Windows NT 4.0 and earlier is enabled.

Use of RPC Locator and NetBIOS

NetBIOS is used for mail slots in the RPC Locator. The RPC Locator uses mail slots for broadcasts and uses their corresponding responses in the domain. This feature is used only to provide compatibility with Windows NT 4.0. For Windows 2000, NetBIOS is enabled on TCP/IP by default. You can choose to disable NetBIOS on a per-computer basis, or you can disable it when you configure DHCP.

For information about disabling NetBIOS over TCP/IP, see the *Microsoft® Windows® 2000 TCP/IP Core Networking Guide*.

Security Considerations for All Services

Services can execute in one of two security contexts:

- LocalSystem account
- A Windows 2000 Domain account, referred to as a *service account*

The security context under which the service runs affects the access rights that the service has on the computer and on the network.

A service can run in the context of the LocalSystem account or in a specific service account. LocalSystem is a special, predefined local account available only to system processes. This account does not have a password.

On computers running Windows NT 4.0 and earlier, a service that runs in the context of LocalSystem inherits the security context of the Service Control Manager. The service is not associated with any logged-on user account and does not have credentials (domain name, user name, and password) to be used for verification. The service has limited access to network resources, such as shares and pipes, because it has no credentials and must connect using a null session.

On computers running Windows 2000, a service that runs in the context of the LocalSystem account uses the credentials of the computer when accessing resources over the network and has full access to local resources. A service that runs in the context of LocalSystem on a domain controller has full access to the directory, because the domain controller hosts a directory replica and LocalSystem has complete access to local resources.

In general, a service needs to run under a separate service account on any system, regardless of role, and under a separate service account on domain controllers. Create the service account as part of the service setup and configure ACLs appropriately to give the rights to the service account.

Do not run a service under LocalSystem on a domain controller. Doing so gives your service too much access to Active Directory, because LocalSystem on a domain controller has complete control of Active Directory. Most security-conscious customers will not accept applications that require this context because of its potential to cause serious damage to directory information.

LocalSystem vs. Service Account

For services that access Active Directory, the LocalSystem context is constraining. When a service runs under the LocalSystem account on a system that is a domain member, such as Microsoft® Windows® 2000 Server running as a member server and Microsoft® Windows® 2000 Professional, that service runs under the context of the computer account when accessing domain resources, such as Active Directory. Computer accounts typically have very few privileges and do not belong to groups. Adding machine accounts to groups is not recommended; the accounts are subject to deletion and re-creation if the computer leaves and then rejoins the domain. The default configuration of an ACL for Active Directory permits minimal access for computer accounts. Services running as LocalSystem, therefore, have only minimal access to Active Directory.

Installing a service with a domain account and password allows the service to have the access that the domain account offers. The domain account can be a member of multiple security groups and is not subject to deletion and re-creation if the machine leaves and rejoins the domain. Group memberships can be easily used to grant access to the desired areas of Active Directory for use by the services that use service accounts. However, running a service in the context of a service account has these disadvantages:

1. The account must be created before the service can run. If the service setup creates the account, the setup must run from an account with sufficient privileges to create accounts in the directory service.
2. Service account passwords are stored on each computer on which the service is installed. If the password for a service account on a computer is changed, the service cannot start on that computer until the password is set to the new password for that service.

Important If your service runs under the LocalSystem account, you must test your service on a member server to ensure that your service has sufficient rights to read/write to Active Directory. A domain controller must not be the only Windows 2000–based computer on which you test your service. Remember that a service running under LocalSystem on a Windows 2000 domain controller has complete access to Active Directory, while a member server runs in the context of the computer account, which has substantially fewer rights than a domain controller.

Mutual Authentication

Mutual Authentication is a security feature in which a client process must prove its identity to a server, and the server must prove its identity to the client, before any application traffic is sent over the client-to-server connection. Support for mutual authentication is provided by the security support provider interface (SSPI) and is exposed directly through the SSPI APIs and services that layer upon SSPI, including RPC and COM+.

Not all security packages available to SSPI, nor all services running Windows 2000, support mutual authentication. An application must request mutual authentication and a supporting security package to obtain mutual authentication.

Mutual authentication requires that the client and server prove their respective identities to each other before performing any application functions. Identity can be proved through a trusted third party and use shared secrets, as in Kerberos v5, or through cryptographic means, as with a public key infrastructure. Each party is identified by a *principal name*.

Principal Names

The central principal of mutual authentication is that neither party must "trust" the other before identity has been proven. This means, in practical terms, that the server must be able to determine who the client is without asking the client, and the client must be able to determine who the server is without asking the server. This prevents security from being compromised through simple impersonation.

Mutual Authentication and Kerberos

Clients establish a local security context, either by executing in a previously established context—for example, in the session of a logged-in user—or by explicitly presenting credentials to the underlying security provider. The server simply refuses to accept connections from any client that is not authenticated. The client authenticates the server by composing a service principal name based on information it already knows about the server or obtains from another trusted source (not including the server, which is not trusted until authenticated). The client presents the service principal name to the security system, demanding that the server prove it can authenticate using the presented SPN. The client refuses further communications with a server that cannot authenticate the SPN.

Note If the account of the service is in a forest different from the account of the client, mutual authentication fails because Kerberos cannot find the service account.

Both service and client must be running on Windows 2000–based computers; otherwise, mutual authentication with Kerberos will fail, because earlier versions of Windows do not support Kerberos.

Service principal names include the DNS name of the host on which the service is running. You must use the DNS name; NetBIOS names are not supported.

Service Principal Names

Service principal names are associated with the security principal (user or groups) in whose security context the service executes. SPNs are used to support mutual authentication between a client application and a service. An SPN is assembled from information that a client knows about a service. Or, it can obtain information from a trusted third party, such as Active Directory. A service principal name is associated with an account and an account can have many service principal names.

For information about registering service principal names in Active Directory at service installation, see the MSDN link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Service Principal Names Syntax

A service uses the following elements to compose a service principal name.

The basic syntax of service principal name is as follows:

<service type>/<instance name>:<port number>/<service name>

where the elements of the syntax have the following meanings:

- *service type*. Type of service, such as "www" for World Wide Web service or "ldap" for Lightweight Directory Access Protocol.
- *instance name*. Name of the instance of the service. Depending upon the service type, it is either the name or IP address of the host running the service.
- *port number*. Number of the port used by the service on the host, if it is different from the default for the service type.
- *service name*. Name of the service. The name can be the DNS name of a host, of a replicated service, or of a domain; or it can be the distinguished name of a **service connection point** object or of an **RPC service** object.

If *service name* and *instance name* are the same, as they are for most host-based services, then a service principal name can be abbreviated to two components:

- *<service type>/<instance name>:<port number>*

  If *port number* is a port number different from the default for the service type specified by *service type*, you must specify the port number.
- *<service type>/<instance name>*
  - If *port number* is the default port number for *service type*, you do not need to specify the port number, which is intended to be compatible with the Generic Security Services API (GSS) form

For more information about GSS and SSPI, see "Authentication" in this book.

Creating the Service Principal Name

A client creates the service principal name for a service. A service principal name can be one of the following: the DNS name of a domain, the DNS name of a host, or the distinguished name of a service connection point object. The SPN is the same for any method of authentication. When using Kerberos to authenticate itself to a server, the client requests a session ticket for the service principal name; when using certificate-based authentication, the SPN is validated against the contents of the "SubjectName" field of the certificate of the server.

Host-Based Service Named in DNS

A host-based service is a service that is identified by the name of the host on which the service runs. In such cases, the service principal name is of the service is as follows:

<service type>/<host name>:<port number>

Or, if the service is using the default port for the service type specified by *service type*, then the SPN can be abbreviated to the following:

<service type>/<host name>

Services Named in the Directory Service

The service principal name for services named in the directory service has the following syntax:

<service type>/<host name>:<port number>/<distinguished name>

where the elements of the syntax have the following meanings:

- *service type*. Type of service that is sought (for example, "print").
- *distinguished name*. Distinguished name in the format specified by Request for Comments (RFC) 1779 of the Internet Engineering Task Force, of an instance of the service type *service type* (for example, "cn=bldg26,dc=ntdom,dc=reskit,dc=com").
- *host name*. DNS name of the host running an instance of *distinguished name*.
- *domain name*. Name of the domain that contains the account running the service specified by *distinguished name* (formed from the "dc=" components of *distinguished name*).

For example, the service principal name for the print service for the NTDOM group in building 26 at Reskit, whose distinguished name is "cn=bldg26,dc=ntdom,dc=reskit,dc=com", which is running on nonstandard port number 1234 on host "prt1.ntdom.reskit.com", is as follows:

print/prt1.ntdom.reskit.com:1234/cn=bldg26,dc=ntdom,dc=reskit,dc=com

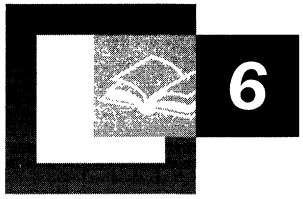
For more information about service principal names, see the MSDN link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Additional Resources

For more information about developing services for publication in Active Directory, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

CHAPTER 6

Active Directory Replication



Active Directory™, the directory service that is included with Microsoft® Windows® 2000, is a distributed directory service. Objects in the directory are distributed across the domain controllers in a forest, and all domain controllers in a domain can be updated directly. Replication is the process by which the changes that are made on one domain controller are synchronized with all other domain controllers in the domain or forest that store copies of the same information. Data integrity is maintained by tracking changes on each domain controller and updating other domain controllers in a systematic way. Active Directory replication uses a connection topology that is created automatically, which makes optimal use of beneficial network connections and frees the administrators from having to make such decisions.

In This Chapter

Related Information in the Resource Kit

- For more information about planning sites, site topology, and domain controller location, see "Designing the Active Directory Structure" in the *Microsoft® Windows® 2000 Server Resource Kit Deployment Planning Guide*.
- For more information about the Active Directory database and object storage, see "Active Directory Data Storage" in this book.
- For more information about the directory tree, see "Active Directory Logical Structure" in this book.
- For information about troubleshooting Active Directory replication, see "Active Directory Diagnostics, Troubleshooting, and Recovery" in this book.
- For information about Active Directory replication and restoring domain controllers, see "Active Directory Backup and Restore" in this book.

Active Directory Replication Model

In Active Directory, the directory tree represents all of the objects in a Windows 2000 forest, and this directory tree is partitioned in a way that allows it to be distributed to domain controllers in different domains within a forest. The Active Directory replication model encompasses the manner in which changes are propagated and tracked among domain controllers. Each domain controller in a forest stores a copy of specific parts of the directory. Each defined segment of the directory is called a *directory partition*. A copy of the contents of one directory partition on a specific domain controller is called a *replica*. Updates to replicas are synchronized among the domain controllers that store the same directory partitions during the process of replication.

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 one schema directory partition per forest.
- The configuration partition, which contains replication configuration information (and other information) for the forest. There is one configuration directory partition per forest.
- The domain partition, which contains all objects that are 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 all domain controllers of that domain (and nowhere else); a full replica of a forest's configuration and schema partitions is stored on all domain controllers of that forest (and nowhere else).

A partial replica contains a subset of the attributes of all directory partition objects and is read-only. Partial replicas are stored only on Global Catalog servers. An attribute is contained in a partial replica if and only if the attribute's *attributeSchema* object has *isMemberOfPartialAttributeSet* equal to TRUE.

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. On domain controllers that are Global Catalog servers, the database also stores partial replicas of directory partition objects from other domains. Partial replicas are stored on Global Catalog servers so that searches of the entire directory can be achieved without requiring referrals from one domain controller to another.

Important Note the difference between the directory tree (the Windows 2000 forest) and the physical database on a specific domain controller in that forest. The directory includes all of the objects in the forest. The directory database on a specific domain controller in the forest includes replicas of the domain objects for only that domain in addition to the replicas of the configuration and schema objects for the entire forest.

For more information about directory partitions, see "Active Directory Data Storage" in this book. For information about directory searches, see "Name Resolution in Active Directory" in this book. For more information about Global Catalog servers, see "Active Directory Logical Structure" in this book.

Replication Model Benefits

The key benefits of the Active Directory replication model are the following:

- Active Directory always replicates changes to the correct object and is able to 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.
- This replication model minimizes update conflicts and supports attribute-level conflict detection and resolution. Only those attribute values that have changed, not the entire object, are transmitted as updates.
- 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 that are required for the forest and those that are specific to the server domain. Even though all Global Catalog servers contain all objects in the forest, they contain only a subset of the attributes for these objects.
- Replication topology (including choice of transports) is flexible to make the best use of different network topologies. Sites provide an intelligent basis for automatically generating replication topology and helping clients perform intelligent domain controller selection.
- 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 because the number of network round-trips by replication protocols is minimized.
- Dependencies on other services, such as time synchronization (W32Time), are minimized.

Replication Model Components

The following mechanisms contribute to the overall replication system:

- Multimaster loose consistency with convergence, which maintains data integrity.
  - "Multimaster" means that a directory partition can have many writable replicas, or copies, that must be kept consistent between domain controllers in the same forest. The replication system propagates changes made on any specific domain controller to all other domain controllers in the forest that store the directory partition in which the change occurs.
  - "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 is allowed to reach a steady state in which no new updates are occurring and all previous 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 not sent directly from one domain controller to all other domain controllers. Instead, a change is sent directly to only a subset of domain controllers. This subset of domain controllers is then responsible for sending the change to other domain controllers, and so on, until the change has reached every domain controller.
- Pull replication, which means that domain controllers request (pull) updates from replication partners. The domain controller in which a change originates does not "push" the change unsolicited to other domain controllers.
- 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.

Multimaster Replication

Active Directory uses multimaster replication to accomplish the synchronization of directory information. True multimaster replication can be contrasted with other directory services that use a *master-slave* approach to updates wherein all updates must be made to the master copy of the directory and then be replicated to the slave copies. This system is adequate for a directory that has a small number of copies and for an environment where all of the changes can be applied centrally. But this approach does not scale beyond small-sized organizations nor does it address the needs of decentralized organizations. With Active Directory, no one domain controller is the master. Instead, all domain controllers within a domain are equivalent. Changes can be made to any domain controller, unlike a single-master system, where changes must be made to one server. In the single-master system, the primary server replicates the updated information to all other directory servers in the domain.

With multimaster replication, it is not necessary for every domain controller to replicate with every other domain controller. Instead, the system implements a robust set of connections that determines which domain controllers replicate to which other domain controllers to ensure that networks are not overloaded with replication traffic and that replication latency is not so long that it causes inconvenience to users. The set of connections through which changes are replicated to domain controllers in an enterprise is called the *replication topology*.

Multimaster update capability provides high availability of write access to directory objects because several servers can contain writable copies of an object. Each domain controller in the domain can accept updates independently, without communicating with other domain controllers. The system resolves any conflicts in updates to a specific directory object. If updates cease and replication continues, all copies of an object eventually reach the same value.

The manner in which a directory service stores information directly determines the performance and scalability of the directory service. Directory services must handle a large number of queries compared to the number of updates they must process. A typical ratio of queries to updates is 99:1. By creating multiple copies of the directory and keeping the copies consistent, the directory service can handle more queries per second.

Multimaster replication provides the following advantages over single-master replication:

- If one domain controller becomes inoperable, other domain controllers can continue to update the directory. In single-master replication, if the primary domain controller becomes inoperable, directory updates cannot take place. For example, if the failed server holds your password and your password has expired, you cannot reset your password and therefore you cannot log on to the domain.
- Servers that are capable of making changes to the directory, which in Windows 2000 are domain controllers, can be distributed across the network and can be located in multiple physical sites.

Store-and-Forward Replication

Store-and-forward replication is designed to reduce communication over slow WAN links. An update replicates first to nearby replicas and from there to replicas that are farther away.

Store-and-forward replication eliminates the need to send every change directly from the server that is accepting the change to all other servers that hold replicas of the affected directory partition. A server that is accepting a change can send the change to nearby servers. One of these servers can then send the change to a distant server, which in turn forwards the change to nearby servers. Store-and-forward greatly reduces the WAN traffic that is produced by replication.

To alleviate the administrative complexity of managing connections between all domain controllers, the system can create the topology automatically. You administer replication indirectly by defining a simplified network model within your directory. This model is based on concepts of 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. You also have the option of manually creating replication connections to exert finer control. Manually created connections coexist with automatically generated ones, so if you want to fine-tune one connection, you need not sacrifice the benefits of automatic management for other connections.

Pull Replication

Active Directory uses *pull* replication. In pull replication, 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 push replication, a source sends information to a destination unsolicited, in an attempt to bring the destination more up-to-date. Push replication is problematical because it is difficult for the source to know what information the destination needs. Perhaps the destination has received the same information from another source. If a source sends information to a destination, there is no guarantee that the destination is going to apply it; if the source assumes otherwise, the system is unreliable.

State-based Replication

Active Directory uses a state-based approach to replication. In state-based replication, each master in the multimaster system applies updates to its replica as they arrive, without maintaining a change log file. (The database system used by Active Directory does use a transaction log file, but this log is part of the database system, not the replication system.) In a typical log-based replication system, each master keeps a log of the updates that it originated. In the log-based system, the goal of each master is to communicate its log to every other replica. After a log has arrived at a replica, the replica applies the log, bringing its state more up-to-date.

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 that is used to resolve conflicts and avoid sending the full replica on each replication cycle. Each originating write operation is assigned a sequence number at its originating domain controller. 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 the sequence numbers 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 because repeated or conflicting updates to an attribute are collapsed into a single state.

Generally speaking, an Active Directory partition replica maintains all of its objects in a list ordered by last modification time. This is a log of sorts, but a coarse one whose size is at most a tiny fraction of the size of the replica itself. A typical replication request can be satisfied by examining only the last few objects on this list because the replication destination server is aware of how much of its replication source's list has already been processed.

Replication Behavior

Replication behavior is consistent and predictable; given a set of changes to a specific replica, it can be predicted that changes are going to be propagated to all other replicas.

The following key points are central to understanding the behavior of Active Directory replication:

- An object is available for replication as soon as it is written. Writes to single objects are atomic, so "partially written" objects are not possible.
- Objects are not necessarily replicated in the order in which they are updated.
- After an update cycle is initiated, a specific replication cycle sends all available changes from the source replica to the destination replica, including changes that occur while the replication cycle is in progress.
- Replication is store-and-forward and "ripples" through a set of connected replicas.
- Multimaster conflict resolution is guaranteed reliable even if clocks become unsynchronized or move backward.
- The graph of replication connections is not always a spanning tree (which by definition does not contain redundant links)—the graph can, and generally does, contain cycles. Redundant connections reduce replication latency, especially in case of failure. A propagation-dampening mechanism eliminates redundant replication.
- Replication within a site is triggered by a change notification mechanism when an update occurs, moderated by a short, configurable delay (because groups of updates frequently occur together).
- Replication between sites typically occurs at scheduled intervals (change notification between sites is optional).
- The system is resilient in the face of load spikes and temporary failures.

- The replication system is designed to be stable. Every time that a replication destination receives information from a replication source, the destination becomes more up-to-date. Recovery from failures involves a minimum of extra work.
- Store and forward replication makes efficient use of WAN links—each update crosses an expensive link only once and is compressed.
- Replication topology is managed automatically and optimizes existing connections.

Microsoft® Exchange Server version 5.5 uses sites and replication somewhat differently than Windows 2000 does. For Exchange administrators who are familiar with Exchange directory replication, Table 6.1 provides a summary of the significant differences between Exchange and Active Directory replication. Treating Active Directory as if it were an Exchange directory does not make optimal use of Active Directory.

Table 6.1 Differences Between Windows 2000 and Exchange Server Version 5.5 Directory Services

Windows 2000 Directory Service	Exchange Server 5.5 Directory Service
Master replicas accept updates independently without communicating with other master replicas.	Each directory service object is mastered in a specific site that can be determined by its distinguished name. Updates are multimaster within the master site.
The basis for replication is the object GUID. When an object is renamed, its GUID does not change, so renaming the object cannot lead to replication errors.	The basis for replication is distinguished names. Therefore, to avoid problems, Exchange does not rename objects.
Replicates an update by transmitting only the changed attributes.	Replicates an update by transmitting the entire object.
Supports compression of replication data between sites over remote procedure call (RPC) or Simple Mail Transfer Protocol (SMTP) transports.	Supports compression of replication data between sites over SMTP transport only.
Supports servers that contain only a subset of the objects in the entire directory in addition to Global Catalog servers, which contain all objects but only a partial set of attributes.	Holds a full replica of the directory in each directory server. (The schema in Exchange is site-specific and is not replicated out of its site.)

(continued)

Table 6.1 Differences Between Windows 2000 and Exchange Server Version 5.5 Directory Services *(continued)*

Windows 2000 Directory Service	Exchange Server 5.5 Directory Service
Has a flexible replication topology (including choice of transports).	Has a replication topology between sites that is limited to a spanning tree, which cannot contain redundant links. Replication transport between sites is limited to e-mail.
Uses sites to help generate replication topology and to help clients perform intelligent replica selection; but sites are not tied to directory partitioning.	Uses sites to generate replication topology, but sites are also the unit of directory partitioning.

Active Directory Updates

Replication of updates is triggered when a user (or an application or service) updates a particular object or objects on a domain controller. When an update occurs, a timer is started such that changes are collected for a set period, after which the replication engine notifies adjacent domain controllers in the replication topology within the site (in other sites if notification between sites is enabled). 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.

Originating Updates: Initiating Changes

A Lightweight Directory Access Protocol (LDAP) directory server supports the following four types of update requests:

- Add an object to the directory.
- Modify (add, delete, or replace) attribute values of an object in the directory.
- Move an object by changing the name or parent of the object.
- Delete an object from the directory.

An LDAP directory server processes each write request as an atomic transaction. Separate LDAP requests are separate write transactions. A write request either commits and all its effects are durable, or it fails before completion and has no effect. A write request that commits is called an *originating update*. An originating update is initiated and committed at a specific replica. The absolute success or failure of an update applies even for requests, such as Add or Modify, that might affect several attributes of a single object. In this case, if one attribute update fails, they all fail and the object is not updated.

When an update that originates on one domain controller is replicated to another domain controller, the update on the nonoriginating domain controller is called a replicated update and is distinguished by the replication system from an originating update.

An originating update enforces schema restrictions (allowable parent object types for an object, mandatory and optional attributes for an object, syntax for an attribute) according to the schema that exists on the domain controller at the moment of the update.

Tracking Updates

Some directory services use timestamps to determine what changes need to be propagated. In these systems, it is important to keep the clocks on all directory servers synchronized. But keeping time closely synchronized in a large network is essentially impossible. Network links fail and clocks drift. And with some systems, unless time is perfectly synchronized among all copies of the directory, there is a chance for data loss or directory corruption.

Active Directory replication does not depend on time to determine what changes need to be propagated. It relies instead on the use of update sequence numbers (USNs) that are assigned by a counter that is local to each domain controller. Because these USN counters are local, it is easy to ensure that they are reliable and never "run backward" (that is, decrease in value). The trade-off is that it is meaningless to compare a USN assigned on one domain controller to a USN assigned on a different domain controller. The replication system is designed with this restriction in mind.

Some directory services use timestamps as the primary mechanism to determine what updates "win" (are preserved) in a conflict resolution. The typical policy is "last writer wins." Again, this leads to problems when clocks are inaccurate. For example, suppose that you set the clock forward to December 31, 9999, to perform Year 10,000 testing. As part of the testing, you update an attribute of some object. Then you set the clock back again. If the directory service employs a "last writer wins" policy, the update that you performed during the test cannot be overwritten until the year 10,000.

Active Directory replication does not use timestamps as the primary mechanism to determine what updates "win" (are preserved) in a conflict resolution. Instead, Active Directory uses volatility (number of changes) as the first element of the per-attribute "stamps" that are compared during conflict resolution. The second element is a timestamp. So if an attribute is updated once on domain controller A and once on domain controller B, the last writer's update wins. But if the attribute is updated twice on domain controller A and once on domain controller B, the update of domain controller A wins even if the clock of domain controller B is set forward to December 31, 9999. With Active Directory, clock skew can never prevent a value from being overwritten.

Deciding What Changes to Replicate: Update Sequence Numbers

The current USN is a 64-bit counter that is maintained by each Active Directory domain controller. At the start of each update transaction (originating or replicated) on a domain controller, the domain controller increments its current USN and associates this new value with the update request. This USN value is stored on an updated object in two ways for a replicated write and three ways for an originating write.

- The update's USN value is stored with each attribute changed by the update as the *local USN* of that attribute (originating and replicated writes). You can use the Repadmin command-line tool to view the local USN. Type **repadmin /showmeta *<object_DN>*** at a command prompt and view the column labeled "Loc. USN" in the output. (To use Repadmin, install the Support Tools that are located in the Support\Tools folder on the Windows 2000 Server installation CD. To install the tools, double-click the **Setup** icon in that folder. For information about installing and using the Windows 2000 Support Tools and Support Tools Help, see the file Sreadme.doc in the Support\Tools folder of the Windows 2000 operating system CD. For information about using Repadmin, see *Windows 2000 Support Tools* Help.)

- The maximum local USN among all of an object's attributes is stored as the object's *usnChanged* attribute (originating and replicated writes). You can use Ldp or ADSI Edit to examine an object's *usnChanged* value. Ldp is a graphical tool that you can use to perform connect, bind, search, modify, and delete operations against any LDAP-compliant directory. ADSI Edit is an MMC snap-in that you can install and use to manage Active Directory objects, including viewing and editing attributes. (To use Ldp and ADSI Edit, install the Support Tools that are located in the Support\Tools folder on the Windows 2000 Server installation CD. For more information about using Ldp and ADSI Edit, see *Windows 2000 Support Tools* Help.)
- For an originating write, the update's USN value is stored with each updated attribute as the *originating USN* of that attribute. Unlike the local USN and *usnChanged*, the originating USN travels with the attribute's value as it replicates. You can see the originating USN as the column labeled "Org. USN" in the output from the **repadmin /showmeta** command.

Note The *usnChanged* attribute is indexed, which allows objects to be enumerated efficiently in the order of their most recent attribute write.

You can use Ldp or ADSI Edit to read the current USN of a domain controller. These tools use LDAP to read the *highestCommittedUsn* attribute on the rootDSE object for the domain controller. (For information about using ADSI Edit and Ldp to read rootDSE attributes, see "Active Directory Data Storage" in this book.)

High-Watermark

The *high-watermark* is a value that the destination domain controller maintains to keep track of the most recent change that it has received from a specific source domain controller for an object in a specific directory partition. The source domain controller uses this value to filter the objects that it considers for replication to the destination.

When a destination domain controller requests changes to a directory partition from a source domain controller, the source domain controller provides the changes in increasing order of the *usnChanged* attribute value. The *usnChanged* values from the source domain controller are not stored on objects at the destination domain controller, but the destination domain controller keeps track of the *usnChanged* value of the most recent object it successfully received from the source domain controller for a specific directory partition. This USN is called the destination's high-watermark with respect to the directory partition and the source domain controller.

When requesting changes, the destination domain controller sends its high-watermark value to the source domain controller. The source domain controller uses the information in the high-watermark to reduce the set of objects that it must consider for replication to the destination. No object whose *usnChanged* value is less than or equal to the high-watermark value can hold updates that the destination domain controller has not already received.

The high-watermark serves to decrease the CPU time and number of disk I/O operations that would otherwise be required to send only the changes that the destination domain controller has not yet received.

You can see the high-watermark in the output of the **repadmin /showreps /verbose** command. Look for lines that begin with "USNs:". The high-watermark USN is the number that is followed by "/OU".

Up-to-Dateness Vector

The *up-to-dateness vector* is a value that the destination domain controller maintains for tracking the originating updates that are received from all source domain controllers. The source domain controller uses this value to reduce the set of attributes that it sends to the destination domain controller.

When a destination domain controller requests changes for a directory partition, it provides its up-to-dateness vector to the source domain controller. On the basis of this value, the source domain controller can determine that the destination does or does not have an up-to-date value (or multivalue) for an attribute, and it sends updates accordingly. If the destination already has an up-to-date value, the source domain controller does not send that attribute. If the source has no attributes to send for an object, it sends no information at all about that object.

The up-to-dateness vector can contain an entry for each domain controller that holds a full replica of the directory partition. If the up-to-dateness entry that corresponds to source domain controller X contains the USN *n*, the destination domain controller guarantees that it holds all updates to a specific directory partition that originated at domain controller X and that have an originating USN value of less than or equal to *n*.

The up-to-dateness vector and the high-watermark are complementary filter mechanisms that work together to decrease replication latency. Whereas the high-watermark prevents irrelevant objects from being considered by the source domain controller with respect to a single destination, the up-to-dateness vector helps the source domain controller to filter irrelevant attributes (and entire objects if all attributes are filtered) on the basis of the relationships between all sources of originating updates and a single destination.

For a specific directory partition, a domain controller maintains a high-watermark value for only those domain controllers from which it requests changes, but it maintains an up-to-dateness vector entry for every domain controller that has ever performed an originating update, which is typically every domain controller that holds a full replica of the directory partition.

You can see the up-to-dateness vector in the output of the **repadmin /showvector** command. (For more information about using Repadmin, see *Windows 2000 Support Tools* Help.)

Resolving Conflicts: Stamps

Suppose that an attribute of some object is changed on domain controller X. Then before the change on domain controller X has replicated, the same attribute of the same object is changed on domain controller Y. Active Directory must ensure that when replication has occurred, all replicas agree on the value of the updated attribute.

Active Directory ensures agreement by attaching a unique *stamp* to each replicated attribute value (or multivalue) during an originating update. This stamp travels with the value as the value replicates. If the stamp of the value that was replicated is larger than the stamp of the current value, the current value (including the stamp) is replaced; otherwise, the current value (including the stamp) is left alone.

The stamp has the following three components:

- The *version* is a number that is incremented for each originating write. That is, when performing an originating write, the version of the new value is one larger than the version of the value that is being overwritten. If the attribute was never written before, the version that was assigned to its first originating write is 1.
- The *originating time* is the time of the originating write, to a one-second resolution, according to the system clock of the domain controller that performed the write.
- The *originating DSA* is a GUID that identifies the domain controller that performed the originating write.

You can see all three components of the stamp in output from the **repadmin /showmeta** command. The column labeled "Ver" contains the version, the column labeled "Org. Time/Date" contains the originating time, and the column labeled "Originating DSA" contains the originating DSA (expressed as "site\server" rather than GUID).

When stamps are compared, the version is the most significant, followed by the originating time and then the originating DSA. So if two stamps have the same version, the originating time almost always breaks the tie. In the extremely rare event that the same attribute is updated on two different domain controllers during the same second, the originating DSA breaks the tie in an arbitrary fashion.

Two different originating writes of a specific attribute of a particular object cannot assign the same stamp because each originating write advances the version at a specified originating DSA. (Notice that the originating time does not contribute to uniqueness.) Replicated writes cannot decrease the version because values with smaller versions lose during conflict resolution.

For more information about using Repadmin, see *Windows 2000 Support Tools* Help.

Originating Add

An Add request makes a new object with a unique *objectGuid*. The values of all replicated attributes that are set by the Add request are stamped Version = 1.

The Add request fails immediately if the parent object does not exist or if the originating domain controller does not contain a master replica of the parent object's directory partition.

Originating Modify

All Modify requests can be reduced to requests to replace the current value of an attribute with a new value as follows:

- A Modify request might specify that an attribute be deleted from the object. Attribute deletion is best thought of as replacing the attribute value with NULL. The NULL value occupies no storage of its own but does carry a stamp, as does any value that is stored as a directory attribute.
- A Modify request might specify that a value be added to the current value of an attribute. (This is most useful with an attribute that can have multiple values.) The effect is the same as the equivalent replace request (that is, replace the current values with the current values plus the added value).

For each attribute in the request, a Modify request compares the new value in the request with the existing value in the object. If the values are the same, the request to modify that attribute is ignored. If the resulting Modify request doesn't change any attributes, the entire request is ignored.

Otherwise, a Modify request computes a stamp for each new replicated attribute value by reading the version from the existing value (version=0 for an attribute that has never been written) and then adding 1 to this value. The Modify request replaces the old stamp values with new stamp values.

Originating Move

A Move request is essentially a special Modify request for a single attribute, the *name* attribute. The operation proceeds as described for the Modify request.

Originating Delete

A Delete request is essentially a special Modify request that does the following:

1. Sets the *isDeleted* attribute to TRUE.
2. Marks the object as a *tombstone*, which is an object that has been deleted but not fully removed from the directory.
3. Changes the relative distinguished name to a value that is otherwise impossible (cannot be set by an LDAP application).
4. Strips all attributes that are not needed by Active Directory. A few key attributes, including *objectGuid*, *objectSid*, *distinguishedName*, *nTSecurityDescriptor*, and *usnChanged*, are preserved on the tombstone.
5. Moves the tombstone to the Deleted Objects container, which is a hidden container within the directory partition.

Object deletions are replicated by replicating tombstones. A tombstone is invisible to normal LDAP searches. (A tombstone is visible to searches that use the special LDAP control 1.2.840.113556.1.4.417.) Object references that formerly referred to the deleted object now refer to the tombstone. Therefore, reading such a reference returns the distinguished name of the tombstone, not the distinguished name of the object prior to the object's deletion.

Garbage Collection: Deleting Tombstones

Although they represent deleted objects, tombstones take up space in every directory partition replica; so eventually the tombstones themselves must be deleted to keep the directory database from growing without limit. The garbage collection mechanism deletes tombstones.

Two settings on the object cn=Directory Service,cn=Windows NT,cn=Service, cn=Configuration,dc=ForestRootDomain determine how often garbage collection occurs and which tombstones are deleted:

- The *garbage collection interval* determines the number of hours between garbage collection on a domain controller. The default setting is 12 hours, and the minimum setting is 1 hour.
- The *tombstone lifetime* determines the number of days that tombstones persist before they are vulnerable to garbage collection. The default setting is 60 days, and the minimum setting is 2 days.

Garbage collection runs independently on each domain controller. When garbage collection occurs, it finds the set of tombstones whose originating delete occurred more than a tombstone lifetime ago. Garbage collection deletes each tombstone in the set.

The tombstone lifetime must be larger than the worst-case replication latency for any directory partition. Otherwise, a tombstone might be deleted before it has replicated to every directory partition replica. A directory partition replica that fails to replicate the tombstone would never delete the object, so it would be inconsistent with those replicas that had deleted the object.

Tombstone Lifetime Effect on Restore from Backup

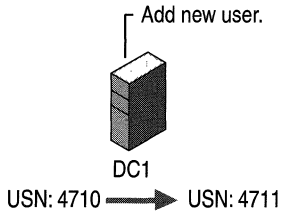
A restore from backup creates a directory partition replica that has not performed replication since the time of backup (or earlier). If the backup was taken more than a tombstone lifetime before the restore, objects deleted in the meantime have no tombstones. The new directory partition replica never learns about these deletions. For this reason, a restore from backup refuses to restore a backup that was taken more than one tombstone lifetime before the time of the restore.

For more information about tombstones and garbage collection, see "Active Directory Data Storage"VDOV in this book. For information about restoring Active Directory from backup, see "Active Directory Backup and Restore" in this book.

Tracking Object Creation, Replication, and Change

The following series of diagrams illustrates the replication-related data for a single object and its attributes as it goes from creation through replication.

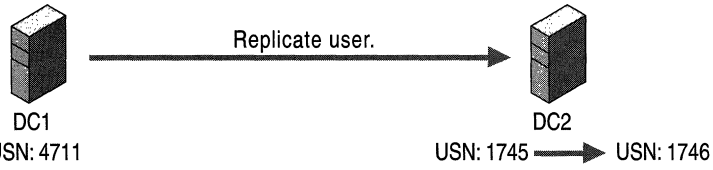
Figure 6.1 shows the replication-related data for the user object when it is first created on domain controller DC1.



Object: *usnCreated:* 4711				Object: *usnChanged:* 4711		
Property	Value	Local USN	Version #	Originating Time	Originating DSA	Originating USN
cn	James Smith	4711	1	1999-09-10 10:49.03	*<DC1_GUID>*	4711
userPassword	6Be8W5q-	4711	1	1999-09-10 10:49.03	*<DC1_GUID>*	4711
sAMAccountName	JSmith	4711	1	1999-09-10 10:49.03	*<DC1_GUID>*	4711
userPrincipalName	JSmith@reskit.com	4711	1	1999-09-10 10:49.03	*<DC1_GUID>*	4711

Figure 6.1 Replication-related Data on DC1 When a User Object Is Created

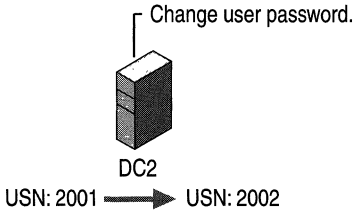
Figure 6.2 shows the change to the destination domain controller when the new user object is replicated. The object is created as a replicated update on DC2. Notice that the per-attribute originating USN and stamp (version, originating time, originating DSA) are replicated from DC1 to DC2, but the per-attribute local USN and per-object *usnChanged* are unique to DC2.



Object: *usnCreated:* 1746				Object: *usnChanged:* 1746		
Property	Value	Local USN	Version #	Originating Time	Originating DSA	Originating USN
cn	James Smith	1746	1	1999-09-10 10:49.03	*<DC1_GUID>*	4711
userPassword	6Be8W5q-	1746	1	1999-09-10 10:49.03	*<DC1_GUID>*	4711
sAMAccountName	JSmith	1746	1	1999-09-10 10:49.03	*<DC1_GUID>*	4711
userPrincipalName	JSmith@reskit.com	1746	1	1999-09-10 10:49.03	*<DC1_GUID>*	4711

Figure 6.2 Replication-related Data on DC2 When a New User Object Is Replicated from DC1

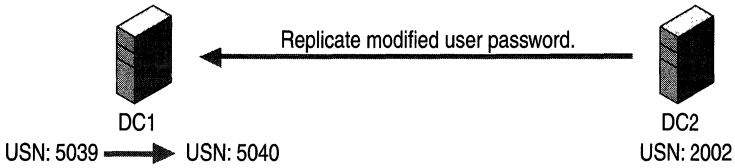
Figure 6.3 illustrates the change in the replicated object on DC2 when someone changes the password (*userPassword* propertyin the diagram) of the object on that domain controller. By this time, the current USN on DC2 has increased from 1746 to 2001. The update request that changes the password increments the current USN to 2002 on DC2. The request also sets the password attribute's originating USN and local USN to 2002 and creates a new stamp for the password value. The version number of this password's stamp is 2, which is one version number higher than the version of the previous password.



Object: *usnCreated:* 1746				Object: *usnChanged:* 2002		
Property	Value	Local USN	Version #	Originating Time	Originating DSA	Originating USN
cn	James Smith	1746	1	1999-09-10 10:49.03	*<DC1_GUID>*	4711
userPassword	Secaucus2	2002	2	1999-09-10 11:53.29	*<DC2_GUID>*	2002
sAMAccountName	JSmith	1746	1	1999-09-10 10:49.03	*<DC1_GUID>*	4711
userPrincipalName	JSmith@reskit.com	1746	1	1999-09-10 10:49.03	*<DC1_GUID>*	4711

Figure 6.3 Replication-related Data on DC2 After the User Password Value Has Been Changed on DC2

In Figure 6.4, the changed password is now replicated back to the original domain controller, whose current USN has increased to 5039. The replicated update increments the current USN to 5040. The per-attribute originating USN and stamp (version, originating time, originating DSA) are replicated from DC2 to DC1, and the per-attribute local USN and per-object *usnChanged* values are set to 5040.



Object: *usnCreated*: 1746				Object: *usnChanged*: 5040		
Property	Value	Local USN	Version #	Originating Time	Originating DSA	Originating USN
cn	James Smith	1746	1	1999-09-10 10:49.03	*<DC1_GUID>*	4711
userPassword	Secaucus2	5040	2	1999-09-10 11:53.29	*<DC2_GUID>*	2002
sAMAccountName	JSmith	1746	1	1999-09-10 10:49.03	*<DC1_GUID>*	4711
userPrincipalName	JSmith@reskit.com	1746	1	1999-09-10 10:49.03	*<DC1_GUID>*	4711

Figure 6.4 Replication-related Data on DC1 After the Password Change Has Been Replicated to DC1

Propagation Dampening

Multiple replication paths can exist between a pair of domain controllers. Multiple paths provide fault tolerance and can reduce latency. However, when multiple paths exist, you might expect the same change to be sent along each path to a specific domain controller. You might even be concerned that a change might replicate in an endless loop. Active Directory prevents these potential problems with multiple paths by using the up-to-dateness vector described earlier.

The following is an example of how ordinary replication occurs:

1. DC A updates a password attribute. The originating USN of the attribute is set to 3.
2. Destination DC B requests changes from source DC A and sends its high-watermark and up-to-dateness vector to DC A.

3. According to the high-watermark that was passed by DC B, source DC A examines one or more objects, one of which contains the changed password. When DC A encounters the changed password attribute, it proceeds as follows:
   - First, DC A finds that the originating DSA of the password attribute is DC A.
   - Therefore, DC A reads the up-to-dateness vector supplied by DC B and finds that DC B is guaranteed to be up-to-date with updates that originated at DC A and that have an originating USN of less than or equal to 2.
   - DC A then finds that the originating USN of the password attribute is 3.
   - Because 3 is greater than 2, DC A sends the changed password attribute to DC B.

To illustrate propagation dampening, suppose that DC B had already received the password update from DC C, which had received it from DC A. In this case, the entry in the up-to-dateness vector of DC B for DC A would contain the USN value 3, not 2. Therefore, DC A would not send the changed password to DC B.

Multimaster Conflict Resolution Policy

As described in "Resolving Conflicts: Stamps" earlier in this chapter, Active Directory resolves attribute value conflicts by comparing versions. However, attribute value conflicts are not the only conflicts that arise.

Given the semantics of LDAP directories, there are four possible conflicts that are created by multimaster updates. Two of these conflicts are in fact different sides of the same conflict, reducing the number of conflict situations to the following three:

- *Attribute value.* A Modify operation sets the value of an attribute. Concurrently, at another domain controller, a Modify operation sets the value of the same attribute to a different value.
- *Add or Move under deleted parent, Delete non-leaf object.* An Add or Move operation makes object C a child of object P. Concurrently, at another domain controller, a Delete operation deletes object P.
- *Sibling name conflict.* An Add or Move operation makes C1 a child of P with C1.rdn = R. Concurrently, at another domain controller, an Add or Move operation makes C2 a child of P with C2.rdn = R.

These conflicts can occur in any multimaster LDAP directory.

In Active Directory, the general approach to resolving these conflicts is to order all update operations (Add, Modify, Move, and Delete) by assigning a globally unique (per-object and per-attribute) stamp to the originating update. In the case of a conflict, the ordering of stamps allows a consistent resolution. This approach is applied to the three preceding conflicts as follows:

- Attribute value. Let V be the value among {V1, V2} with the larger stamp. After resolution, the attribute value at all domain controllers is V.

  Note Conflicts are resolved at the level of the entire attribute value, even for multivalue attributes. The value after resolution is one multivalue or the other, not some combination.

- Add or Move under deleted parent, Delete non-leaf object. After resolution, at all replicas, object P is deleted and object C is a child of the special LostAndFound container in the directory partition. Stamps are not involved in the resolution.
- Sibling name conflict. Let C be the object among {C1, C2} whose relative distinguished name attribute value has the smaller stamp. After resolution, at all domain controllers, C.rdn is a system-assigned value unique to C that cannot conflict with any client-assigned value. For example, if the relative distinguished name of object C was "ABC" before conflict resolution, its relative distinguished name after resolution is "ABC*CNF:*<guid>*", where "*" represents a reserved character, "CNF" is a constant that indicates a conflict resolution, and "*<guid>*" represents a printable representation of the *objectGuid* attribute value.

Replication Topology

Replication topology is the set of connections by which domain controllers in a forest synchronize the directory partition replicas that they have in common. Replication topology is created on the basis of information stored in Active Directory.

The Knowledge Consistency Checker (KCC) is a built-in process that runs on all domain controllers and creates the replication topology for the forest. By default, the KCC runs at 15-minute intervals and designates the replication routes between domain controllers on the basis of the most favorable connections that are available at the time. The KCC creates replication connections between domain controllers in the same site automatically. When you have more than one site, you configure links between the sites; the KCC can then create the connections automatically between the sites as well.

Topology Concepts and Components

Although replication has the effect of synchronizing Active Directory information for an entire forest of domain controllers, the actual process of replication occurs between two domain controllers at a time. Creation of replication topology involves the determination of what domain controller replicates with what other domain controller or domain controllers. When this determination is made for the entire set of domain controllers in a specific site (taking into account that each domain controller must be able to receive changes from all domain controllers in the forest that store the same information), the result is the replication topology for replication within the site. When a forest has domain controllers in more than one site, some of the replication connections between computers must span sites, and a topology for replication between sites is also created.

The total topology is actually composed of several underlying topologies: one for each combination of directory partitions that must be replicated. Domain controllers that store the same domain directory partition must have connections to each other, and all domain controllers must be able to replicate the schema and configuration directory partitions. The schema and configuration directory partitions are replicated over a separate topology; however, where the connections for these directory partitions are identical between domain controllers—for example, two domain controllers store the same domain directory partition—a single connection can be used.

The routes for the following combinations of directory partitions are aggregated to arrive at the overall topology:

- Configuration and schema within a site.
- Each domain directory partition within a site.
- Global Catalog read-only, partial directory partitions within a site.
- Configuration and schema between sites.
- Each domain directory partition between sites.
- Global Catalog read-only, partial directory partitions between sites.

Topology-related Components

Active Directory uses information stored in the forest-wide configuration directory partition to establish and implement the replication topology. Several configuration objects define the components that are required by replication:

- The sites and the domain controllers that are associated with them.
- The connections that identify the routes that replication takes between domain controllers within sites.
- The links that make replication connections between sites possible.
- The transports that the links use to communicate between sites.

The KCC uses these and other objects and their properties to create and manage the connections by which directory updates are transferred and to specify one or more domain controllers from which a particular server requests changes. The domain controllers that replicate directly with each other are called *replication partners*. Each time the KCC runs (every 15 minutes, by default), these partnerships are added, removed, or modified automatically, as necessary, on the basis of what domain controllers are available and how close they are to each other on the network.

The KCC uses the following components to manage replication:

Connections The KCC creates connections that enable domain controllers to replicate with each other. A connection defines a one-way, inbound route from one domain controller, the source, to another domain controller, the destination. The KCC reuses existing connections where it can, deletes unused connections, and creates new connections if none exist that meet the current need.

Servers Each domain controller is represented by a server object. The server has a child object, NTDS Settings, which stores the inbound connections; that is, the connection objects for a server designate the connections *from* source domain controllers *to* the server object.

Sites Sites define sets of domain controllers that are well connected in terms of speed and cost. Domain controllers in the same site replicate on the basis of notification: when a domain controller has changes, it notifies its replication partners. Then the notified partner requests the changes, and replication takes place. Because there is no concern about replication speed or cost, replication within sites occurs as needed rather than as scheduled.

Note To allow for the possibility of network failure, which might cause one or more notifications to be missed, a default schedule of once per hour is applied to replication within a site, in addition to change notification.

Replication between sites occurs according to a schedule; you can use the schedule to determine the most beneficial time for replication to occur on the basis of network traffic and cost. A site is the equivalent of a set of one or more Internet Protocol (IP) subnets.

Note Under circumstances where connections cannot be initiated between both sites (for example, when one site requires a dial-up connection), reciprocal replication can be initiated on the basis of changes rather than a schedule. (For more information about reciprocal replication, see "Reciprocal Replication" later in this chapter.)

Subnets Computers on TCP/IP networks are assigned to sites based on their location in a subnet or a set of subnets. Subnets group computers in a way that identifies their physical proximity on the network. Subnet information is used during the process of domain controller location to find a domain controller in the same site as the computer that is logging on. This information also is used during Active Directory replication to determine the best routes between domain controllers. For more information about subnets, see "Introduction to TCP/IP" in the *Microsoft® Windows® 2000 Server Resource Kit TCP/IP Core Networking Guide*.

Site Links For replication to occur between two sites, a link must be established between the sites. Site links are not generated automatically and can be created in Active Directory Sites and Services. Unless a site link is in place, the KCC cannot create connections automatically between computers in the two sites, and replication between the sites cannot take place. Each site link contains the schedule that determines when replication can occur between the sites that it connects. The Active Directory Sites and Services user interface guarantees that every site is placed in at least one site link. A site link can contain more than two sites, in which case all the sites are equally well connected.

Bridgehead Servers To communicate across site links, the KCC automatically designates a single server, called the *bridgehead server*, in each site to perform site-to-site replication. Subsequent replication occurs by replication within a site. When you establish site links, you can designate the bridgehead servers that you want to receive replication between sites. By designating a specific server to receive replication between sites, rather than using any available server, you can specify the most beneficial conditions for the connection between sites. Bridgehead servers ensure that most replication occurs within sites rather than between sites.

Site Link Bridges When more than two sites are linked for replication and use the same transport, all of the site links are "bridged" in terms of cost by default, assuming that the site links have common sites. When site links are bridged, they are *transitive*. That is, all site links for a specific transport implicitly belong to a single site link bridge for that transport. So in the common case of a fully routed IP network (in which all sites can communicate with each other by IP), you do not have to configure any site link bridges. If your IP network is not fully routed, you can turn off the transitive site link feature for the IP transport (the **Bridge all site links** option on the **General** tab in the IP transport object property sheet or SMTP transport object property sheet). In this case, all IP site links are considered intransitive, and you configure site link bridges. A site link bridge is the equivalent of a disjoint network; all site links within the bridge can route transitively, but they do not route outside the bridge.

Sites Container Hierarchy in Active Directory

Active Directory Sites and Services is the administrative tool that you use to view and manage the hierarchy of objects that are used by the KCC to effect the replication topology. The hierarchy is displayed as the contents of the Sites container, which is a child of the Configuration container. The distinguished name of the Sites container is cn=Sites,cn=Configuration,dc=<*ForestRootDomain*>. The Configuration container is the topmost object in the configuration directory partition and the Sites container is the topmost object in the hierarchy of objects that are used to manage and implement Active Directory replication. The Sites container hierarchy contains the following objects:

- The Sites container, which contains an object for each site in the forest. In addition to site objects, Sites also contains the Subnets container, which contains subnet definitions in the form of subnet objects. Each subnet object has a *siteObject* attribute that links it to a site object.
- Site objects, each of which contains two child objects:
  - The NTDS Site Settings object, which stores directory properties common to all domain controllers in the site, including the schedule for replication within the site.
  - The Servers container, which stores a server object for each domain controller in the site.
- Server objects, each of which contains an NTDS Settings object.
- NTDS Settings object, which represents an instantiation of Active Directory on that server. (When Active Directory is removed from a server, its NTDS Settings object is deleted from Active Directory, but its server object remains.) For a specific server object, the NTDS Settings object contains the individual connection objects that represent the inbound connections from other domain controllers in the forest that are currently available to send changes to this domain controller.
- Connection objects, each of which represents a unidirectional replication agreement between two specific domain controllers, where the destination domain controller is the server object that is the parent object of the NTDS Settings object that stores the connection. Connection objects are created automatically by the KCC; they can also be created manually.

Figure 6.5 shows the expanded Sites container. The NTDS Settings object for one server is selected, and the related connection objects are displayed in the details pane. The **From Server** column displays the name of the domain controllers from which the selected domain controller receives replication (the names of its current replication partners).

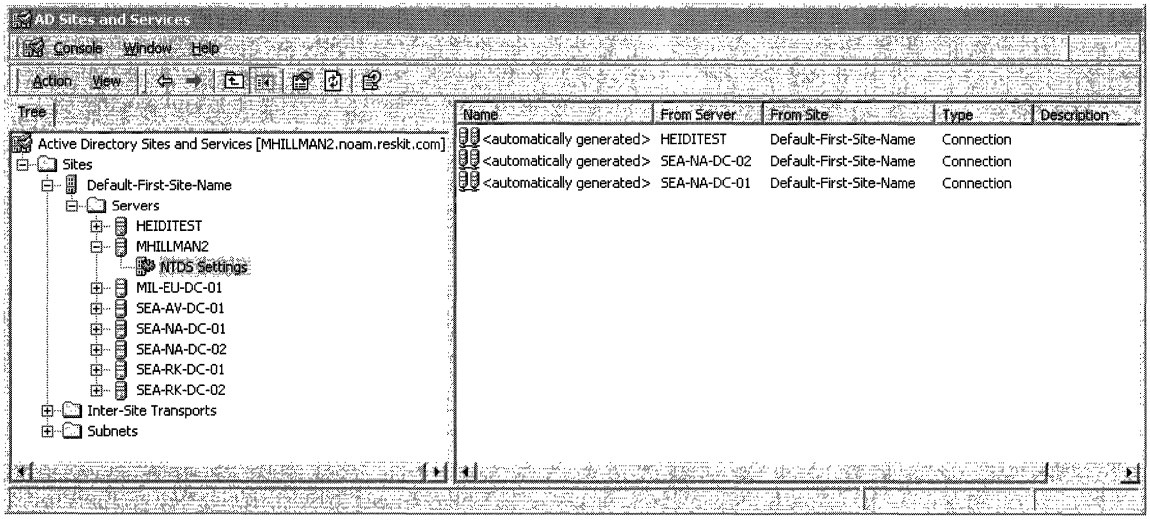


Figure 6.5 Hierarchy of Objects in the Sites Container

For information about how to create and manage objects in the Sites container, see Windows 2000 Server Help.

Sites and Replication

A *site* represents a region of uniformly good network access, which can be interpreted as being generally equivalent to local area network (LAN) connectivity. LAN connectivity assumes high, inexpensive bandwidth that allows similar and reliable network performance, regardless of which two computers in the site are communicating. This quality of connectivity does not indicate that all servers in the site must be on the same network segment nor that hop counts between all servers must be identical. Rather, it can be interpreted as the measure by which you know that if a large amount of data needed to be copied from one server to another, it would not matter to you which servers were involved. If you find that you are concerned about such situations, you might consider creating another site.

Replication Efficiency

Replication within a site is driven by changes. The high speed and reliability of LAN connectivity lends itself to on-demand data transfer. When a change is applied to a specific replica, the replication engine is triggered. The replication engine waits for a configurable interval (by default, five minutes) and then notifies the first replication partner. Each additional partner is notified after a configurable delay (by default, 30 seconds).

Generation of the topology within a site is achieved by using the minimum number of connections possible. The default replication topology in a site is a bidirectional ring, with sufficient additional connections added to keep the number of hops between replication partners to three or less. To summarize, the default worst-case delay for propagation within a site is the maximum number of hops between a source and destination domain controller (3 hops) multiplied by the delay at each domain controller in the path (5 minutes), or 15 minutes for three hops.

Replication between sites is scheduled so that you can control replication costs and keep replication from overwhelming your communication links. Several factors contribute to the speed of replication between sites, including the nature of the physical connections and how expensive they are to use. Active Directory supports site connections in all speed ranges, from T3 links on the high end to dial-up lines on the low end. You use the speed of the connection between your sites to assign a cost to the communication link, and replication uses the cost to establish the least expensive route for replication traffic.

For information about planning networks, see "Determining Network Connectivity Strategies" in the *Deployment Planning Guide*.

Site Design with Replication in Mind

When you group a set of IP subnets into a site, you do so based on the fact that these subnets have high bandwidth, LAN connectivity, possibly involving hops through high-performance routers.

A single domain can span multiple sites, and a single site can contain multiple domains. Domain architecture should be constructed independently of site design —sites exist primarily, but not solely, to assist the domain controller Locator and the replication infrastructure. Although there are no absolute rules for deciding when to place two subnets in the same site, understanding how Active Directory uses site information can help you make an informed decision.

Active Directory uses site information in the following ways:

- When a client requests a connection to a domain controller (for example, when logging on), sites are used to enable the client to connect to a domain controller with good connectivity whenever possible. Fast connections reduce network latency and conserve network bandwidth.
- When the KCC configures replication connections between domain controllers, it creates more connections between domain controllers in the same site than between domain controllers in different sites. The results are lower replication latency within a site and less replication bandwidth between sites.
- Replication messages between domain controllers within a site are uncompressed, which means that fewer CPU cycles are used on the domain controllers. Replication messages between domain controllers in different sites are compressed, which means that less network bandwidth is used.
- Replication between domain controllers within a site is triggered by the arrival of updates, which thereby reduces replication latency within a site. Replication between domain controllers in different sites is performed on a schedule, which thereby conserves network bandwidth between sites.

Sites are not tied in any way to the Active Directory namespace that is used by the domain directory partitions. The name of a domain directory object does not reflect the site or sites in which the object is stored.

Note In the Exchange Server directory service, sites are tied to the namespace.

The hierarchy in the Sites container reflects object names within the configuration directory partition, where site names do appear in the distinguished names of objects in the Configuration container hierarchy, as seen in Active Directory Sites and Services.

For information about planning sites and site topology for Active Directory, see "Designing the Active Directory Structure" in the *Deployment Planning Guide*.

Subnet-to-Site Mapping

When you plan your sites, you decide what subnets provide the best connectivity for replication and add them to the same site. You can use Active Directory Sites and Services to create subnets, and then you can create a site and associate the subnets with the site. You create a site by creating a site object in Active Directory and then defining a set of one or more subnets that belong to the site.

The determination of whether a computer is in a site is that its IP address maps to a subnet object in Active Directory. In the default directory, there is no default subnet object, so potentially a computer can be in the forest but have an IP subnet that is not contained in any site. For private networks, you can specify the network addresses that are provided by the Internet Assigned Numbers Authority (IANA). By definition, that range covers all of the subnets for the organization. However, where several class B or class C addresses are assigned, there would necessarily be multiple subnet objects that all mapped to the same default site.

To accommodate this situation, use the following subnets:

- For class B addresses, subnet 128.0.0.0/2 covers all class B addresses.
- For class C addresses, subnet 192.0.0.0/3 covers all class C addresses.

For information about subnets and subnet masks, see "Introduction to TCP/IP" in the *TCP/IP Core Networking Guide*. For information about creating subnets and assigning subnet masks, see Windows 2000 Server Help. For information about designing your network, see "Determining Network Connectivity Strategies" in the *Deployment Planning Guide*. For information about designing sites for Active Directory, see "Designing the Active Directory Structure" in the *Deployment Planning Guide*.

When to Define 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 per site.
- Deploy DNS servers on a site level.

The first domain controller in the forest is designated automatically as a Global Catalog server. When you create additional sites, you can use Active Directory Sites and Services to select the Global Catalog option in the properties for the NTDS Settings object of the server that you want to be the Global Catalog. Having a Global Catalog server in each site improves search performance because searches do not have to cross site boundaries. In addition, a Global Catalog server is required for logging on to the domain; if a connection between sites is not available, logging on is not possible.

Note If a Global Catalog server is not available in one site but there is another Global Catalog server in a remote site, the server in the remote site can be used for the logon process. If no Global Catalog is available in any site, the logon process proceeds with cached logon information.

The availability of DNS directly affects the availability of Active Directory. Clients rely on DNS to be able to find a domain controller, and domain controllers rely on DNS to find other domain controllers. As a general rule, you configure at least one DNS server in every site.

When you create sites and put servers into the sites, you connect the sites with site links and configure the links to reflect the network characteristics in terms of how often you want replication to occur on the link.

For information about planning sites and about domain controller, DNS server, and Global Catalog server placement, see "Designing the Active Directory Structure" in the *Deployment Planning Guide*. For information about Global Catalog servers, see "Active Directory Logical Structure" in this book.

Default Site

When you install Active Directory on the first domain controller in the site, an object named Default-First-Site-Name is created in the Sites container. The first domain controller is necessarily installed into this site. Subsequent domain controllers are either installed into the site of the source domain controller (assuming the IP address maps to the site) or installed directly into an existing site. When your first domain controller has been installed, you can rename Default-First-Site-Name to the name that you want to use for the site.

When you install Active Directory on subsequent servers, if alternative sites have been defined in Active Directory and the IP address of the installation computer matches an existing subnet in a defined site, the domain controller is added to that site. Otherwise, it is added to the site of the source domain controller.

For information about how sites are identified for new domain controllers, see "Active Directory Data Storage" in this book.

Server and Site Connections

Replication occurs between all domain controllers in the same site and between bridgehead servers in different sites. The replication connections between domain controllers in the same site are always created automatically by the KCC (and also can be created manually, although there is usually no need to do so). When you have more than one site, you create site links to connect them. When two sites are connected by a site link, the KCC also creates connections between the two bridgehead servers—one in each site for each domain that has domain controllers in the site. If there are multiple domains per site and each domain has domain controllers in more than one site, there are multiple bridgehead servers per site because domain controllers in the same domain must have connections that include only servers in that domain. The KCC selects the bridgehead servers automatically, or you can designate them manually if you want the same servers to be always used as bridgehead servers.

Replication transport protocols determine the manner in which replication data is transferred over the network media. Your network environment dictates the transports that you can use.

Server Objects

When you install Active Directory on a Windows 2000 Server–based computer, the installation process creates a server object that represents the domain controller. A server object is distinct from the computer object that represents the computer as a security principal. These objects are in separate directory partitions; the computer object represents the domain controller in the domain directory partition; the server object represents the domain controller in the configuration directory partition. The server object contains a reference to the associated computer object.

Server objects are children of site objects. The parent site of a server must contain the server subnet. If a domain controller's IP address or the subnet-to-site associations are changed after Active Directory is installed on the server computer, the domain controller does not change sites automatically. It must be moved to the new site manually if that site is the desired location.

Make sure that you create the site link before you move a domain controller to a different site. When replication between sites uses the SMTP transport, it is especially important that the domain controller first be properly configured within the site with RPC over IP connectivity.

Note When you make configuration changes that affect replication, the configuration changes replicate by using the old replication settings until they reach the systems where they take effect. Replication between sites does not occur until the following events occur:

- The site link is created somewhere in the directory.
- This change has replicated to the system in the site that is responsible for creating the intersite topology. (For more information about intersite topology generation, see "Automated Intersite Topology Generation" later in this chapter.)
- The KCC on that system has run and created the new connections.
- The new connections have replicated to the bridgehead server.
- The KCC on the bridgehead server has run and has translated the connections into replication partner relationships.
- Replication occurs on the bridgehead server.

Server Connections

The KCC creates connections for every domain controller that has a server object in the Sites container. These objects specify a one-way communication to the current system from another system. The connection object is a child of the replication destination's NTDS Settings object, and the connection object references the replication source domain controller in the *fromServer* attribute—that is, it represents the inbound half of a connection. The connection object contains a replication schedule and specifies a replication transport.

Connection objects are created in two ways:

- Automatically by the KCC.
- Manually by a directory administrator who is using Active Directory Sites and Services, ADSI Edit, or scripts.

A connection is unidirectional; a bidirectional replication connection is represented as two connection objects under two different NTDS Settings objects.

Ownership of Connection Objects

Connections that are created automatically by the KCC are "owned" by the KCC. Likewise, if an administrator creates a new connection or modifies an existing connection, that connection is owned by the administrator. If a connection object is not owned by the KCC, the KCC does not modify it or delete it. The implication of creating or modifying a connection object is that you want the object to remain as you define it. Ownership protects the object from being changed by the KCC.

Note Ownership of a connection object does not affect security access to the object; it determines only whether the KCC can modify or delete the object.

Ownership of a connection object is determined by the value in the *options* attribute on the connection object. If this value (*value* BITWISE-AND 1) equals 1, the KCC owns the connection. If you modify a KCC-generated connection, the *options* value changes. If you create a new connection object, the value of the *options* attribute is set to 0.

You can take ownership of a KCC-generated connection object by using Active Directory Sites and Services to modify the connection. Click the **Change** button to modify object properties; when the changes replicate, the *options* attribute value changes to reflect that the KCC does not own the object. For more information about creating a connection object, see Windows 2000 Server Help.

Note It is usually not necessary to create manual connections when the KCC is being used to generate automatic connections. The KCC automatically reconfigures connections as conditions change. Adding manual connections when the KCC is employed potentially increases replication traffic by adding redundant connections to the optimal set chosen by the KCC. When manually generated connections exist, the KCC uses them wherever possible.

Connection Schedule

Each connection object has a schedule that is set automatically by the KCC when it determines the best route for replication. The connection schedule controls how frequently periodic replication occurs. The connection schedule has a minimum increment of 15 minutes. The default setting for replication within a site is once per hour, which you can change by modifying the NTDS Site Settings object. By using Active Directory Sites and Services, you can set a schedule of None (no replication), once per hour (default), twice per hour, or four times per hour.

Note The Active Directory Sites and Services user interface confines the settings to once every 15 minutes for a specific hour or hours (from 1 hour to 24 hours) during the week. By using ADSI Edit or scripts, you can set replication on or off independently during each 15-minute window in the week.

The connection schedule receives its default from the schedule on the NTDS Site Settings object. When the KCC creates a new connection, the connection receives the schedule that is set on the NTDS Site Settings object. To override the default on the NTDS Site Settings object, you can manually set a schedule on the connection objects.

Note This manual override is effective only for manually created objects. If you attempt to update the schedule on a KCC-owned object, the KCC reverts to the schedule on the NTDS Site Settings object the next time it runs.

For a connection between sites, the KCC derives the schedule from the site link schedule, which controls how frequently the site link is active. The default setting for replication between sites is three hours, which you can change by modifying the associated site link object or objects.

Within a site, replication is triggered by changes; if a change occurs, replication is initiated (the domain controller with changes sends a notification to its replication partners that it has changes) after a default interval of five minutes. When no changes occur on the domain controller during the time allowed by the connection object schedule, replication is triggered (the domain controller requests changes from its replication partners) on the basis of the schedule (by default, once per hour).

The schedule is most important in replication between sites, where it is the primary mechanism that triggers replication. Because the granularity of the connection object schedule is 15 minutes, you cannot schedule replication between sites to occur any more frequently than once every 15 minutes.

Connecting Directory Partitions

Replication is performed between directory partition replicas, and connections are created between the servers to accommodate replication of as many partitions as the two servers have in common. Two domain controllers in the same forest always have at least two directory partitions in common: the configuration directory partition and schema directory partition. If the domain controllers are in the same domain, they also have a domain directory partition in common. If the domain controllers are Global Catalog servers for the same forest, they have partial domain directory partitions for every domain in common; if they are in the same domain, they have a full domain directory partition in common as well.

If a connection exists from one domain controller to another, it is used for replicating as many directory partitions as are common to the two servers. There is never a need for the KCC to create multiple connections linking the same two domain controllers in the same direction.

Site Links

When you have more than one site in your replication system, you must create links to connect the sites for replication. In Active Directory, a site link object identifies a set of sites that can be scheduled to communicate at uniform cost through some transport between sites. Site links communicate the connectivity that the KCC assumes between sites. Usually, you name site links for the sites that they connect. For example, if you have a site named "New York" and a site named "London," you might name your site link "NY-London."

Site links also specify the cost of the link, which controls the desirability of remote sites as sources of replication information. In addition, site links specify the schedule, how frequently periodic replication occurs over this link during the schedule window, and the behavioral options that might influence how replication occurs.

By default, site links for the same IP transport that have sites in common are bridged by site link bridges, which enable the KCC to treat the set of associated site links as a single route.

Bridgehead Servers

A "bridgehead" is a point where a connection leaves or enters a site. Servers that have connection objects for connections between sites are called *bridgehead servers.* Any server that has a connection object with a "from" server in another site is acting as a destination bridgehead. Any server that is acting as a source for a connection to another site acts as a source bridgehead. The KCC generates the connections and thereby causes the domain controllers that store the connections between sites to act as bridgeheads in the topology.

If you want to limit the KCC's choices of servers that it can designate as bridgeheads (that is, restrict the domain controllers in which the KCC can create connections between sites), select one or more domain controllers in the site that you want the KCC to always consider as "preferred" bridgehead servers. These servers are used exclusively to replicate changes collected from the site. If you specify preferred bridgehead servers, be aware of the following consequences:

- You limit the ability of the KCC to fail over to another bridgehead server when the designated server is down.
- You must assign one bridgehead server for each domain and writable directory partition combination in your forest, the cost of which might be prohibitive in a large organization.

Replication Transports

Replication transports provide the wire protocols that are 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. Therefore, replication between sites over SMTP is supported for only schema, configuration, and Global Catalog replication, which means that domains can span sites only when point-to-point, synchronous RPC is available between sites.

The Inter-Site Transports container provides the means for mapping site links to the transport that the link uses. When you create a site link object, you create it in either the IP container (which associates the site link with the RPC over IP transport) or the SMTP container (which associates the site link with the SMTP transport).

Synchronous vs. Asynchronous Communication

In the context of Active Directory replication, synchronous communication implies that after the destination domain controller sends the request for data, it waits for the source domain controller to receive the request, construct the reply, and send the reply before it requests changes from any other domain controllers. Therefore, at any particular time, a domain controller has a maximum of one synchronous request outstanding.

In the case of asynchronous replication, the destination domain controller does not wait for the reply, and it can have multiple asynchronous requests outstanding at any particular time. In synchronous transmission, therefore, the reply is received in a short amount of time; in asynchronous transmission, the reply is not necessarily received in a short time.

Transport for Replication Within a Site

All replication within a site occurs over synchronous RPC over IP transport. The focus for replication within a site is the fast, uncompressed delivery of updates. Replication events occur more frequently within a site than between sites, and the overhead of compression would be inefficient over fast connections.

Note Replication between sites is compressed.

By default, RPC-based replication uses dynamic port mapping. When connecting to an RPC endpoint during Active Directory replication, the RPC run time on the client contacts the RPC endpoint mapper on the server at a well-known port (port 135). The server queries the RPC locator on this port to determine what port has been assigned for Active Directory replication on the server. This query occurs whether the port assignment is dynamic (the default) or fixed. The client therefore never needs to know which port to use for Active Directory replication.

Note An endpoint comprises the protocol, local address, and port address.

Active Directory registers the endpoint when it starts, and it uses either a dynamically assigned port or a specified port, whichever you have configured. To specify a fixed port for routed environments or where port filtering is employed, you can add or modify the **TCP/IP Port** entry in HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS\Parameters to configure Active Directory to register a specific port with the endpoint mapper. This value can be any valid TCP/IP port number.

▶ **To set the TCP/IP Port entry**

1. In a registry editor, navigate to HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS\Parameters.
2. Double-click the **TCP/IP Port** entry, and assign a valid port number.
3. Close the registry editor.

Caution Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. There are programs available in Control Panel or Microsoft Management Console (MMC) for performing most administrative tasks. These programs provide safeguards that prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your system. Registry editors bypass the standard safeguards that are provided by these administrative tools. Modifying the registry is recommended only when no administrative tool is available. Before you make changes to the registry, it is recommended that you back up any valuable data on the computer. For instructions about how to edit registry entries, see Help for the registry editor that you are using. For more information about the registry, see the *Microsoft Windows 2000 Resource Kit* Technical Reference to the Windows 2000 Registry (Regentry.chm).

Transports for Replication Between Sites

Windows 2000 supports two default transports for replication between sites:

- RPC over TCP/IP (referred to as "IP" in administrative tools), which enables low-speed, point-to-point, synchronous replication between all directory partitions.
- SMTP, which enables low-speed, asynchronous replication between the schema, configuration, and Global Catalog directory partitions, but not between domain directory partitions.

When sites are on opposite ends of a WAN link (or the Internet), it is not always desirable—or even possible—to perform synchronous, RPC-based directory replication. In some cases, the only method of communication between two sites is e-mail. To support such configurations, replication must be possible across asynchronous, store-and-forward transports such as SMTP.

SMTP replication substitutes mail messaging for the RPC transport. The message syntax is the same as for RPC-based replication. There is no change notification for SMTP-based replication, and scheduling information on the site link object is used as follows:

- SMTP replication ignores the **Replication Available** and **Replication Not Available** settings on the site link schedule in Active Directory Sites and Services (the information that indicates when these sites are connected).
- SMTP replication uses the replication interval to indicate how often the server requests changes. The interval (**Replicate every ____ minutes**) is set in hourly intervals on the **General** tab in site link **Properties** in Active Directory Sites and Services.

These settings combine to form the replication schedule on the connection object. The underlying SMTP messaging system is responsible for message routing between SMTP servers.

The Intersite Messaging (ISM) service allows for multiple transports to be used as add-ins to the ISM architecture. ISM provides services to the KCC in the form of querying the available replication paths. ISM enables messaging communication that can use SMTP servers other than those that are dedicated to processing e-mail applications such as Exchange Server.

Comparison of SMTP and RPC Replication

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. (For information about configuring the replication packet size, see "Replication Packet Size" later in this chapter.)
- Active Directory can have only a single change request outstanding for a specific directory partition to a specific replication partner.
- The response data (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. For example, SMTP can be used by companies that have a network backbone that is not based on TCP/IP, such as companies that use an X.400 backbone.

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.

Benefits of SMTP Replication Between Sites

Although SMTP replication usually is slower than RPC, there are conditions that call for SMTP replication exclusively, and others that favor it, such as the following:

- RPC-based replication is currently synchronous, whereas SMTP is asynchronous. Where bandwidth is limited, it can be disadvantageous to force an entire transaction to complete before another can begin. With SMTP, several transactions can be processing simultaneously so that each transaction is receiving some attention most of the time, as opposed to no attention for prolonged periods, which can result in RPC time-outs.
- SMTP traffic can be secured, monitored, and managed across a WAN environment.
- Where end-to-end online IP connectivity is impossible—for example, A can communicate with B, and B can communicate with C, but A can never communicate with C directly—mail can be used and routed A to B, B to C, C to B, or B to A.

Replication Packet Size

By default, packet sizes are computed on the basis of memory size unless you have more than 1 gigabyte (GB) or less than 100 megabytes (MB) of memory. You can override these memory-based values in the registry.

To adjust the default size of the packets that transport Active Directory replication data, you can modify or add entries to the following registry path with the REG_DWORD data type: HKEY_LOCAL_MACHINE\System \CurrentControlSet\Services\NTDS\Parameters. These entries determine the maximum number of objects per packet and maximum size of the packets.

- For RPC replication within a site:
  - **Replicator intra site packet size (objects)**

    Range: >=1
  - **Replicator intra site packet size (bytes)**

    Range: >=10 KB
- For RPC replication between sites:
  - **Replicator inter site packet size (objects)**

    Range: >=1
  - **Replicator inter site packet size (bytes)**

    Range: >=10 KB

- For SMTP replication between sites:
  - **Replicator async inter site packet size (objects)**

    Range: >=1
  - **Replicator async inter site packet size (bytes)**

    Range: >=10 KB

If the preceding registry entries are not set, the system limits the packet size as follows:

- The packet size in bytes is 1/100th the size of RAM with a minimum of 1 MB and a maximum of 10 MB.
- The packet size in objects is 1/1,000,000th the size of RAM, with a minimum of 100 objects and a maximum of 1,000 objects.

There is one exception: the value of the **Replicator async inter site packet size (bytes)** entry is always 1 MB. Many mail systems limit the amount of data that can be sent in a mail message (2 MB to 4 MB is common), although most Windows-based mail systems can handle large 10-MB mail messages.

Caution Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. There are programs available in Control Panel or Microsoft Management Console (MMC) for performing most administrative tasks. These programs provide safeguards that prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your system. Registry editors bypass the standard safeguards that are provided by these administrative tools. Modifying the registry is recommended only when no administrative tool is available. Before you make changes to the registry, it is recommended that you back up any valuable data on the computer. For instructions about how to edit registry entries, see Help for the registry editor that you are using. For more information about the registry, see the *Microsoft Windows 2000 Resource Kit* Technical Reference to the Windows 2000 Registry (Regentry.chm).

Managing Replication Between Sites

Although replication within sites is optimized for LAN connectivity and requires little or no management, you have control over when and how replication between sites occurs. You want to maximize efficiency and minimize cost, and there are decisions that must be made on the basis of your network environment, physical location, and business needs. The KCC generates the intersite topology automatically, but the settings on the site links are the factors that the KCC considers in the process.

When multiple sites participate in the replication topology of domain controllers in the same forest, the default intersite topology is a least-cost spanning tree, where "cost" is administratively set to favor various routes. Replication between sites can occur synchronously by RPC over IP transport or asynchronously by SMTP over IP transport.

Note A spanning tree algorithm is applied to network connections to eliminate redundant routes and thereby reduce consumption of expensive WAN network bandwidth.

Planning Replication Between Sites

Replication within sites requires little or no planning because it is fully automatic. However, when you have multiple sites, the following steps can be used to plan how replication occurs between them:

1. Identify sites that are well connected through backbones, and create low-cost site links between these sites.
2. Identify sites that are all connected to each other with a comparable transport, and create medium-cost site links between them—for example, full mesh links (remote sites that are connected over telecommunication links), frame relay cloud links (a point-to-point system that uses a private virtual circuit), medium area network (MAN) links with T1 connections.
3. Identify remaining WAN links.
4. Create a site link for each pair of sites that cross a WAN link.
5. Create a schedule that meets user needs.

   Avoid high-frequency times. Site links must have windows of time in common that are available for routing.

For information about planning sites and site topology for Active Directory, see "Designing the Active Directory Structure" in the *Deployment Planning Guide*.

Managing Site Links

Connection objects are created automatically by the KCC for replication both within a site and between sites. For connection objects to be created between two sites, however, you must manually create a link that connects the two sites. These links, implemented through site link objects in Active Directory, identify the transport protocol and scheduling required to replicate between two sites. Administrators use Active Directory Sites and Services to create the site links, and the KCC creates the connections accordingly when it generates the intersite topology.

Site link objects can be created in two transport-specific containers within the Inter-Site Transports container in Active Directory Sites and Services. By creating the link in one or the other container, you associate the link with the respective replication transport. The Inter-Site Transports container is a child of the Sites container, and it also has child containers:

- The IP container, which contains site link objects that use RPC over IP synchronous replication transport.
- The SMTP container, which contains site link objects that use SMTP over IP asynchronous replication transport.

When the KCC configures the connection objects for replication between sites, it takes the settings on the site link object into account to create the best connection. For example, one of the site link settings is the cost of the connection. When it has a choice, the KCC chooses a remote site whose link has the lowest cost when it forms connections.

For IP transport, a typical site link connects only two sites and corresponds to an actual WAN link. An IP site link connecting more than two sites might correspond to an ATM backbone that connects, for example, more than two clusters of buildings on a large campus or connects several offices in a large metropolitan area that are connected by leased lines and IP routers.

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. Otherwise, it cannot replicate with domain controllers in any other site, so the directory is disconnected. Therefore, if there is more than one site in the forest, you must configure at least one site link.

Figure 6.6 shows two sites that are connected by one site link. A single domain has domain controllers in both sites. When topology generation occurs, connection objects between bridgehead servers in the site are created by the KCC and replication occurs according to the settings on the site link.

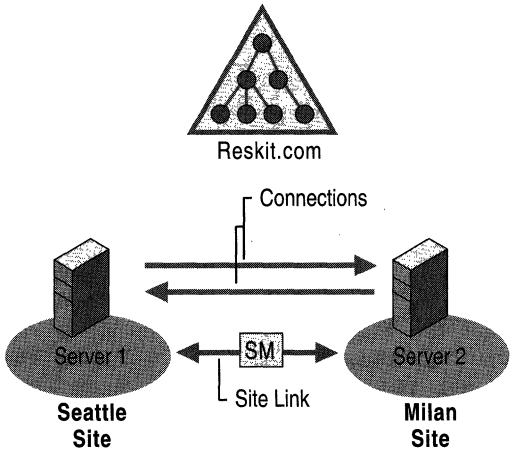


Figure 6.6 Two Sites That Are Connected by a Site Link

Figure 6.7 shows three sites connected by two site links. By default, site links are transitive. Therefore, replication messages can flow from the Atlanta site, through the Seattle site, and on to the Milan site. In this scenario, because the Seattle site contains a full replica of reskit.com, there is no need for direct replication between Milan and Atlanta; all replication between them is transitive through the Seattle site.

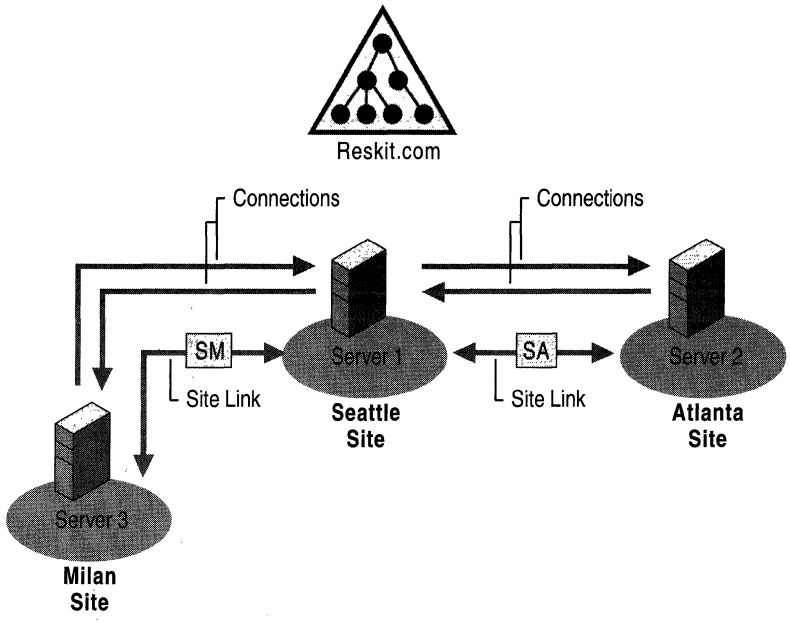


Figure 6.7 Three Sites That Are Connected by Two Site Links

Site Link Settings

In Active Directory Sites and Services, the **General** tab in the **Site Link Properties** dialog box contains the following options for configuring site links to control the replication topology:

- A list of two or more sites to be connected.
- A schedule that determines during what time periods the link is available for replication. For example, you might schedule a site link for a dial-up line to be available during off hours (when telephone rates are low) and unavailable during high-cost regular business hours.

Note Scheduling information is ignored by site links that use SMTP transports; the mail is stockpiled and then exchanged at the times that are configured for your mail infrastructure.

- Cost, a single numeric cost factor associated with communication over the link. Higher cost numbers represent more expensive messages. For example, sites that are connected by low-speed or dial-up connections would have high-cost site links between them. Sites that are well connected through backbone lines would have low-cost site links. Where multiple routes or transports exist between two sites, the least expensive route and transport are used.

- Period, an interval in minutes that determines how often replication can occur (default is 180 minutes, or 3 hours). The minimum period is 15 minutes.

The site link settings let you control replication topology and timing independently:

- You control topology by setting the costs on site links. In a common scenario, you might set cost = 1 for site links that are part of your backbone network, and cost = 100 for site links that correspond to slow connections to branch offices. Setting costs in this way ensures that a branch office replicates with a domain controller in a site that is part of the backbone, never directly with a second branch office. These cost numbers have no influence on the replication period.
- You control the replication period by setting an interval in minutes on site links.
- You control link availability by setting a schedule on site links. You might use the default (100 percent available) schedule on most links, but block replication traffic during peak business hours on links to certain branches. By blocking replication, you give priority to other traffic, but you also increase replication latency.

Cost Factor

You assign cost values to site links to favor inexpensive connections over expensive connections. The costs of providing bandwidth is a factor that is to be taken into account when you define site boundaries; it is recommended that these costs be defined on a sitewide basis. Cost is usually based not only on the total bandwidth of the link but also on the availability, latency, and monetary cost of the link.

For example, a 128-kilobits per second (Kbps) permanent link might be assigned a lower cost than a dial-up 128-Kbps dual ISDN link because the dial-up ISDN link has replication latency-producing delay that occurs as the links are being established or removed. Furthermore, in this example, the permanent link might have a fixed monthly cost, whereas the ISDN line is charged according to actual usage. Because the company is paying up-front for the permanent link, the administrator might assign it a lower cost than the ISDN line. The ISDN connections in this example only add extra monetary cost to the already paid-for permanent line.

Table 6.2 shows the speeds for different types of networks; you can use these network speeds to estimate cost.

Table 6.2 Network Speeds for Estimating Cost

Network Type	Speed
Very slow	56 Kbps
Slow (typical in Europe)	64 Kbps
ISDN	64 Kbps or 128 Kbps
Frame relay	Variable rate, commonly between 56 Kbps and 1.5 megabits per second (Mbps)
T1	1.5 Mbps
T3	45 Mbps
Asynchronous Transfer Mode (ATM)	Variable rate, commonly between 155 Mbps and 622 Mbps
Gigabit Ethernet	1 gigabit per second (Gbps)

Before you assign any costs, define a model for your WAN. On the basis of the cost plus other factors (availability and replication latency), you can establish a set of costs that can be implemented throughout the forest. Where a cost is assigned, it must always mean the same thing in any other place where the same cost is assigned. Table 6.3. shows an example of the cost breakdown in a forest for a network where a high speed has a lower cost.

Table 6.3 Example of Cost Breakdown for a Forest on a High-Speed Network

Network Type	Cost Value
T1 to backbone	1
56-kilobit link	500
Branch office	1,000
International link	5,000

The **Cost** setting on a site link provides a relative value for the cost of communication between all sites that are part of the link. (By default, site link settings are *transitive* between the sites that they connect.) For example, if you create an IP site link object XYZ that connects the sites X, Y, and Z with cost 5, you establish that an IP message can be sent between all pairs of sites (X to Y, X to Z, Y to X, Y to Z, Z to X, Z to Y) with cost 5.

Note By default, all site links are transitive; that is, all site links for a specific transport implicitly belong to a single site link bridge for that transport. (For information about site link bridges, see "Site Link Bridges" later in this chapter.) 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 are considered nontransitive and you can configure site link bridges.

The KCC determines the least-cost path from each site to every other site for each directory partition. The KCC then reviews the comparison of multiple paths to and from every destination and computes the spanning tree of the least-cost path.

For information about IP routing, see "Determining Network Connectivity Strategies" in the *Deployment Planning Guide*.

Replication Period

For each site link object, you can specify a value for the replication period, which determines how often replication occurs over the site link during the time that the schedule allows. For example, if the schedule allows replication between 02:00 hours and 04:00 hours, and the replication period is set for 30 minutes, replication can occur up to four times during the scheduled time.

The default replication period is 180 minutes, or 3 hours. When the KCC creates a connection between a domain controller in one site and a domain controller in another site, the replication period of the connection is the maximum period along the minimum-cost path of site link objects from one end of the connection to the other.

Schedule

In the case of RPC transport between sites, the replication between sites can be scheduled. Site links are associated with a schedule, which opens one or many windows for when replication can occur. If replication goes through multiple site links, there must be at least one common window; otherwise, the connection is treated as not available. For example, if site link 1 has a schedule (window of opportunity) of 18:00 hours to 24:00 hours and site link 2 has a schedule (window of opportunity) of 17:00 hours to 20:00 hours, the resulting window of opportunity is 18:00 hours through 20:00 hours, which is the intersection of site link 1 and site link 2.

Replication Path

The path that replication takes between sites is computed from the information on the site link objects. When a change is made to a site link setting, the following events must occur before the change takes effect:

- The site link change must replicate to each topology-generating system by using the previous topology.
- The KCC must run on the topology-generating systems.

As the path of connections is transitively figured through a set of site links, the attributes (settings) of the site link objects are "aggregated" along the path as follows:

- Costs are added together.
- The replication period is the maximum of the intervals that are set along the path.
- The options, if any are set, are "ANDed" together.

Note Options are the values of the *options* attribute on the site link object. The value of this attribute determines special behavior of the site link, such as reciprocal replication and intersite change notification. (For more information about these behaviors, see "Reciprocal Replication" and "Change Notification" later in this chapter.)

Thus the site link schedule is the "overlap" of all of the schedules of the subpaths. If none of the schedules overlap, the path is not used.

Site Link Settings and Replication Latency

The schedule determines the time intervals during which the site link is available, and the replication period determines how often replication can occur during those intervals. The interaction of these factors determines the replication latency. For sites where the maximum replication period within the site is 15 minutes, the worst-case, end-to-end replication latency from a source domain controller to a destination domain controller in a remote site is the sum of the replication period settings for the connections between the source and destination sites, plus 15 minutes for each site in the path (including the source and destination sites). This sum assumes that the RPC transport is used between sites and that the required physical connections are available.

Interaction of Schedule and Replication Period

When multiple site links are required to complete replication for all sites, the replication periods on each link combine to affect the entire length of the connection between sites. In addition, when schedules on each link do not coincide, replication can occur only during the window of opportunity where the schedules intersect.

Suppose that site A and site B have site link AB, and site B and site C have site link BC. When a domain controller in site A replicates with a domain controller in site C, it can do so only as often as the maximum period set for site link AB and site link BC allow. Table 6.4 shows the site link settings that determine how often and during what times replication can occur between domain controllers in site A, site B, and site C.

Table 6.4 Replication Period and Schedule Settings for Two Site Links

Site Link	Replication Period	Schedule
AB	30 minutes	12:00 hours to 04:00 hours
BC	60 minutes	01:00 hours to 05:00 hours

Given the settings in Table 6.4, a domain controller in domain A can replicate with a domain controller in site B according to the AB site link schedule and period, which is once every 30 minutes between the hours of 12:00 and 04:00. However, assuming that there is no site link AC, a domain controller in site A can replicate with a domain controller in site C once every 60 minutes, which is the greater of the two replication periods, and between the hours of 01:00 and 04:00, which is where the schedules on the two site links intersect.

Schedule Implementation

The times that you can set in the **Schedule** setting on the site link are in one-hour increments. For example, you can schedule replication to occur between 00:00 hours and 01:00 hours, between 01:00 hours and 02:00 hours, and so forth. However, each block in the actual connection schedule is 15 minutes. For this reason, when you set a schedule of 01:00 hours to 02:00 hours, you can assume that replication is queued at some point between 01:00 hours and 01:14:59 hours.

Note RPC synchronous inbound replication is serialized so that if the server is busy replicating this directory partition from another source, replication from a different source does not begin until the first synchronization is complete. SMTP results are processed serially by order of arrival.

Specifically, a replication event is queued at time $t + n$, where t is the replication period that is applied across the schedule and n is a pseudo-random number between 1 minute and 15 minutes, inclusive. For example, if the site link indicates that replication can take place between 02:00 hours and 07:00 hours (inclusive), and the replication period is 2 hours (120 minutes), t is 02:00 hours, 04:00 hours, and 06:00 hours. A replication event is queued between 02:00 hours and 02:14:59 hours, and another replication event is queued between 04:00 hours and 04:14:59 hours. Assuming that the first replication event that was queued is complete, another replication event is queued between 06:00 hours and 06:14:59 hours. If the synchronization took longer than two hours, the second synchronization would be ignored because there is already an event in the queue.

Note The replication queue is shared with other events, and the time at which replication takes place is approximate. Duplicate replication events are not queued for the same directory partition and transport.

For information about how to create and configure site links, see Windows 2000 Server Help.

Preferred Bridgehead Server Configuration

Bridgehead servers must be able to accommodate more replication traffic than non-bridgehead servers, and you might want to choose which servers are to carry out this task. Knowing which system is acting as a bridgehead also can be useful for troubleshooting.

When you manually configure a single domain controller as the bridgehead server for a site, the KCC uses only that server. When multiple domain controllers in a site are configured to be preferred bridgehead servers, the KCC ultimately selects one of these servers on the basis on other variables.

Depending on what transports are available, which directory partitions must be replicated, and the availability of Global Catalog servers, multiple bridgehead servers might be required to replicate full copies of data from one site to another.

Suppose that there are two sites, site A and site B, and each site has a single domain controller from each of two domains, domain X and domain Y. In this case, the only way that replication of the respective domain directory partitions can occur between the two sites is if the domain controllers for domain X and domain Y are selected as bridgehead servers in each site. Therefore, if there is a single domain controller for a specific domain in a specific site, that domain controller must be a bridgehead server in its site because it can replicate domain data to only a domain controller in its own domain. In addition, that single domain controller must be able to connect to a bridgehead server in the alternative site that also holds the same domain directory partition.

If you want the KCC to consider certain domain controllers over others as bridgehead servers, you can specify a domain controller and an associated transport to indicate this preference by using the server object properties in Active Directory Sites and Services.

You specify the server when you select its server object, and then you add each transport for which the selected domain controller is a preferred bridgehead (IP for RPC over IP, or SMTP for SMTP over IP). If you select more than one server for a specific transport that can replicate a particular domain directory partition, the KCC chooses one arbitrarily. For information about how to specify preferred bridgehead servers, see Windows 2000 Server Help.

Bridgehead Server Failure

When a bridgehead server goes down, potentially it can cut off replication between this site and the other site. In most cases, the KCC selects a different bridgehead server automatically.

Failure of KCC-Selected Bridgehead Servers

Unless you specify a preferred bridgehead server or servers, the KCC selects them automatically. When the KCC selects bridgehead servers automatically and the current bridgehead server fails, after a time interval (the point at which a failure has occurred and the time since the last successful replication is greater than 2 hours), the KCC selects another bridgehead server to take its place. If all potential bridgehead servers are unavailable, the KCC logs an event that describes the condition.

Failure of Preferred Bridgehead Servers

If you explicitly set a preferred bridgehead server or servers and none is available, the KCC does not select alternative bridgehead servers automatically. In this case, the KCC logs an error message that states that you have designated preferred bridgehead servers that can replicate a specific directory partition, but none of them is available.

Replacement of a Failed Preferred Bridgehead Server

If you want the KCC to be able to fail over to other domain controllers but there are no other preferred bridgehead servers available, you must do one of the following *at a domain controller in each site*:

- Add new domain controllers as preferred bridgehead servers for the corresponding directory partitions, site, and transport. (If there is more than one domain represented in the site, you must add a preferred bridgehead server for the correct domain.)
- Remove all preferred bridgehead designations that you have made for the corresponding site and transport, in which case the KCC selects new ones automatically.

Important The KCC creates only inbound connections. A bridgehead server cannot create an outbound connection to another bridgehead server. For this reason, unless there is already a functioning replication path to the other site, changes to preferred bridgehead server status must be made on a domain controller in each affected site so that inbound connections are created in each site.

If the only preferred bridgehead server that is available for a specific directory partition and transport fails and you want to assign a new bridgehead server to replace it, you must add the new bridgehead server twice—once on a domain controller in the site of the failed bridgehead server, and once on a domain controller in the site on the other end of the affected site link. This process might involve two administrators if the site locations are far away from each other.

Note If there are preferred bridgehead servers available and you want to add another preferred bridgehead server in the site, you do not have to add the server in both sites because the change replicates to the other site through the currently available bridgehead servers.

If you remove all preferred bridgehead servers so that the KCC can select bridgehead servers automatically, you must remove them for each domain directory partition and for each transport on a domain controller in each affected site.

Figure 6.8 shows the connections between bridgehead servers in two sites. The bridgehead server in site A is a preferred bridgehead server.

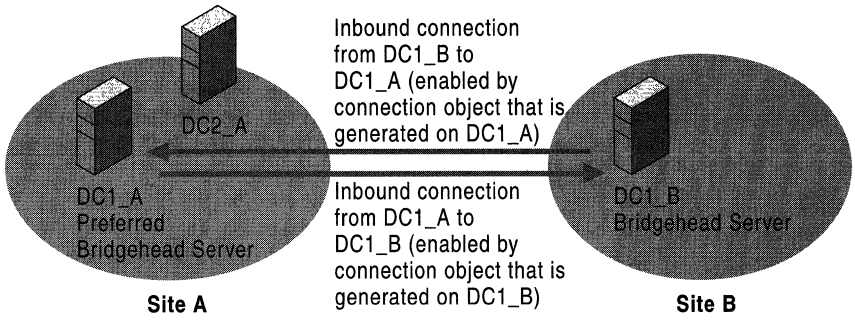


Figure 6.8 Two Sites and Two Bridgehead Servers with Inbound Connections from Each Other

If the preferred bridgehead server in site A fails, the bridgehead server in site B loses its inbound connection from the failed server and, thus, is disconnected from the site. If you assign a server to replace the failed bridgehead server in site A, the new bridgehead server creates only inbound connections. This change cannot replicate to site B because there is no inbound connection at the bridgehead server in site B. Figure 6.9 shows the connection that is created when you add a new bridgehead server in site A.

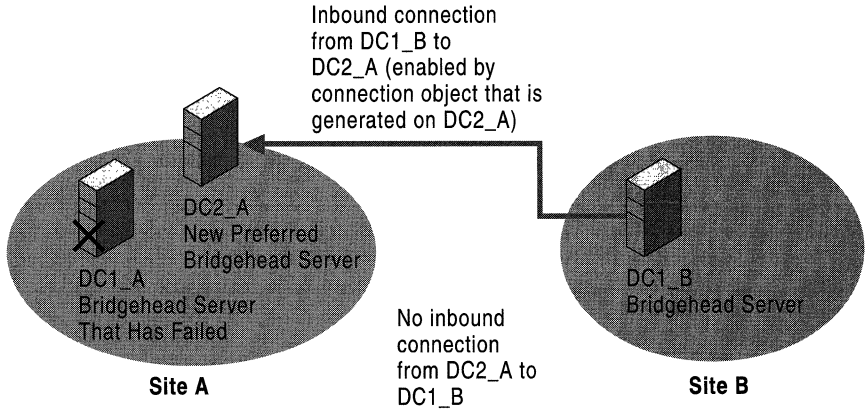


Figure 6.9 New Inbound Connection from the Existing Server with No Inbound Connection from the New Bridgehead Server

Until you add the new site A bridgehead server to a domain controller in site B, there is no inbound connection possible from site A to site B, even though you have added the bridgehead server in site A. The reason is that the KCC creates only inbound connections and has no knowledge of the new server.

In Figure 6.10, the new bridgehead server for site A has been added to a domain controller in site B. The figure shows the new inbound connection that results. The sequence of events is as follows:

1. An administrator in site B goes to a domain controller and adds DC2_A as a preferred bridgehead server by selecting DC2_A in the Servers container under site A in Active Directory Sites and Services. He or she then adds the server to the list of preferred bridgehead servers for the appropriate transport.
2. The bridgehead server designation is replicated (as a change to the configuration directory partition) to all domain controllers in site B, including the bridgehead server (DC1_B).
3. DC1_B generates an inbound connection object from DC2_A, which completes the two-way replication route between bridgehead servers in site A and site B. At this point, replication is possible between the two bridgehead servers.

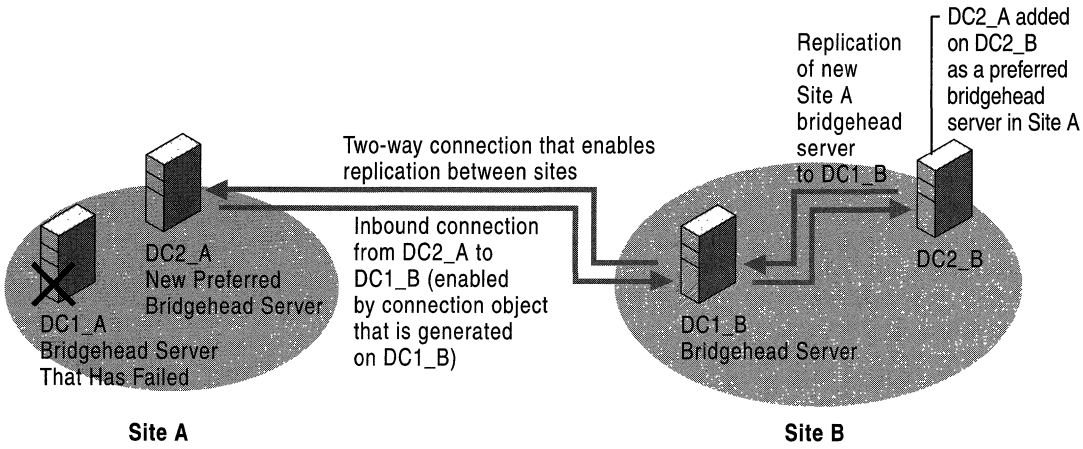


Figure 6.10 Two-Way Connection Between Sites After Adding the New Site A Bridgehead Server in Site B

Failure of a Preferred Bridgehead as a Result of Incorrect Configuration

If you have configured preferred bridgehead servers but none of them is capable of replicating a directory partition that must be replicated, the KCC logs an event for a configuration error. (A domain that has servers in the site is not represented by a bridgehead server.) The KCC then proceeds to select an alternative in the same manner as if no preferred bridgehead servers are configured.

Site Link Bridges

A site link bridge object represents a set of site links, all of whose sites can communicate through some transport. A site link bridge usually corresponds to a router (or a set of routers) in an IP network.

Note If no bridgehead server that is capable of the site link bridge transport is available in two linked sites, a route is not available.

By default, all site links that you create are bridged ("transitive"); all site links for a specific transport implicitly belong to a single site link bridge for that transport. Therefore, in the common case of a fully routed IP network, you do not need to configure any site link bridges. Figure 6.11 shows a case where three sites are connected by two site links and the site link bridge allows connections to be created between two sites that are not connected by an explicit site link.

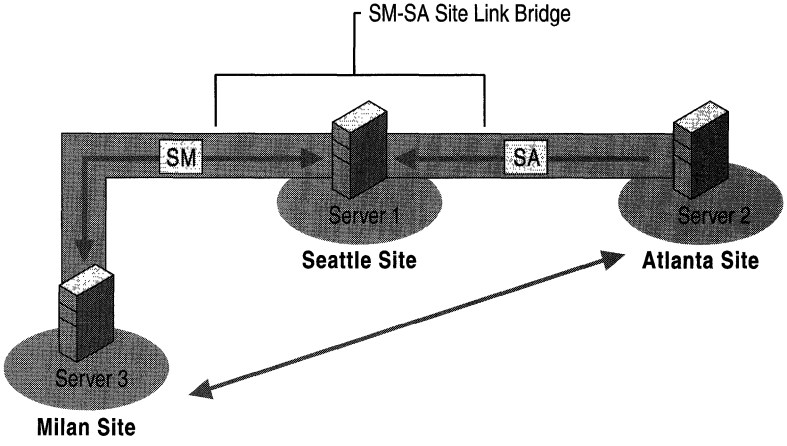


Figure 6.11 Site Link Bridge That Enables Connections Between Milan and Atlanta Sites

If your IP network is not fully routed, you can turn off the **Bridge all site links** for IP transport (on the **General** tab in the IP transport object property sheet or SMTP transport object property sheet). In this case, all IP site links are considered nontransitive, and you can configure site link bridges to model the actual routing behavior of your network.

Managing Site Link Bridges

You create a site link bridge object for a specific transport by specifying two or more site links for the specified transport.

To understand what a site link bridge means, consider the following example:

- Site link SM connects the Seattle site and the Milan site over IP with cost 4.
- Site link SA connects the Seattle site and the Atlanta site over IP with cost 3.
- There is no site link between the Milan site and the Atlanta site.
- Site link bridge SM-SA connects site link SM and site link SA.

In this simple example, the site link bridge SM-SA implies that an IP message can be sent from the Milan site to the Atlanta site with cost 4+3 = 7.

Each site link in a bridge must have at least one site in common with another site link in the bridge. Otherwise, the bridge cannot compute the cost from sites in one link to the sites in other links of the bridge. For example, if you have four sites (W, X, Y, and Z), a site link WX that connects W and X, and a site link YZ that connects Y and Z, a site link bridge that connects WX and YZ serves no purpose.

Separate site link bridges, even for the same transport, are independent. To illustrate this independence, the following objects are added to the Milan-Seattle-Atlanta example:

- Site link DA connects the Detroit site and the Atlanta site over IP with cost 2.
- Site link bridge DA-SA connects site link DA and site link SA.

The presence of this additional bridge means that an IP message can be sent from the Seattle site to the Detroit site with cost 2 + 3 = 5; but it does not imply that an IP message can be sent from the Detroit site to the Milan site with cost 2 + 3 + 4 = 9. In almost all cases, you use a single site link bridge to model the entire IP network.

Performance Considerations

Any network that you can describe by a combination of site links and site link bridges also can be described by site links alone. The advantage to bridging all site links is that your network description is much smaller and easier to maintain because you don't need a site link to describe every possible path between pairs of sites.

However, when the number of sites exceeds 200, periods of high CPU activity occur every 15 minutes when the KCC runs. The **Bridge all site links** setting creates a single bridge for the entire network, which generates more routes that must be processed than if site link bridges are not used or are applied selectively.

For example, under the conditions identified in Figure 6.11, the following factors affect KCC performance:

- The KCC recognizes the explicit site links between Atlanta and Seattle and between Seattle and Milan.
- With the site link bridge in place, the KCC also must consider the implicit paths between Milan and Atlanta as a single path with a combined cost.
- Where the site links represent hops between domain controllers in the same domain, replication changes flow transitively without the site link bridge. When a site link bridge is in place, the KCC must compute transitivity between the sites.
- The site link bridge adds more connection possibilities, which are usually eliminated because they have a higher aggregated cost. The KCC must spend extra time and cycles to eliminate these connection possibilities.

Thus, in a large network where processing time is a concern, there are performance advantages to turning off **Bridge all site links** and configuring site link bridges only where they are advantageous. If you still experience delays, the next step is to replace the bridges with a large number of sites that have explicit site links. For more information about KCC scaling recommendations, see the Microsoft Knowledge Base link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. Search the Knowledge Base by using the keyword "Q244368."

Global Catalog Replication

A Global Catalog server is a domain controller that stores specific information about all objects in the forest. The Global Catalog is required for the logon process, so it is best to have at least one Global Catalog server per site. The Global Catalog stores a replica of every directory partition in the forest: It stores full replicas of the schema and configuration directory partitions, a full replica of the domain directory partition for which the domain controller is authoritative, and partial replicas of all other domain directory partitions in the forest. When an *attributeSchema* object has the *isMemberOfPartialAttributeSet* attribute set to TRUE, the attribute is replicated from the domain directory partition to the corresponding directory partition replicas on all authoritative domain controllers and also to all Global Catalog servers.

Figure 6.12 depicts logical directory partitions in the Active Directory database of a Global Catalog server. (The database itself, Ntds.dit, is not actually partitioned.) The top three segments represent directory partitions that are full replicas for the domain controller. The bottom three segments represent directory partitions that are partial replicas for the Global Catalog.

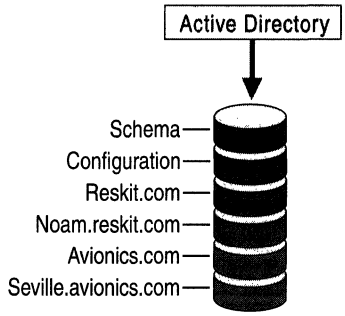


Figure 6.12 Directory Database on a Domain Controller That Is a Global Catalog

Global Catalog servers request updates from a source domain controller for each domain directory partition in the forest. This source domain controller can be either a normal domain controller or another Global Catalog server.

Figure 6.13 shows the partner associations between directory partitions in two Global Catalog servers that are authoritative domain controllers for different domains. As is true for all domain controllers, the Global Catalog uses a single topology to replicate the schema and configuration directory partitions, and it uses a separate topology for each domain directory partition. Replication is two-way between the domain directory partitions.

One server is authoritative for avionics.com; the other server is authoritative for reskit.com. As such, the avionics.com domain controller can be the source for replication to the partial replica of avionics.com on the reskit.com Global Catalog server, and the reskit.com domain controller can be the source for replication to the partial replica of reskit.com on the avionics.com Global Catalog server. The connection arrows indicate the one-way flow of replication from the read-write sources to the read-only destinations. In the case of the noam.reskit.com domain, neither domain controller is authoritative for that domain, so the Global Catalog servers replicate these partial replicas to and from each other.

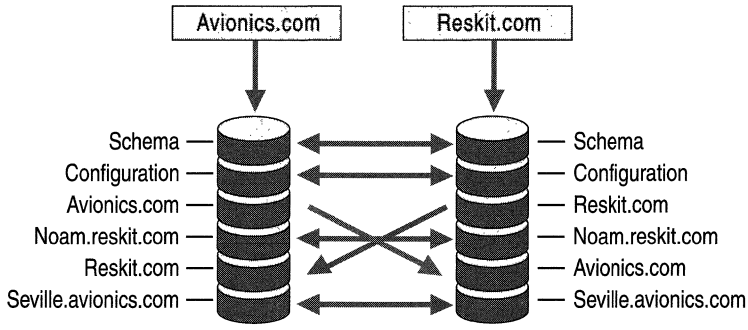


Figure 6.13 Directory Partition Connections Between Two Global Catalog Servers in Different Domains

Replication of Removed Properties

When you want to remove an attribute from the Global Catalog, you must set its *isMemberOfPartialAttributeSet* value to FALSE. The attribute then is removed from the Global Catalog immediately after the next replication cycle.

Added Attributes

If an attribute has been added to the partial attribute set, the domain controller must replicate the value of this attribute for each of its partial directory partition replica objects. This is accomplished by performing a full synchronization across all of the Global Catalog's replication connections. If the partial directory partition replica can be synchronized over an RPC connection, the domain controller attempts a full synchronization over an RPC connection before it uses any other connections; if full synchronization is completed, the up-to-dateness vector that it creates optimizes later full synchronization on other connections.

Caution Caution should be exercised when you add attributes to the Global Catalog attribute set because doing so causes full synchronization of the Global Catalog on all Global Catalog servers in the forest. Although interruption of service does not occur, this replication causes higher bandwidth consumption than is required for usual day-to-day replication. The resulting bandwidth consumption for each Global Catalog server is equivalent to that caused by promoting a regular domain controller to the role of Global Catalog server.

Universal Group Replication

A *universal group* can have members from any domain in the forest, and thus the membership for universal groups cannot be stored on every domain controller (each domain controller stores objects for only one domain), but only on Global Catalog servers, which store every object in the forest. Therefore, the Global Catalog servers are the only domain controllers that can enumerate the membership of a universal group. For this reason, a Global Catalog server is required for logging on to domains that use universal groups. During the logon process, the Global Catalog enumerates the membership of universal groups and attaches any found membership to the security token of the user. Other types of groups (global and local) are not enumerated by the Global Catalog; only the group object name is listed. The enumeration of global and local groups is the responsibility of the resource domain controller. For replication, this arrangement means that the replication of global and local group memberships is not required by Global Catalog servers, which significantly reduces replication traffic.

Scenarios for Replication Between Sites

The following scenarios combine different approaches to using site links and site link bridges.

One Domain Spanning Multiple Sites

In Figure 6.14, two site links have been defined, NYC-SEA and SEA-DFW. In this environment, the default behavior of transitive site links has been disabled because the network is not fully routed, so there is no site link bridge. For this reason, the KCC can create connections only between NYC and SEA and between SEA and DFW. Because all domain controllers are in the same domain and therefore maintain a full copy of the same domain directory partition, replication can occur through the hub site SEA without requiring a connection between DFW and NYC. As a change occurs in NYC, this change (in its entirety) is replicated to SEA. Because domain controllers that are holding full domain directory partition replicas can replicate inbound changes from only another full domain replica, SEA-DC-01 is a valid replication partner for DFW-DC-01, and therefore, in turn, DFW-DC-01 transitively replicates the NYC change from SEA-DC-01. The reverse is also true—changes replicate from DFW to SEA and then to NYC. Thus, changes that originate in SEA replicate to both DFW and NYC directly, but replication must occur twice for changes to flow from DFW to NYC and from NYC to DFW.

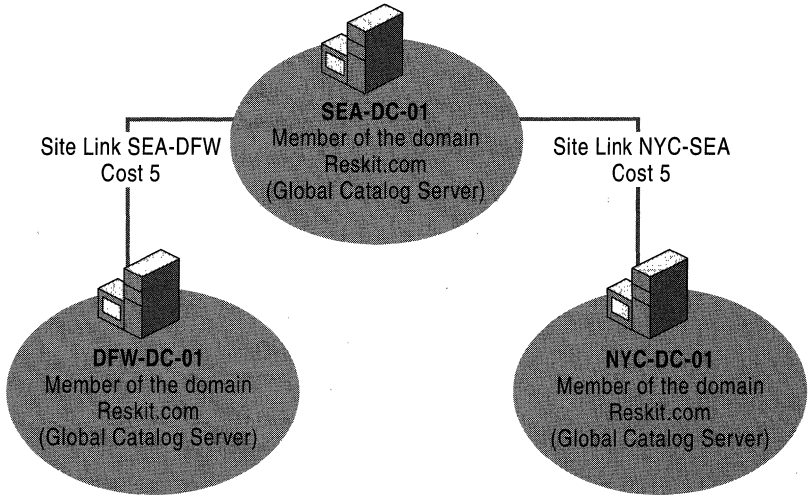


Figure 6.14 A Single Domain in Three Sites Connected by Nontransitive Site Links

Site Links with Two Domains in Three Sites

Figure 6.15 shows that the SEA site contains a domain controller for the noam.reskit.com domain but not for the reskit.com domain. In this case, how would changes replicate from NYC to DFW?

If transitive site links were enabled or if a site link bridge were created manually between NYC-SEA and SEA-DFW, the KCC could create a connection object to replicate data between DFW and NYC. If neither condition were true, DFW-DC-01 would never see changes from NYC, or vice versa, because the only domain controller in the SEA site is a domain controller for a different domain (noam.reskit.com). If SEA-DC-01 does not contain a replica of the reskit.com domain directory partition, it cannot transitively replicate domain data between NYC and DFW. Schema and configuration data can replicate between NYC and SEA and between SEA and DFW.

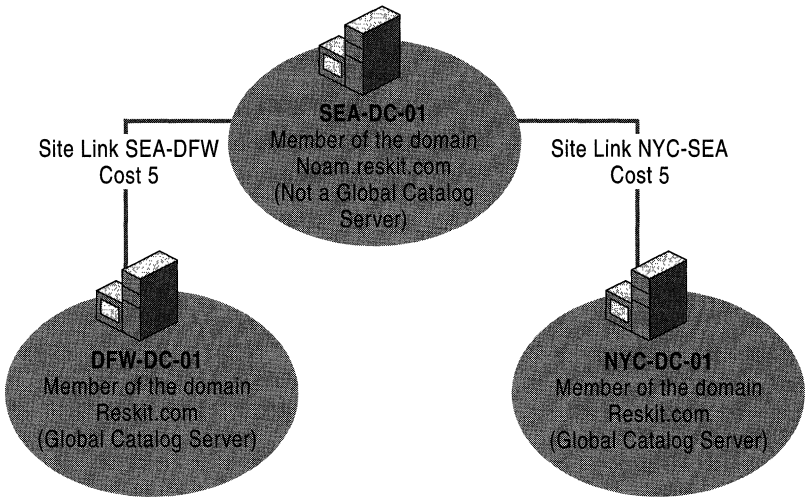


Figure 6.15 Site Links Between Three Sites for Two Domains

In Figure 6.15, if SEA-DC-01 were to become a Global Catalog server, the scenario would not change relative to replication because the domain controller is authoritative for the noam.reskit.com domain and, as such, would hold a full domain directory partition for only that domain. As a Global Catalog server, SEA-DC-01 would hold only a partial replica of the reskit.com domain. In this scenario, SEA-DC-01 would receive changes from NYC, but DFW-DC-01 would not be able to replicate from SEA-DC-01 because SEA-DC-01 contains only a partial replica of the reskit.com partition.

Site Link Bridge with Two Domains in Different Sites

In Figure 6.16, with transitive site links disabled, adding a site link bridge that contains DFW-SEA and SEA-NYC changes the rules on the possible replication partners that the KCC can select. The bridge gives the KCC a route from DFW to NYC in which to create a connection object that allows replication to occur, even if no domain controller for the same domain directory partition exists in SEA. Schema and configuration changes can be replicated over this connection.

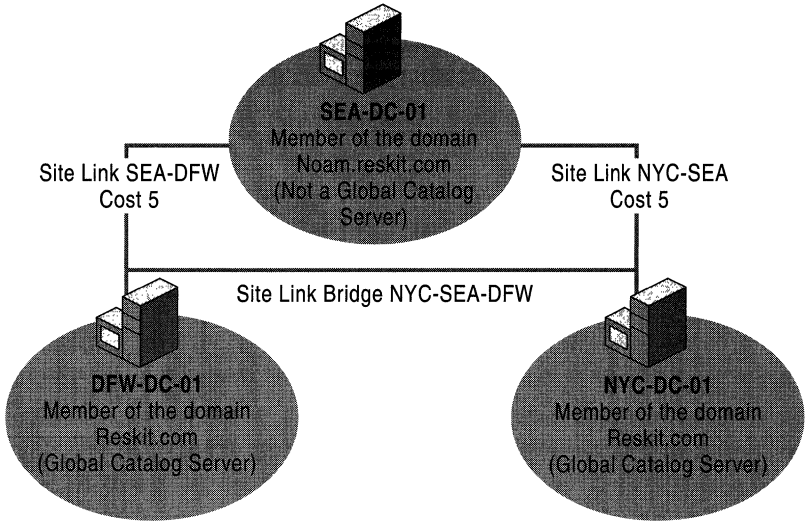


Figure 6.16 Site Link Bridge That Connects SEA-DFW and SEA-NYC

By default, all site links are defined as transitive and do not require definitions of site link bridges. Without a site link bridge, the KCC still constructs connection objects between sites such that replication is the most efficient, based on the defined costs for each site link. However, with a site link bridge, the connection objects might reflect a direct connection from a domain controller in one site to a domain controller in another site where there is no site link.

For information about network routing, see "Unicast IP Routing" in the *Microsoft® Windows® 2000 Server Resource Kit Internetworking Guide* and see "Determining Network Connectivity Strategies" in the *Deployment Planning Guide*.

KCC and Topology Generation

The KCC is a built-in process that runs on all domain controllers. It is a dynamic-link library that modifies data in the local directory in response to systemwide changes, which are made known to the KCC by changes to the data within Active Directory. The KCC generates and maintains the replication topology for replication within sites and between sites.

The KCC has two major functions:

- Configures replication connections (connection objects) between domain controllers. Each connection object defines incoming replication from a replication partner. Within a site, each KCC generates its own connections. For replication between sites, a single KCC per site generates all connections between sites.
- Converts the connection objects that represent inbound replication to the local domain controller into the replication agreements that are actually used by the replication engine.

By default, the KCC reviews and makes modifications to the Active Directory replication topology every 15 minutes to ensure propagation of data, either directly or transitively, by creating and deleting connection objects as needed. The KCC recognizes changes that occur in the environment and ensures that domain controllers are not orphaned in the replication topology.

Tools That Communicate with the KCC

Although the work done by the KCC is evidenced by the automatically generated connection objects that are visible in Active Directory Sites and Services, there is no UI component for managing the KCC per se.

Most replication tasks that affect the KCC can be managed by using Active Directory Sites and Services. For information about non-MMC tools that can be used for advanced replication management and diagnosis, see "Active Directory Diagnostics, Troubleshooting, and Recovery" in this book.

Active Directory Sites and Services

Active Directory Sites and Services is the primary administrative tool that is used to manage replication. Use this tool to create connection objects and site links that the KCC uses to implement replication. Replication within a site is completely automatic and usually requires no intervention. Replication between sites is managed most effectively by changing the settings on the site link objects, as described in this chapter.

For information about procedures for managing replication through Active Directory Sites and Services, see Windows 2000 Server Help.

Event Viewer

Communication from the KCC to the administrator occurs through event logs that you can view in Event Viewer.

The following examples contain a few of the events that are generated by the KCC in the event log:

- Event 1009 (informational): The consistency checker has started updating the replication topology for this server.
- Event 1013 (informational): The replication topology update task terminated normally.
- Event 1265 (warning): The attempt to establish a replication link with parameters *<parameters>* failed with the following status: *<error message>*. The record data is the status code. This operation is going to be re-tried.

The KCC, like all subsystems in Active Directory, has a variable event logging level. By default, only the most important events are logged. You can increase the level of detail in the event log by modifying the value in the **Replication Events** entry in HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS\Diagnostics in the registry. Increasing the level of detail can be used to better understand the behavior of the KCC in different situations. However, a logging level value of greater than 2 generally results in excessive logging that degrades the performance of the component. Increasing the logging level can be useful for troubleshooting problems, but it is not recommended for normal operation. For information about how to modify the registry to increase logging levels, see "Active Directory Diagnostics, Troubleshooting, and Recovery" in this book. For information about using Event Viewer, see Windows 2000 Server Help.

Objects Required by the KCC for Building Topology

When the KCC builds the topology, it must determine which servers that are present in each site in order to construct an efficient topology. The following objects provide the information required by the KCC to create the topology:

- Server object: All domain controllers are identified as server objects in the configuration directory partition, broken down by site.
- The NTDS Settings object: Each server object that represents a domain controller has a child NTDS Settings object, which identifies the domain controller as having Active Directory installed. The NTDS Settings object must be present for the server to be considered as part of the replication topology.

The presence of these objects also determines the site in which the domain controller is to be located. For example, the distinguished name of the NTDS Settings object contains the site to which that domain controller belongs. If the server is physically located in one site but is configured for another site in Active Directory, the KCC uses the information in Active Directory to construct the topology. Therefore, the improper configuration of servers in sites can affect network bandwidth.

Topology Generation Phases

There are two phases of topology generation. During phase one, the KCC evaluates the current topology, determines whether replication failures have occurred with the existing connections, and constructs whatever new connection objects are required to complete the replication topology. During phase two, the KCC implements, or "translates," the decisions that were made in phase one into agreements between the replication partners.

Phase One: Evaluating the Current Topology and Generating Connection Objects

The topology is evaluated by reading the connection objects. When the KCC notices a connection object, it reads the NTDS Settings object of the source domain controller (indicated by the *fromServer* value on the connection object) to determine what directory partitions its destination controller has in common with the source domain controller. The *hasMasterNCs* attribute (where "NC" stands for "naming context," a synonym for "directory partition") of an NTDS Settings object contains the set of writable (non-Global Catalog) directory partitions that are located on that domain controller. The *hasPartialReplicaNCs* attribute contains the set of partial-replica directory partitions (Global Catalog partitions) that are located on that domain controller. For each directory partition that the two domain controllers have in common and that matches the full and partial characteristics of a replication source, the KCC creates (or updates) a replication agreement.

Note Within a site, all KCCs generate connection objects for replication within the site. When there is more than one site, a single KCC in each site generates all connection objects for replication between sites.

Phase Two: Translating Connections

In phase two, all KCCs process their connection objects and translate them into connection agreements between pairs of domain controllers. At specified intervals, Active Directory replicates data from one replication partner to the other for directory partitions that they have in common. These replication agreements do not appear in the administrative tools. They are used internally by the replication engine to track the directory partitions that are to be replicated from specified servers.

For example, suppose that you define a connection object between two domain controllers from different domains. In phase two, assuming that neither of these domain controllers is a Global Catalog server, the KCC identifies the only two directory partitions that the domain controllers have in common—the schema directory partition and the configuration directory partition. If you create a connection object that links domain controllers in the same domain, at least three directory partitions are replicated: the schema directory partition, the configuration directory partition, and the domain directory partition.

In contrast, if the connection object is established between two domain controllers that are Global Catalog servers, a partial replica of each directory partition (which includes only specified attributes) in the forest is replicated between the two domain controllers.

Intervals at Which the KCC Runs

The KCC evaluates the replication topology at specified intervals, which can be modified.

By default, the KCC runs its first replication topology check five minutes after the domain controller starts. This interval can be modified by changing the **Repl topology update delay (secs)** entry in HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS\Parameters as follows:

Value: Number of seconds to wait between the time Active Directory starts and the KCC runs for the first time.

Default: 300 seconds (5 minutes)

Data type: REG_DWORD

By default, as long as services are running, the KCC checks the topology every 15 minutes and makes changes as necessary. The administrator can modify the interval at which the KCC performs this review by changing the **Repl topology update period** entry in HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS\Parameters as follows:

Value: Number of seconds between KCC topology updates

Default: 900 seconds (15 minutes)

Data type: REG_DWORD

Caution Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. There are programs available in Control Panel or Microsoft Management Console (MMC) for performing most administrative tasks. These programs provide safeguards that prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your system. Registry editors bypass the standard safeguards that are provided by these administrative tools. Modifying the registry is recommended only when no administrative tool is available. Before you make changes to the registry, it is recommended that you back up any valuable data on the computer. For instructions about how to edit registry entries, see Help for the registry editor that you are using. For more information about the registry, see the *Microsoft Windows 2000 Resource Kit* Technical Reference to the Windows 2000 Registry (Regentry.chm).

Automated Replication Topology Generation Within a Site

During topology creation within a site, KCC activity includes topology translation and topology generation. The KCC "translates" the information provided by an algorithm and then creates (generates) connection objects that implement the topology that is dictated by the translation. In general, topology translation and generation are accomplished as follows:

- The KCC on each domain controller runs an algorithm to determine the topology for replication within its site. All servers in the same site have the same information about each other, so they all deduce the same topology.
- The KCC creates the connection objects for servers from which the domain controller receives data (source servers), as dictated by the algorithmically generated topology.
- The sum total of the connection objects for all servers is the desired topology.

By listing and sorting the domain controllers that hold replicas of the same partition, the topology is generated and then optimized to minimize the number of hops.

Forced Topology Generation

Topology generation creates the topology for replication within a site automatically. Topology generation occurs on a schedule that determines how often the KCC runs. Topology generation can also be started manually by right-clicking the NTDS Settings object, clicking **All Tasks**, and then clicking **Check Replication Topology**.

Simplified Ring Topology Generation

An overly simplified process for creating the topology for replication within a site begins as follows:

- The KCC generates a list of all servers in the site that hold that directory partition.
- These servers are connected in a ring.
- For each neighboring server in the ring from which the current domain controller is to replicate, the KCC creates a connection object if one does not already exist.

This simple approach guarantees a topology that tolerates a single point of failure. If a domain controller is not available, it is not included in the ring that is generated by the list of servers because its NTDS Settings object is not available. However, this topology, with no other adjustments, accommodates only seven servers. Beyond this number, the ring would require more than three hops for some servers.

The simplest case scenario—seven or fewer domain controllers, all in the same domain and site—would result in the topology shown in Figure 6.17. Even if one or all of these domain controllers were Global Catalog servers, when the KCC runs on those particular computers, no extra connections would be necessary. The only directory partitions to replicate are a single domain directory partition, the schema directory partition, and the configuration directory partition. Those topologies are generated first, and at that point, sufficient connections to replicate each directory partition have already been created.

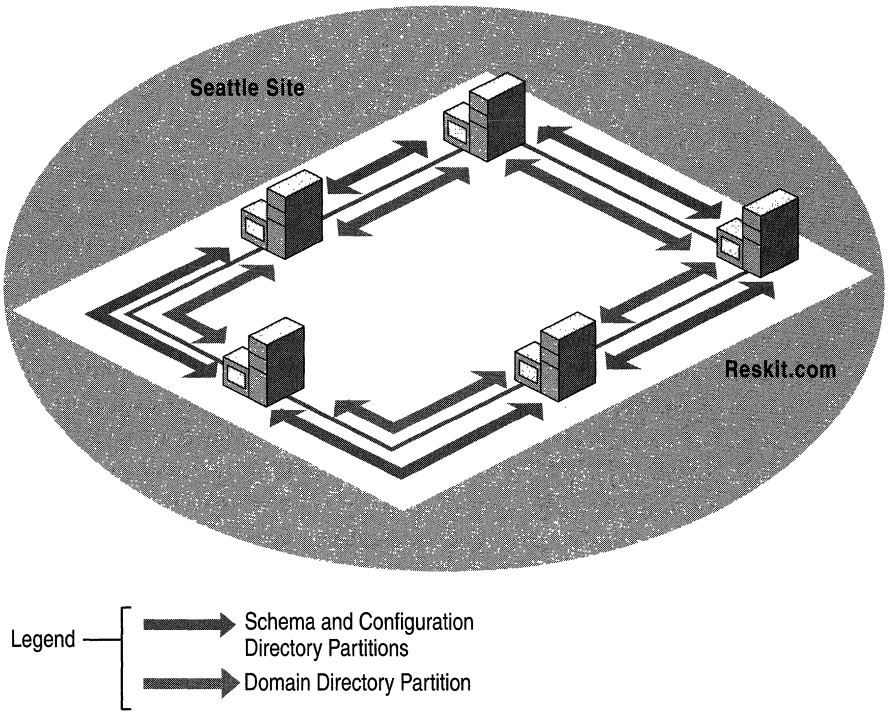


Figure 6.17 Simple Ring Topology That Requires No Optimization

Because a ring topology is created for each directory partition, the topology might look different if domain controllers from a second domain were present in the site. Figure 6.18 illustrates the topology for domain controllers from two domains in the same site with no Global Catalog servers defined in the site.

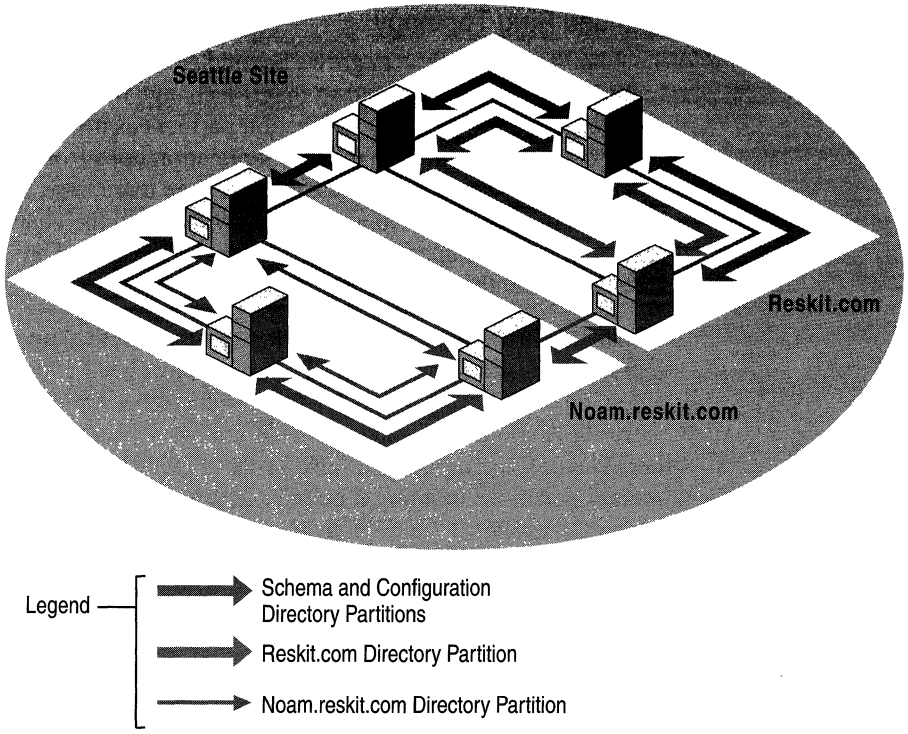


Figure 6.18 Ring Topology for Two Domains in a Site That Has No Global Catalog Server

Note The examples in Figure 6.18 and Figure 6.19 are designed to illustrate only how the KCC creates connections, not how to configure a site. A Global Catalog server is a requirement for logging on to a domain, and for this reason, it is advisable to have at least one Global Catalog server in a site. If a Global Catalog server is not available in a site and there is a Global Catalog server in a remote site, the server in the remote site can be used for the logon process. If no Global Catalog is available in any site, the logon process proceeds with cached logon information. (For more information about Global Catalog support for logging on to domains, see "Active Directory Logical Structure" in this book.)

Expanded Ring Topology

When the number of servers grows beyond seven, the KCC estimates the number of connections that are needed so that if a change occurs at any one domain controller, there are as many replication partners as needed to ensure that no domain controller is more than three replication hops from another domain controller (that is, a change takes no more than three hops before it reaches another domain controller that has already received the change by another path). These optimizing connections are created at random and are not necessarily created on every third domain controller.

In Figure 6.19, there is no Global Catalog server in the site, all domain controllers are in the same domain, but enough servers have been added to require optimizing connections. Although they are located in the same domain and site, Domain Controller A and Domain Controller B are more than three hops away from each other. The optimizing connections for the domain, schema, and configuration directory partitions that might be created from Domain Controller A to Domain Controller B are depicted as a single straight line in the diagram for readability, but in reality, these partitions are replicated separately as shown between the neighboring replication partners. There would also be more optimizing connections than the one shown.

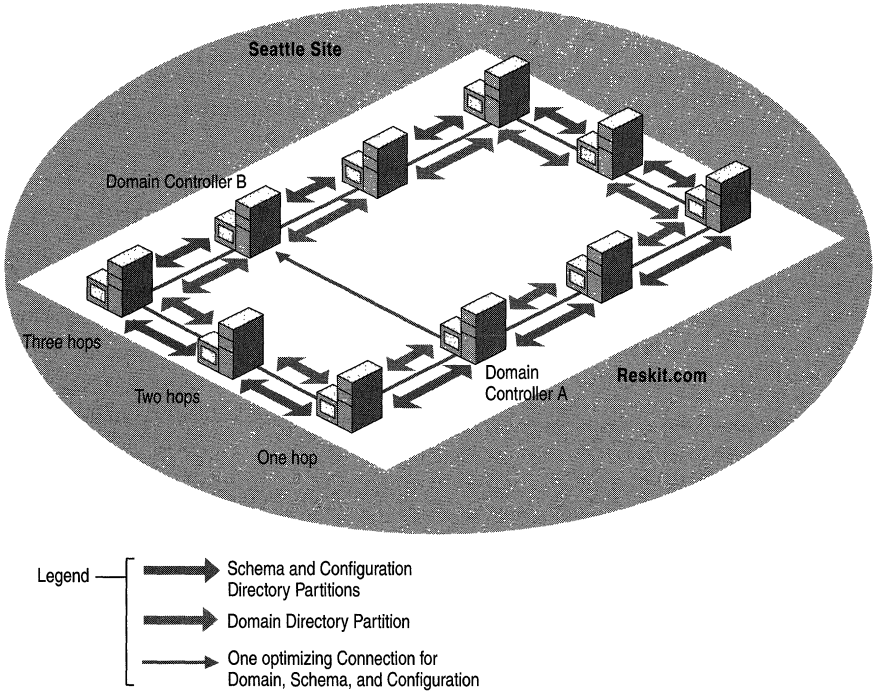


Figure 6.19 Optimized Connections for Ten Domain Controllers in the Same Domain in a Single Site

Figure 6.20 illustrates the connections required between a Global Catalog domain controller and three other domains to which it does not belong. When a Global Catalog server is added to the site, additional connections are required to replicate copies of the directory partitions to the domains to which the Global Catalog server does not belong. Because the reskit.com domain has only seven servers, no optimizing connections are required in the replication topology for the reskit.com directory partition. However, the Global Catalog server is the source for each domain directory partition in the forest, and in this example, the KCC has created connection objects to replicate from domain controllers for each of those domain directory partitions within the site. Wherever a domain directory partition is replicated, the KCC also uses the connection to replicate the schema and configuration directory partitions.

Note Connection objects are generated independently for the configuration and schema directory partitions (one connection) and for the separate domain directory partitions, unless a connection from the same source to destination domain controllers already exists for one directory partition, in which case the same connection is used for both (a duplicate connection is not created).

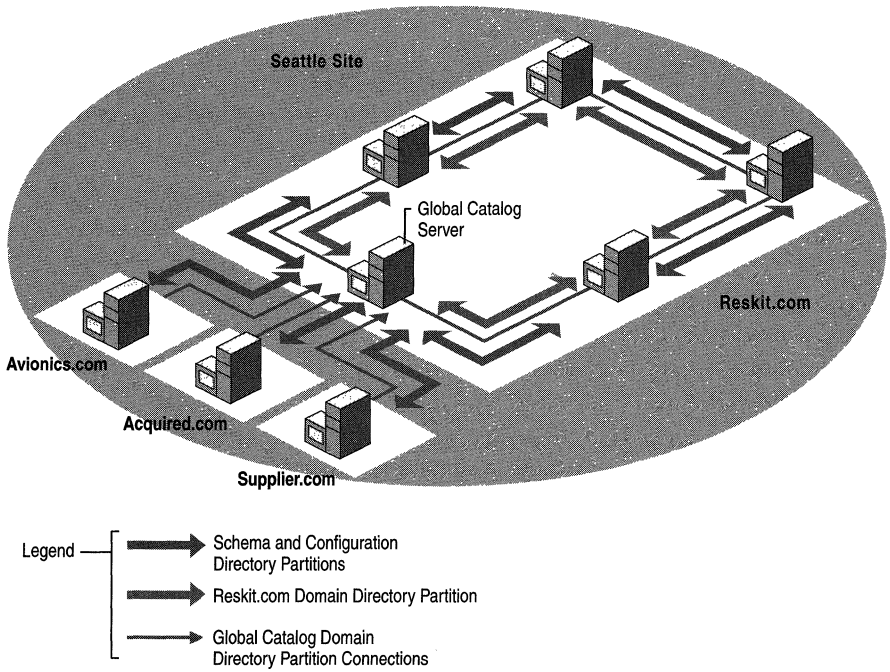


Figure 6.20 Optimized Connections That Are Required for Site That Has Four Domains and a Global Catalog Server

Optimized Ring Topology Connections Within a Site

Connections are added to optimize a ring topology within a site on the basis of the answer to the following question:

> Given a set of nodes in a ring, what is the minimum number of connections, *n*, that each server must have to ensure a path of no more than three hops to another server?

Given *n*, topology generation proceeds as follows.

If the local server does not have *n* extra connections, do the following:

- Choose *n* other servers randomly in the site as source servers.
- For each of those servers, create a connection object.

This approach approximates the minimum-hop goal of three servers. In addition, it grows well, because as the site grows in server count, old optimizing connections are still useful and are not removed. Also, every time an additional 9 to 11 servers are added, a connection object is deleted at random; then a new one is created, ideally having one of the new servers as its source. This process ensures that, over time, the extra edges are distributed well over the entire site.

Excluded Nonresponding Servers

The KCC automatically rebuilds the replication topology when it recognizes that a domain controller has failed or is unresponsive.

The criteria that the KCC uses to determine when a domain controller is not responsive depend upon whether the server computer is within the site or not. Two thresholds must be reached before a domain controller is declared "unavailable" by the KCC:

- The requesting domain controller must have made *n* number of attempts to replicate from the target domain controller.
  - For replication between sites, the default value is 1 attempt.
  - For replication within a site, the following distinctions are made between the two immediate neighbors (in the ring) and the optimizing connections:

    For immediate neighbors, the default value is 0 failed attempts. (Thus, as soon as an attempt fails, a new server is tried.)

    For optimizing connections, the default value is 1 failed attempt. (Thus, as soon as a second failed attempt occurs, a new server is tried.)

- A certain amount of time must pass since the last successful replication attempt.
  - For replication between sites, the default time is 2 hours.
  - For replication within a site, a distinction is made between the two immediate neighbors (in the ring) and the optimizing connections:

    For immediate neighbors, the default time is 2 hours.

    For optimizing connections, the default value is 12 hours.

To modify the thresholds for excluding nonresponding servers, use the following registry entries in HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet \Services\NTDS\Parameters, with the data type REG_DWORD. You can modify these values to any desired value as follows:

For replication between sites, use the following entries:

- **IntersiteFailuresAllowed**

  Value: Number of failed attempts

  Default: 1
- **MaxFailureTimeForIntersiteLink (secs)**

  Value: Time that must elapse before being considered stale, in seconds

  Default: 7200 (2 hours)

For optimizing connections within a site, use the following entries:

- **NonCriticalLinkFailuresAllowed**

  Value: Number of failed attempts

  Default: 1
- **MaxFailureTimeForNonCriticalLink**

  Value: Time that must elapse before considered stale, in seconds

  Default: 43200 (12 hours)

For immediate neighbor connections within a site, use the following entries:

- **CriticalLinkFailuresAllowed**

  Value: Number of failed attempts

  Default: 0
- **MaxFailureTimeForCriticalLink**

  Value: Time that must elapse before considered stale, in seconds

  Default: 7200 (2 hours)

Caution Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. There are programs available in Control Panel or Microsoft Management Console (MMC) for performing most administrative tasks. These programs provide safeguards that prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your system. Registry editors bypass the standard safeguards that are provided by these administrative tools. Modifying the registry is recommended only when no administrative tool is available. Before you make changes to the registry, it is recommended that you back up any valuable data on the computer. For instructions about how to edit registry entries, see Help for the registry editor that you are using. For more information about the registry, see the *Microsoft Windows 2000 Resource Kit* Technical Reference to the Windows 2000 Registry (Regentry.chm).

When the original domain controller begins responding again, the KCC automatically restores the replication topology to its pre-failure condition the next time that the KCC runs.

Fully Optimized Ring Topology Generation

Taking the addition of extra connections, management of nonresponding servers, and growth-management mechanisms into account, the fully optimized intrasite topology generation proceeds and the appropriate connection objects are created and deleted according to the available criteria.

Note Connection objects from nonresponding servers are not deleted because the condition is expected to be transient.

The following key points characterize the optimized approach to topology generation:

- Sorting the servers ensures that each server arrives at the same desired topology.
- Intrasite topology generation never creates or deletes connection objects that are not directly under the NTDS Settings object for the local domain controller.

Automated Intersite Topology Generation

Topology generation for replication between sites is more complex than for replication within a site because Windows 2000 supports replication between sites over asynchronous transport (SMTP). Therefore, whereas for intrasite topology generation the KCC can assume that any server can replicate to any other server, the same assumption cannot be made for replication between sites.

Intersite Topology Generator Role

A fundamental concept in the generation of the topology within a site is that each server does its part to create a sitewide topology. In a similar manner, the generation of the topology between sites depends on each site doing its part to create a forest-wide topology between sites. As part of this effort, one domain controller per site assumes the role of the intersite topology generator. The KCC on this domain controller is responsible for creating the connections between the domain controllers in its site and the domain controllers in other sites, which includes specifically the inbound replication connection objects for all bridgehead servers in the site in which the domain controller is located.

After the intersite topology generator assesses the topology and determines that its own site is the only site, it performs no further processing because no connections between sites are possible for the current configuration.

Generation of Connections Between Sites vs. Generation of Connections Within a Site

Connection objects between bridgehead servers for replication between sites and connection objects for connections within a site are created differently.

When the KCC on each domain controller generates the intrasite topology for the site in which it resides, the KCC creates a connection object in Active Directory only when a connection object is required for the local computer. This change propagates to other domain controllers through the normal replication process. Each domain controller uses the same algorithm to compute the replication topology; in a state of equilibrium between domain controllers, each domain controller arrives at the same result with respect to what the replication topology should be. In the process, each domain controller creates its own connection objects.

For connections between sites, in each site, the KCC that has the intersite topology generator role (regardless of the domain) is responsible for reviewing the intersite topology and creating inbound replication connection objects as necessary for bridgehead servers in the site in which it resides.

When the intersite topology generator determines that a connection object needs to be modified on a specific bridgehead server in the site, the intersite topology generator makes the change to its local Active Directory copy. These changes propagate to the bridgehead servers in the site as part of normal replication within the site. When the KCC on the bridgehead server reviews the topology after receiving these changes, it translates the connection objects into replication agreements (replication partners) that Active Directory uses to replicate data from remote bridgehead servers.

Bridgehead Server Selection

As the KCC constructs the intersite topology for each directory partition, the servers in each site are evaluated for becoming bridgehead servers. If preferred bridgehead servers are configured, these servers are the candidates for selection. Otherwise, all domain controllers in the site that host the directory partition and can communicate over a specific transport are candidates for becoming a bridgehead server. In either case, the first domain controller that meets the requirements becomes the bridgehead server. All preferred bridgehead servers in the same site that are configured for the same transport are considered to be equal. In a state of equilibrium of the configuration directory partition, the intersite topology generator from each site selects the same bridgehead server, given the same requirements. For example, if the same bridgehead server (either preferred or automatically selected) is capable of replication over the IP transport and holds the requested directory partition, that server is selected by bridgehead servers from other sites that require the same directory partition data.

If preferred bridgehead servers are defined for a specific site and all of the servers specified are unavailable, no failover is performed to the other domain controllers in the site, even if they can act as bridgehead servers. The same is true in the case where no preferred bridgeheads are specified and where all domain controllers that can act as a bridgehead for the required data are unavailable.

Intersite Topology Generator Role Owner

The current owner of the intersite topology generator role is communicated through the normal Active Directory replication process. Initially, the first domain controller in the site becomes the intersite topology generator for the site. The role does not change as additional domain controllers are added to the site until the current intersite topology generator becomes unavailable.

▶ **To determine the intersite topology generator role owner for a site**

1. In Active Directory Sites and Services, click the site object.
2. In the details pane, right-click the **NTDS Site Settings** object, and then click **Properties**. The current role owner appears in the **Server** box under **Inter-Site Topology Generator**.

Note You cannot change the intersite topology generator role.

Intersite Topology Generator Role Owner Notification

At 30-minute intervals, the current intersite topology generator notifies every other domain controller in the site of its existence by writing the attribute *interSiteTopologyGenerator* on the NTDS Settings object under its domain controller object in the configuration directory partition.

As the *interSiteTopologyGenerator* attribute gets propagated to other domain controllers by Active Directory replication, the KCC on each of these computers monitors this attribute to verify that it has been written. If a period of 60 minutes elapses without a modification, a new intersite topology generator takes over.

Establishing a New Intersite Topology Generator

When a new intersite topology generator is required, each domain controller requests the list of servers in the site in ascending order. The domain controller that takes over the role is the one that is next in the list of servers after the current owner. This domain controller then writes the *interSiteTopologyGenerator* attribute and performs the necessary KCC processes to manage inbound connection objects for bridgehead servers.

When there are two domain controllers in the site that appear to own the intersite topology generator role, there might be a temporary state of inbound replication connection objects being created by both computers. However, after replication occurs and all domain controllers receive the change that identifies the new intersite topology generator, the KCC on the intersite topology generator adjusts the topology.

Intersite Topology Generator and Modified Connections

It is possible for a connection object to be modified by an administrator on one domain controller and be modified subsequently by the KCC on another domain controller prior to the initial change being replicated. Overwriting such a change can occur within the local site or when a connection object changes in a remote site. By default, the KCC runs every 15 minutes. If the connection object change is not replicated to another domain controller before the KCC on that domain controller runs, the KCC on that domain controller might modify the same connection object. In this case, ownership of the connection object belongs to the KCC because the latest write to the connection object is the write that is applied.

To ensure that modification of an intersite connection object is not overwritten by the KCC, make the modification on the computer that has the intersite topology generator role in the site of the modified connection object.

Security Between Replication Partners

Security of replication is important to ensure that an unauthorized program cannot act as a replication destination because replication destinations have access to secrets. Security must also prevent an unauthorized program from acting as a replication source because a rogue replication source can pretend to originate updates and thereby make unauthorized directory changes. Therefore, mutual authentication of replication partners and access control at the replication source are both required. The security technology that is used in replication depends on the replication transport.

RPC Transport Security

For authentication, mutual authentication is performed either by the Kerberos v5 authentication protocol or by NTLM when authenticating to a Microsoft® Windows® NT version 4.0–based backup domain controller.

For authorization, the DS-Replication-Get-Changes control access right provides access control at the replication source. This control access right is granted in access control entries (ACEs) in the security descriptor on the topmost object in the directory partition. By default, this right is granted to Enterprise Domain Controllers and Domain Admins and is required in order to run the Active Directory Installation Wizard (Dcpromo.exe).

For information about Active Directory authentication, see "Authentication" in this book. For information about Active Directory authorization, see "Access Control" in this book. For information about the Active Directory Installation wizard, see "Active Directory Data Storage" in this book.

ISM Transport Security

The authentication and encryption mechanism that ISM uses is similar to that of Secure/Multipurpose Internet Mail Extensions (S/MIME), which is a standard that enables binary data to be published and read on the Internet. By using S/MIME, the sender always includes its own certificate in addition to the certificate-authority certificate chain that extends to the root certifying authority. Requests to the replication source are signed but not sealed. The request does not contain any secret data, so the domain controller does not require knowledge of the certificates for any other domain controllers. Responses to the replication destination are signed and sealed by using the local domain controller certificate plus the certificate that the requester included in its request message.

Authentication is possible because all certificates used for Active Directory replication (that is, "DomainController" certificates) contain an attribute (*altSubjectName*) that identifies the *objectGuid* attribute value of the owner's computer object. The mapping of the computer *objectGuid* to the server object can be done by any domain controller because the required information is stored in the configuration directory partition.

For authorization, the receiving domain controller verifies the validity of the certificate (including verifying the fact that it was issued by a trusted certificate authority, and so on), and then the receiving domain controller extracts the *objectGuid* value of the sender's computer object. It then attempts to map that GUID to a server object in its forest. If mapping succeeds, the receiving domain controller additionally verifies that the sending server is an Active Directory domain controller (on the basis of the sending server having a child NTDS Settings object of the proper object class, which can be created only by the directory service itself). Blanket replication access is thereby granted to all Active Directory domain controllers in the forest—and only to Active Directory domain controllers.

For information about certificate security, see "Windows 2000 Certificate Services and Public Key Infrastructure" in this book. For information about Active Directory authentication, see "Authentication" in this book. For information about Active Directory authorization, see "Access Control" in this book.

Advanced Replication Management

Active Directory replication occurs automatically and reliably with no administrative intervention, other than that required to configure sites and site links. Some replication events, however, warrant additional understanding for those administrators who need to fine-tune replication beyond the default behavior. For more information about advanced replication management and troubleshooting, see "Active Directory Diagnostics, Troubleshooting, and Recovery" in this book.

Reciprocal Replication

Replication between sites exhibits request-pull behavior (the receiver, or destination, requests changes from the sender, or source, according to a schedule). Replication within a site, on the other hand, exhibits notify-pull behavior (the receiver is notified of changes by the sender, and the receiver then requests changes from the sender). For replication to occur in two directions, both sides of a connection must be able to initiate replication with the other side.

In cases where replication between sites can be initiated by only one side of a site link, such as when a dial-up connection must go through an Internet service provider (ISP), a flag can be set on the connection object (on the basis of site link attribute information) to implement two-way replication between the source and destination domain controllers of the connection as follows: The domain controller on the dial-up side of the link opens a connection and initiates replication (requests changes). After it receives the changes from the domain controller it contacted, it responds by sending a change notification. The change notification prompts the second domain controller to request changes from the first domain controller. The effect is two-way replication over the initial connection that was opened by the dial-up side of the site link.

The most common scenario in which reciprocal replication is enabled is replication between the main office and a branch office of a company. In this scenario, the branch office must dial up an ISP before a tunneled IP connection (also called a virtual private network or VPN link) can be established. In this example, the following occurs:

1. The branch office dials up the ISP.
2. The branch office establishes a tunneled IP connection to the main office.
3. The branch office initiates replication by requesting changes from the main office.
4. After replication, the branch office immediately sends a change notification to the main office.
5. The main office requests replication changes from the branch office.
6. The branch office replicates its changes to the main office.

In this scenario, reciprocal replication is required because the main office cannot instruct the ISP to first dial up the branch office and, therefore, cannot initiate a connection. Only the branch office can initiate communication to the main office. With reciprocal replication enabled, after the VPN link has been established, replication from the branch office can be initiated by the branch office.

Enabling reciprocal replication between two sites involves modifying the *options* attribute value on the site link object. With this attribute set on the site link, the KCC creates the connections across the link with the appropriate setting that is in effect. Use ADSI Edit to enable reciprocal replication.

▶ **To enable reciprocal replication between two sites**

1. In ADSI Edit, expand the Configuration container.
2. Navigate to the **Inter-SiteTransports** container, and select **CN=IP**. (You cannot enable reciprocal replication for SMTP links.)
3. Right-click the site link object for the sites for which you want to enable reciprocal replication, and then click **Properties**.
4. In the **Select a property to view** box, select **options**.
5. In the **Edit Attribute** box, if the **Value(s)** box shows **<not set>**, type **2** in the **Edit Attribute** box.

   If the **Value(s)** box already contains a value, you must derive the new value by using a Boolean BITWISE-OR calculation on the old value, as follows: *old_value* BITWISE-OR 2. For example, if the value in the **Value(s)** box is 1, calculate 0001 OR 0010 to equal 0011. Type the integer value of the result in the **Edit Attribute** box; for this example, the value is 3.
6. Click **OK**.

Change Notification

Change notification is a mechanism by which a domain controller notifies a replication partner that it has changes. Replication within a site occurs as a response to changes; as changes occur on one domain controller, it notifies its replication partner, which prompts the partner to request the changes. When a domain controller performs an update to an attribute, it sends notification to its replication partner within a specified time following the change.

Change Notification Within a Site

For changes that occur within a site, there is a "holdback timer" that determines the interval between the time a change is made and the time that the source server notifies its replication partners. This interval serves to stagger network traffic caused by replication. When a domain controller makes a change (originating or replicated) to a directory partition, it starts the timer; when the timer expires, the domain controller notifies all of its replication partners (for that directory partition and within the site) that it has changes. If a partner is not engaged in requesting changes from another partner, it sends its change request to the notifying server.

The default value for the holdback timer is 300 seconds, or 5 minutes. To change the default registry setting, you can set a new value in the **Replicator notify pause after modify (secs)** entry in HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS\Parameters.

Note Very small values for this timer generate redundant notifications, which can decrease performance.

A domain controller does not notify all of its replication partners at one time. By delaying between notifications, the domain controller spreads out the load of responding to replication requests from its partners. The default delay between notifications is 30 seconds. To change the default delay, set a new value in the **Replicator notify pause between DSAs (secs)** entry in HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS\Parameters.

Caution Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. There are programs available in Control Panel or Microsoft Management Console (MMC) for performing most administrative tasks. These programs provide safeguards that prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your system. Registry editors bypass the standard safeguards that are provided by these administrative tools. Modifying the registry is recommended only when no administrative tool is available. Before you make changes to the registry, it is recommended that you back up any valuable data on the computer. For instructions about how to edit registry entries, see Help for the registry editor that you are using. For more information about the registry, see the *Microsoft Windows 2000 Resource Kit* Technical Reference to the Windows 2000 Registry (Regentry.chm).

Change Notification Between Sites

By default, changes are replicated between sites according to a schedule and not according to when changes occur. For this reason, the greatest replication latency across the forest is the sum of the greatest replication latencies along the single longest replication path of any directory partition.

For special circumstances, you can configure change notifications on connections between sites. By modifying the site link object, you can enable change notification between sites for all connections that occur over that link. Use ADSI Edit to enable change notification between sites.

▶ **To enable change notification between sites**

1. In ADSI Edit, expand the Configuration container.
2. Navigate to the **Inter-Site Transports** container, and select **CN=IP**. (You cannot enable change notification for SMTP links.)
3. Right-click the site link object for the sites for which you want to enable change notification, and then click **Properties**.
4. In the **Select a property to view** box, select **options**.
5. In the **Edit Attribute** box, if the **Value(s)** box shows **<not set>**, type **1** in the **Edit Attribute** box. If the **Value(s)** box contains a value, you must derive the new value by using a Boolean BITWISE-OR calculation on the old value, as follows: *old_value* BITWISE-OR 1. For example, if the value in the **Value(s)** box is 2, calculate 0010 OR 0001 to equal 0011. Type the integer value of the result in the **Edit Attribute** box; for this example, the value is 3.
6. Click **OK**.

Enabling change notifications across site links propagates all change notifications. With change notification between sites set, changes propagate to the remote site with the same frequency that they are propagated within the source site, including changes that warrant urgent replication.

Note Do not enable change notification on demand-dial IP site links or on SMTP site links.

Urgent Replication

Urgent replication is implemented by immediately notifying replication partners over RPC/IP that changes have occurred on a source domain controller. Urgent replication uses regular change notification between destination and source domain controller pairs that otherwise use change notification, but notification is sent immediately in response to urgent events instead of waiting the default period of five minutes. Therefore, if you have change notification enabled on a site link, urgent replication is possible between sites for events that trigger it.

Events That Trigger Urgent Replication

Urgent Active Directory replication is always triggered by certain events on all domain controllers within the same site. When you have enabled change notification between sites, these triggering events also replicate immediately between sites.

Immediate replication between Windows 2000–based domain controllers in the same site is prompted by the following:

- Assigning an account lockout, which prohibits a user from logging on after a certain number of failed attempts.
- Changing a Local Security Authority (LSA) secret, which is a secure form in which private data is stored by the LSA.
- Change in the relative identifier (known as a "RID") master role owner, which is the single domain controller in a domain that assigns relative identifiers to all domain controllers in that domain.

Urgent Replication of Account Lockout Changes

Account lockout is a security feature that sets a limit on the number of failed authentication attempts that are allowed before the account is "locked out" from a further attempt to log on, in addition to a time limit for how long the lockout is in effect.

In Windows 2000, account lockout is urgently replicated to the primary domain controller (PDC) emulator role owner and is then urgently replicated to the following:

- Domain controllers in the same domain that are located in the same site as the PDC emulator.
- Domain controllers in the same domain that are located in the same site as the domain controller that handled the account lockout.
- Domain controllers in the same domain that are located in sites that have been configured to allow change notification between sites (and, therefore, urgent replication) with the site that contains the PDC emulator or with the site where the account lockout was handled. These sites include any site that is included in the same site link as the site that contains the PDC emulator or in the same site link as the site that contains the domain controller that handled the account lockout.

In addition, when authentication fails at a domain controller other than the PDC emulator, the authentication is retried at the PDC emulator. For this reason, the PDC emulator locks the account before the domain controller that handled the failed-password attempt if the bad-password-attempt threshold is reached. For more information about how the PDC emulator role owner manages password changes and account lockouts, see "Managing Flexible Single-Master Operations" in this book.

Managing Urgent Replication

The following guidelines can be useful when deciding whether to enable change notification between sites relative to achieving urgent replication.

- If you want urgent replication everywhere, put all domain controllers for the specific domain in a single site (this option might not be realistic).
- If you want urgent replication everywhere but still want the benefits of site affinity, use multiple sites and enable change notification on all site links.
- By default, a user lockout prompts urgent replication at the site that contains the domain controller that handled the authentication and the site that contains the PDC emulator role owner.

Forced Replication Between Two Servers

You can use a connection object to force replication from the inbound server. In Active Directory Sites and Services, right-click a connection object, and then click **Replicate Now**. For information about how to force replication between two servers by using a connection object, see Windows 2000 Server Help. For more information about forcing replication by using other tools, see "Active Directory Diagnostics, Troubleshooting, and Recovery" in this book.

Replication of Password Changes

Password changes are replicated differently than normal (non-urgent) replication and urgent replication. Changes to security account passwords present a replication latency problem wherein a user's password is changed on domain controller A and the user subsequently attempts to log on, being authenticated by domain controller B. If the password has not replicated from A to B, the attempt to log on fails. Active Directory replication remedies this situation by forwarding password changes immediately to a single domain controller in the forest, the PDC emulator.

In Windows 2000 domains, a single domain controller per forest holds the role of PDC emulator, which simulates the behavior of a Microsoft Windows NT version 3.*x*–based or Windows NT 4.0–based primary domain controller. In Windows NT 4.0, the only domain controller that can accept updates is the primary domain controller. If authentication fails at a backup domain controller, the authentication request is passed immediately to the primary domain controller, which is guaranteed to have the current password.

In Windows 2000, when a user password is changed at a specific domain controller, that domain controller attempts to update the respective replica at the domain controller that holds the PDC emulator role. Update of the PDC emulator occurs immediately, without respect to schedules between sites on site links. The updated password is propagated to other domain controllers by normal replication within a site. When the user logs on to a domain and is authenticated by a domain controller that does not have the updated password, the domain controller refers to the PDC emulator to check the credentials of the user name and password rather than denying authentication based on a nonvalid password. Therefore, the user can log on successfully even when the authenticating domain controller has not yet received the updated password.

If the update at the PDC emulator fails for any reason, the password change is replicated non-urgently by normal replication.

For clients that are running Windows NT 4.0 or clients that are running Microsoft® Windows® 95 or Microsoft® Windows® 98 without the Directory Services Client Update Pack, the client attempts to contact the PDC emulator. If the client is directory-aware, the client contacts any domain controller and the contacted domain controller then attempts to contact the PDC emulator.

Note A domain controller can be set to not contact the PDC emulator if the PDC emulator role owner is not in the current site. If the **AvoidPdcOnWan** entry in HKEY_LOCAL_MACHINE\CurrentControlSet\Services\Netlogon\Parameters\ is set to 1, the password change reaches the PDC emulator non-urgently through normal replication.

Creation of Extra Connection Objects

The KCC is designed to produce a topology that provides low replication latency, that adapts to failures, and that does not need modification. Adding connections is not recommended because extra connections gradually reduce the ability of the KCC to automatically choose the best configurations. In addition, you create a situation where you must continually evaluate whether the manual connections are doing the best possible job of replicating changes.

Adding extra connections does not necessarily reduce replication latency. Within a site, latency issues are usually related to factors other than the KCC's choices for topology. Factors that affect latency include the following:

- Interruption of the service of key domain controllers, such as the PDC emulator, Global Catalog servers, or bridgehead servers.
- Domain controllers that are too busy replicate in a timely manner (too few domain controllers).
- Network connectivity issues.
- DNS server problems.
- Inordinate amounts of directory updates.

For problems such as these, creating a manual connection does not improve replication latency. Adjusting the scheduling and costs that are assigned to the site link is the best way to influence intersite topology.

The KCC automatically creates connections to keep your directory connected, even if extended failures and outages occur. Create connections manually only if the connections that are automatically configured by the KCC do not connect specific domain controllers that you want to connect. For example, if you want to create a connection on a short-term basis in order to force replication from one particular server, you can use Active Directory Sites and Services to create a connection object if one does not already exist for that server.

Note The cost of the extra connections, as compared to the cost associated with the default configuration, is extra CPU cycles, disk reads, and network messages expended on replication.

You might need extra connections between domain controllers that are within a site or between sites in the following situations:

- When you want to reduce the number of hops between domain controllers within a site. By default, an update takes at most three hops from where it originates to another domain controller in the site. To reduce the number of hops to two hops or to one hop, add extra connections.
- When failures occur between domain controllers in different sites. If failures occur, you might want to add connections that bypass the failed server or servers. (The KCC creates new connections in such cases, but this adaptive process adds some replication latency.)

If you create a connection that is similar to one that the KCC would have created (that is, the KCC would have connected the same two domain controllers in the same direction), the KCC does not create another connection between those servers, nor does it delete or modify the connection that you have created. (For information about how to create a connection object, see Windows 2000 Server Help.)

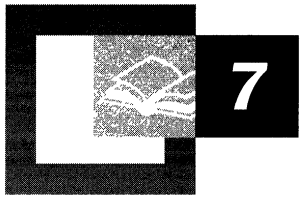
In large networks, use the following general guidelines to manage connections for optimum KCC performance:

- Reduce the number of sites when possible.
- Increase the memory in your domain controllers.
- Turn off the **Bridge all sites** option, if possible.
- As a last resort, turn off automatic generation of intersite topology and create connections manually.

For more information about managing replication by using extra connections and about KCC scaling recommendations, see the Microsoft Knowledge Base link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. Search the Knowledge Base by using the keyword "Q244368."

CHAPTER 7

Managing Flexible Single-Master Operations



Flexible Single-Master Operations (FSMOs) in Active Directory™ are used to prevent conflicting updates in cases where conflict resolution is inappropriate. By planning the management of FSMOs, you enhance the availability of your Active Directory deployment.

In This Chapter

Introduction to Flexible Single-Master Operations

In Active Directory, any domain controller to which you can connect can be updated, even if that domain controller becomes disconnected from the network. This is known as *multimaster update*. When network connectivity is restored, updates are replicated throughout the forest. So if two people simultaneously make conflicting updates, each set of updates is replicated. However, even in the presence of conflicting updates, all domain controllers eventually converge to the same value through a process called *conflict resolution*.

Even with this conflict resolution process, it is sometimes better to prevent conflicts than to resolve them after the fact. For example, if different domain controllers have conflicting versions of the directory schema, the situation could be resolved using the normal conflict resolution methods used by Active Directory. In common cases, the last domain controller to write an update wins. But, because the schema is updated infrequently, and the consistency of the schema is very important, conflict prevention is better than relying on normal conflict resolution methods.

Active Directory performs schema updates in a single master fashion to prevent conflicts. Only one domain controller in the entire forest, the domain controller holding the schema master role, accepts updates to schema objects. An administrator can shift the schema master role from one domain controller to another as the need arises, but at any moment only one domain controller holds the schema master role.

The schema master role is one example of a flexible single-master operation role, also called an operations master role or an FSMO role. Other operations master roles are a part of Microsoft® Windows® 2000 Server; each role controls another specific set of directory changes. For each role, only the domain controller holding that role can make the associated directory changes.

In small Active Directory deployments with a single domain controller, operations master roles are not a consideration. But if you are responsible for the operational health of an Active Directory deployment with more than one domain controller, you need to understand the following:

- Which domain controllers need to hold operations master roles?
- What functionality is lost when a domain controller holding an operations master role is unavailable?
- When a domain controller holding an operations master role is unavailable for an extended period, how do you respond to restore service?

This section discusses each operations master role, answers the questions listed above, and explains how to control access to role placements.

Operations Master Roles

Active Directory defines five operations master roles: schema master, domain naming master, relative identifier (RID) master, primary domain controller emulator, and infrastructure master. The schema master and domain naming master are per-forest roles, meaning that there is only *one* schema master and *one* domain naming master 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. So, in a forest with only one domain there are five operations master roles. In a forest with more than one domain there are more than five roles because the per-domain roles need to exist in each domain.

For example, the Reskit.com forest has three domains:

- Reskit.com. The root domain.
- Na.reskit.com. The North American domain.
- Eur.reskit.com. The European domain.

Therefore, the total number of operations master roles for the Reskit.com forest is eleven:

- Schema master (forest): Reskit.com
- Domain naming master (forest): Reskit.com
- RID master (domain): Reskit.com
- RID master (domain): Na.reskit.com
- RID master (domain): Eur.reskit.com
- Primary domain controller emulator (domain): Reskit.com
- Primary domain controller emulator (domain): Na.reskit.com
- Primary domain controller emulator (domain): Eur.reskit.com
- Infrastructure master (domain): Reskit.com
- Infrastructure master (domain): Na.reskit.com
- Infrastructure master (domain): Eur.reskit.com

A domain controller for any domain within a forest can hold a per-forest role for that forest; only a domain controller for a specific domain can hold a per-domain role for that domain. So, a single domain controller can hold up to five operations master roles including one of each role. Therefore, in the preceding example, the eleven roles might be held by as few as three domain controllers, or as many as eleven.

When you create the first Microsoft® Windows® 2000 domain controller in a forest, the Active Directory Installation Wizard assigns all five roles to it. When you create the first Windows 2000 domain controller for a new domain in an *existing* forest, the Active Directory Installation Wizard assigns the three per-domain roles to it.

In a mixed-mode domain environment (one that contains a mixture of Windows 2000, Microsoft® Windows NT® version 4.0, and Windows NT version 3.*51* domain controllers), only the Windows 2000 domain controllers can hold operations master roles.

Schema Master

The domain controller that holds the schema master role is the only domain controller that can perform write operations to the directory schema. Those schema updates are replicated from the schema master to all other domain controllers in the forest.

As previously mentioned, the schema master is a per-forest operations master role, because regardless of the number of domains, there is only one schema in the forest. To update the directory schema, you must connect to the domain controller holding the forest's schema master role. If, for some reason, you prefer that another domain controller hold the schema master role, you can transfer it to another domain controller.

The schema master role is managed from the Active Directory Schema snap-in. To determine if the Active Directory Schema snap-in is connected to the schema master, in the console tree, right-click the schema node and select **Operations Master**. If the **Current Focus** and **Current Operations Master** values are the same, you are connected to the schema master. Also, to enable schema changes you must select the check box **The Schema may be modified on this server**.

If you attempt to modify the schema using the Active Directory Schema snap-in while connected to a domain controller that is *not* the schema master, you will see an unspecific "attempted schema modification failed" error message.

For more information about using the Active Directory Schema snap-in, the schema master role, and extending the schema, see "Active Directory Schema" in this book.

Domain Naming Master

The domain controller that has the domain naming master role is the only domain controller that can do the following:

- Add new domains to the forest.
- Remove existing domains from the forest.
- Add or remove cross-reference objects to external directories.

By connecting to the domain controller holding the domain naming master role, you can add (or remove) a domain to (or from) the forest. If the domain naming master is unavailable, you cannot add or remove domains. If, for some reason, you prefer that another domain controller hold the domain naming master role, you can transfer it to another domain controller.

To add a domain to a forest, use one of the following methods:

- The Active Directory Installation Wizard. While creating the first domain controller of the domain, the wizard contacts the domain naming master by mean of RPC in order to create the domain. (Note that you must have sufficient access permissions to create the 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 server05.reskit.com). The
  attempt to contact it failed.
  The error was: "The specified server cannot perform the requested
  operation."
  ```

  In this example, "server05.reskit.com" is the domain naming master.

- The Precreate subcommand under the Domain Management option in the Ntdsutil command-line tool, followed by 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 found 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 server05.reskit.com
  binding to server05.reskit.com ...
  DsBindW error 0x6ba(The RPC server is unavailable.)
  ```

Relative Identifier Master

You can create a new security principal object (User, Group, or Computer) on any domain controller. However, after creating 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 repeats. 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.

When using Active Directory Users and Computers snap-in to create new objects, a message similar to the following displays when the domain controller's RID pool is empty and the domain's RID master is unavailable:

```
Active Directory Service
The object James Smith could not be created.
The problem encountered was:
The directory service has exhausted the pool of relative identifiers.
```

In this example, a new User object called "James Smith" could not be created because the RID master has exhausted its pool of RIDs.

To move objects from one domain (the source domain) to another (the destination domain) using the Movetree 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.

If you attempt to move an object from one domain to another using the Movetree tool and you specify a source domain controller that is not the RID master, you will see an unspecific "Movetree failed" error message.

For more information about RIDs, see "Access Control" in this book.

Primary Domain Controller Emulator

In a Windows NT 3.51 or Windows NT 4.0 system, each domain contains a primary domain controller. The functions of the primary domain controller include:

- Processing password changes from both users and computers
- Replicating updates to backup domain controllers
- Running the Domain Master Browser

Windows 2000 interoperates with Windows NT 3.51 and Windows NT 4.0 workstations, member servers, and domain controllers. Therefore one domain controller in a Windows 2000 system, the one holding the primary domain controller emulator role, must serve as primary domain controller for compatibility with older systems.

Active Directory uses multimaster 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. If the primary domain controller emulator is unavailable, you lose the services mentioned above. Symptoms include:

- When a user of a Windows NT Workstation 3.51–based computer, or a user of a computer running Windows NT Workstation 4.0, Windows 95, or Windows 98 without the Active Directory client installed, attempts a password change, the 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 Windows NT 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 Windows NT 4.0 backup domain controller results in a "domain unavailable" error message. If User Manager is already running, you will 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 will see a message similar to the following: "Cannot find the primary domain controller for <*domain name*>. You may administer this domain, but certain domainwide operations will be disabled."

As systems are upgraded, either to Windows 2000 or (for Windows NT Workstation 4.0, Windows 95, and Windows 98) by installing the Active Directory client, they cease to rely on the primary domain controller and, instead, behave in the following manner:

- Clients do not make password changes at the primary domain controller emulator. Instead, clients update passwords at any domain controller in the domain.
- 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 Windows NT 4.0 replication requests.
- Clients use Active Directory to locate network resources. They do not require the Windows NT Computer Browser service.

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.
- 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 might experience an increase in support requests regarding password difficulties.

For more information about upgrading Windows NT 3.51 and Windows NT 4.0 domains, see "Windows 2000 Upgrade and Installation"in the *Microsoft® Windows® Server Resource Kit Deployment Planning Guide*.

Infrastructure Master

Suppose you add a user to a group in the same domain using Active Directory Users and Computers. While still connected to the same domain controller, you can open up the group to examine its members and see the user you just added. If you then rename the user object (that is, change its cn attribute) and then display the group membership, you will instantly see the user's new name in the list of group members.

When the user and group are in different domains there is a time lag between when you rename a user object and when a group containing that user displays the user's new name. (In Active Directory Users and Computers, selecting the Members tab of the group's property page shows the user's old name in the Name column.) This time lag is inevitable in a distributed system where sites function independently.

The domain controller holding the infrastructure master role for the group's domain is responsible for updating the cross-domain group-to-user reference to reflect the user's new name. The infrastructure master updates these references locally and uses replication to bring all other replicas of the domain up to date. If the infrastructure master is unavailable, these updates are delayed.

For more information about viewing group membership, see Windows 2000 Server Help.

Placing Operations Master Roles

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.

Planning Role Placement

It is best to perform your planning of operations master role placements on a domain-by-domain basis. If a domain has only one domain controller, that domain controller holds all the per-domain roles. If a domain has more than one domain controller, choose two domain controllers that are direct replication partners and that are well-connected to the network. In a mixed-mode domain, it is recommended that these two domain controllers be located within the same site. For the rationale behind this guideline, see "Examining Operations Master Technical Details" later in this chapter.

Locating Replication Partners

To find the direct replication partners of a domain controller, use the Active Directory Sites and Services snap-in. Locate the domain controller by browsing to the Sites container to find the site containing the domain controller. Then, through the Servers container, go to the domain controller's Server object. 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.

Choosing Domain Controllers

If the two direct replication partner domain controllers you choose are in the same site, they are probably well connected. If they are in different sites, it is best for their sites to be connected through a network that is always available for replication; not one that is available for replication only at certain times of the day.

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. (As described in "Responding to Operations Master Failures," later in this chapter, you will use the "Standby operations master domain controller" if the "Operations master domain controller" fails.)

Determining Per-Domain Role Placements

In domains that are not large, place both the RID master and primary domain controller emulator roles on the "Operations master domain controller." In a very large domain, you can reduce the peak workload on the primary domain controller emulator by placing the RID master and primary domain controller emulator roles on separate domain controllers, which need to be direct replication partners of the "Standby operations master domain controller." Keep the two roles together unless the workload on your "Operations master domain controller" justifies the extra management burden of separating the roles.

Place the infrastructure master role on a domain controller that does not host a Global Catalog and that has a good network connection to a Global Catalog server (from any domain). Ideally, the infrastructure master needs to be within the same site as a Global Catalog server. If the "Operations master domain controller" meets these requirements, use it unless the infrastructure master role creates an unmanageable workload; in which case, separate domain controllers are justified.

Note The infrastructure master role needs to be held by a domain controller that is *not* a Global Catalog server. If the infrastructure master role is held by a domain controller that is a Global Catalog server, cross-domain object references in that domain will not be updated. If all domain controllers in a domain are Global Catalog servers, it does not matter which domain controller holds the infrastructure master role.

Determining Per-Forest Role Placements

When you have planned all of the per-domain roles, it is recommended you plan the per-forest roles next. The schema master and domain naming master roles should *always* be placed on the same domain controller. Place both roles on a domain controller that is close to the person or group responsible for schema updates and the creation of new domains. This domain controller *must* be a Global Catalog server because the domain naming master requires it. To simplify management, you can place these roles on the "Operations master domain controller" of a domain, if that domain controller is a Global Catalog server.

Maintaining Role Placements

In most cases, your role placement plan does not require changes as your forest grows, which means that role placements do not require revisions. However, when you plan to decommission a domain controller, reduce the connectivity of your network, or change the Global Catalog server status of a domain controller, you need to review your plan and revise it as necessary.

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 upon the role, you can perform role transfers 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 **OK** to complete the operation.

If the current role owner is available, the transfer completes within a few seconds. If the transfer is not completed within a short period of time, the domain controller is not available. In this case, you need to follow the recommendations for responding to failures, which is explained in the next section.

For more information about procedures for performing operations master role transfers, see Windows 2000 Server Help.

Additionally, you can use the Active Directory snap-ins to view the actual roles that a domain controller owns. To accomplish this, you would choose one of the Active Directory snap-ins, right-click the root node of the snap-in in the consoler tree, and select **Operations Master**. The **Operations** dialog box displays the name of the domain controller that is the current focus and its status.

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, the recommended response is simply to wait for the role owner to become available before performing a role-related function.

If the outage is longer, the correct response might be to seize the operations master role from a domain controller. To *seize* a role is to move it *without* the cooperation of its current owner. It is best to avoid seizing roles. The decision to seize an operations master role depends upon 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, either users or administrators. Specifically, an end user running Windows NT Workstation 3.51, or Windows NT 4.0, Windows 95, or Windows 98 without the Active Directory client, cannot change their password without communicating with the primary domain controller emulator. If the user's password has expired, the user is not able to log on. Therefore, you might need to repair a primary domain controller emulator failure quickly.

If the primary domain controller emulator is offline for a significant period of time and the domain has users running Windows NT Workstation 3.51, or Windows NT 4.0, Windows 95, or Windows 98 without the Active Directory client, or domain controllers running earlier versions of Windows NT, you should seize the primary domain controller emulator role to the "Standby operations master domain controller."

The user interface for this seizure is similar to that of a normal operations master role transfer, except it requires an extra confirmation from you. Agree to the confirmation *only* if you know the current primary domain controller emulator will be offline for a significant period. Later, when the original primary domain controller emulator domain controller comes back online, transfer the role back to the original role owner.

Infrastructure Master Failures

Temporary loss of a domain's infrastructure master is not visible to end users, and is not visible to you, as an administrator, unless you recently moved or renamed a large number of accounts. Therefore, in most cases, a temporary loss of the infrastructure master is not a problem worth fixing.

If you anticipate a long outage of a domain's infrastructure master and you need to repair it, first select a domain controller that is not a Global Catalog server and that has good network connectivity to a Global Catalog server located in any domain. Ideally, the domain controller you have chosen should be within the same site as a Global Catalog server. It is not important that the new infrastructure master be near the previous one. When you have selected the domain controller, seize the infrastructure master role to this domain controller.

The user interface for this seizure is similar to that of a normal operations master role transfer, except it requires an extra confirmation from you. Agree to the confirmation *only* if you know that the current infrastructure master will be offline for a very long period. Later, when the original infrastructure master comes back online, transfer the role back to the original role owner.

Other Operations Master Failures

Temporary loss of the schema master, domain naming master, or RID master is ordinarily not visible to end users, and does not usually inhibit your work as an administrator. Therefore, this is usually not a problem worth fixing.

However, if you anticipate an extremely long outage of the domain controller holding one of these roles, you can seize that role to the "Standby operations master domain controller." But, seizing any of these roles is a *drastic* step; one that you would take only when the outage is permanent, as in the case when a domain controller is physically destroyed and cannot be restored from backup media.

A domain controller whose schema master, domain naming master, or RID master role is seized must *never* come back online. Before proceeding with the role seizure, you must ensure that the outage of this domain controller is permanent by physically disconnecting the domain controller from the network.

The domain controller that seizes the role should be fully up-to-date with respect to updates performed on the previous role owner. Because of replication latency, it is possible that 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. The Repadmin command-line tool 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 more information about using the Repadmin tool, see *Windows 2000 Support* Tools Help, which is included on the *Windows 2000 Server* CD and "Active Directory Diagnostics, Troubleshooting, and Recovery" in this book.

For example, to make sure a domain controller is fully up-to-date, suppose that "server05" is the RID master of the domain "reskit.com," "server10" is the "Standby operations master domain controller," and "server12" is the only other domain controller in the "reskit.com" domain. Using the Repadmin tool, you would issue the following commands:

```
C:\>repadmin /showvector dc=reskit,dc=com server10.reskit.com
New-York\server05    @ USN 2604
San-Francisco\server12 @ USN 2706

C:\>repadmin /showvector dc=reskit,dc=com server12.reskit.com
New-York\server05    @ USN 2590
Chicago\server10    @ USN 3110
```

Note In the previous example, user input is in **bold** type.

Ignore all output lines except those for server05. Server10's up-to-date status value with respect to server05 (server05 @ USN 2604) is larger than server12's up-to-date status value with respect to server05 (server05 @ USN 2590), making it is safe for server10 to seize the RID master role formerly held by server05. If the up-to-date status value for server10 was less than the value for server12, you would wait for normal replication to update server10, or use the Repadmin tool's **/sync/force** commands to make the replication happen immediately.

After you have determined that the role owner is fully up-to-date, you can seize the operations master role using the Ntdsutil tool as in the following example:

```
C:\>ntdsutil
ntdsutil: roles
fsmo maintenance: connections
server connections: connect to server10.reskit.com
binding to server10.reskit.com ...
Connected to server10.reskit.com
 using credentials of locally logged on user
server connections: quit
fsmo maintenance: seize RID master
Server "server10.reskit.com" knows about 5 roles
Schema - CN=NTDS Settings,CN=server04,CN=Servers,
 CN=New-York,CN=Sites,CN=Configuration,DC=reskit,DC=com
Domain - CN=NTDS Settings,CN=server04,CN=Servers,
 CN=New-York,CN=Sites,CN=Configuration,DC=reskit,DC=com
PDC - CN=NTDS Settings,CN=server10,CN=Servers,
 CN=Chicago,CN=Sites,CN=Configuration,DC=reskit,DC=com
RID - CN=NTDS Settings,CN=server10,CN=Servers,
 CN=Chicago,CN=Sites,CN=Configuration,DC=reskit,DC=com
Infrastructure - CN=NTDS Settings,CN=server12,CN=Servers,
 CN=San-Francisco,CN=Sites,CN=Configuration,DC=reskit,DC=com
fsmo maintenance: quit
ntdsutil: quit
C:\>
```

Note In the previous example, user input is in **bold** type.

For more information about specific procedures for using the Ntdsutil command-line tool, see *Windows 2000 Support* Tools Help, which is included on the *Windows 2000* Server installation CD.

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 it is simpler and quicker to enter commands than to use multiple windows.

To perform seizures of the schema master, domain naming master, and RID master roles, the Ntdsutil tool is the required method. 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 (?). The following is an example showing the transfer of the domain naming master role (with user input shown in **bold** type):

```
C:\>ntdsutil
ntdsutil: ?

 ?               - 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

ntdsutil: roles
fsmo maintenance: ?

 ?               - Print this help information
 Connections          - Connect to a specific domain controller
 Help             - Print this help information
 Quit             - Return to the prior menu
 Seize domain naming master  - Overwrite domain role on connected server
 Seize infrastructure master  - Overwrite infrastructure role on
connected server
 Seize PDC           - Overwrite PDC role on connected server
 Seize RID master       - Overwrite RID role on connected server
 Seize schema master      - Overwrite schema role on connected server
 Select operation target    - Select sites, servers, domains, roles and
Naming Contexts
 Transfer domain naming master - Make connected server the domain naming
master
 Transfer infrastructure master - Make connected server the
infrastructure master
 Transfer PDC         - Make connected server the PDC
 Transfer RID master      - Make connected server the RID master
 Transfer schema master    - Make connected server the schema master
```

```
fsmo maintenance: connections
server connections: ?

 ?               - Print this help information
 Clear creds          - Clear prior connection credentials
 Connect to domain %s     - Connect to DNS domain name
 Connect to server %s     - Connect to server, DNS name or IP address
 Help             - Print this help information
 Info             - Show connection information
 Quit             - Return to the prior menu
 Set creds %s %s %s      - Set connection creds as domain, user, pwd
                 Use "NULL" for null password

server connections: connect to server reskit1
Binding to reskit1 ...
Connected to reskit1 using credentials of locally logged on user
server connections: quit
fsmo maintenance: transfer domain naming master
Server "reskit1" knows about 5 roles
Schema - CN=NTDS
Settings,CN=RESKIT1,CN=Servers,CN=Washington,CN=Sites,CN=Configuration,D
C=reskit,DC=com
Domain - CN=NTDS
Settings,CN=RESKIT1,CN=Servers,CN=Washington,CN=Sites,CN=Configuration,D
C=reskit,DC=com
PDC - CN=NTDS
Settings,CN=RESKIT1,CN=Servers,CN=Washington,CN=Sites,CN=Configuration,D
C=reskit,DC=com
RID - CN=NTDS
Settings,CN=RESKIT1,CN=Servers,CN=Washington,CN=Sites,CN=Configuration,D
C=reskit,DC=com
Infrastructure - CN=NTDS
Settings,CN=RESKIT1,CN=Servers,CN=Washington,CN=Sites,CN=Configuration,D
C=reskit,DC=com
fsmo maintenance: quit
ntdsutil: quit
Disconnecting from reskit1 ...

C:\>
```

In the previous example, the available Ntdsutil tool commands display after entering a question mark (?). To transfer an operations master role, the **roles** command is entered, 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 example above) by entering the **connect to server** subcommand. Then, after leaving the server connections mode by entering "quit", issue the **transfer domain naming master** command. A confirmation pop-up window (not shown) displays for the transfer domain naming master operation.

Note You must have sufficient permissions to execute commands using the Ntdsutil tool. For more information about controlling access to operations master role placements, see "Controlling Access to Role Placements" later in this chapter.

It is also possible to view the current operations master role owner using the Ntdsutil command-line tool from the **Select Operation Target** menu located under the **Roles** option. By using the **List roles for connected server** command, a list displays of all of the current operations master role owners.

For more information about using the Ntdsutil command-line tool, see *Windows 2000 Support* Tools Help, which is included on the *Windows 2000* Server installation CD.

Controlling Access to Role Placements

Active Directory represents operations master role ownership through the Active Directory schema FSMO-Role-Owner attribute. The LDAP display name for this attribute is fsmoRoleOwner. The FSMO-Role-Owner attribute is a single-valued directory object reference with a value that always references an NTDS-DSA object.

The NTDS-DSA object represents a specific domain controller. For example, the distinguished name (DN) for such an NTDS-DSA object is:

```
CN=NTDS
Settings,CN=RESKIT1,CN=Servers,CN=Washington,CN=Sites,CN=Configuration,D
C=reskit,DC=com
```

where RESKIT1 is the name of the domain controller in the Washington site in the Reskit.com domain.

In principle, any directory object can have a value for the FSMO-Role-Owner attribute. However, Active Directory only sets the FSMO-Role-Owner attribute of certain objects; one per role in a directory. This directory object is called the "role object" for its role. The FSMO-Role-Owner of a role object is the domain controller holding that role.

Following are the distinguished names of the role objects:

- Schema master:

  `CN=Schema,CN=Configuration,<root domain> (root of Schema directory partition)`

- Domain naming master:

  `CN=Partitions,CN=Configuration,<root domain> (within Configuration directory partition)`

- RID master:

  `CN=RID Manager$,CN=System,<any domain> (within domain directory partition)`

- Primary domain controller emulator:

  `<any domain> (root of domain directory partition)`

- Infrastructure master:

  `CN=Infrastructure,<any domain> (within domain directory partition)`

For example, in the Reskit.com forest that has the Reskit.com, Na.reskit.com, and Eur.reskit.com domains, the eleven role objects and associated distinguished name are:

- Schema master role object for the forest Reskit.com:

  `CN=Schema,CN=Configuration,DC=reskit,DC=com`

- Domain naming master role object for the forest Reskit.com:

  `CN=Partitions,CN=Configuration,DC=reskit,DC=com`

- RID master role object for the domain Reskit.com:

  `CN=RID Manager$,CN=System,DC=reskit,DC=com`

- RID master role object for the domain Na.reskit.com:

  `CN=RID Manager$,CN=System,DC=na,DC=reskit,DC=com`

- RID master role object for the domain Eur.reskit.com:

  `CN=RID Manager$,CN=System,DC=eur,DC=reskit,DC=com`

- Primary domain controller emulator role object for the domain Reskit.com:
  `DC=reskit,DC=com`
- Primary domain controller emulator role object for the domain Na.reskit.com:
  `DC=na,DC=reskit,DC=com`
- Primary domain controller emulator for the domain Eur.reskit.com:
  `DC=eur,DC=reskit,DC=com`
- Infrastructure master role object for the domain Reskit.com:
  `CN=Infrastructure,DC=reskit,DC=com`
- Infrastructure master role object for the domain Na.reskit.com:
  `CN=Infrastructure,DC=na,DC=reskit,DC=com`
- Infrastructure master role object for the domain Eur.reskit.com
  `CN=Infrastructure,DC=eur,DC=reskit,DC=com`

Note If an object is not a role object, the value of its FSMO-Role-Owner attribute does not have an effect on Active Directory functionality.

Controlling Role Transfers

As previously defined, an operations master role transfer is the movement of a role *with* the cooperation of its current owner. To perform a role transfer, both domain controllers must be available and connected to each other through the network.

The ability to perform a role transfer is controlled through a special object permission on the role object itself at the new role owner. There is a different object permission for each role, granted by default to a particular group of administrators:

- Schema master: The Change Schema Master permission is granted by default to the Schema Admins group.
- Domain naming master: The Change Domain Master permission is granted by default to the Enterprise Admins group.
- RID master: The Change Rid Master permission is granted by default to the Domain Admins group.
- Primary domain controller emulator: The Change PDC permission is granted by default to the Domain Admins group.
- Infrastructure master: The Change Infrastructure Master permission is granted by default to the Domain Admins group.

If you attempt to perform a role transfer and you 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 ADSI Edit 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 example above, the Domain Naming Role Admins can transfer the domain naming master role, but they cannot create cross ref objects; only Enterprise Admins can do that.

For more information about setting access control permissions, see "Access Control" in this book, or see Windows 2000 Server Help.

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 controls role transfers, plus the Write fsmoRoleOwner property permission at the new role owner. 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.

Scripting Role Placement

You can also place operations master role owners programmatically for both role transfers and seizures through Microsoft® Visual Basic® Script programs.

Scripting Role Transfers

Active Directory operations master role transfers are exposed as an LDAP update operation to a root DSE operational attribute of the domain controller taking the role. There is a root DSE operational attribute corresponding to each role:

- *becomeSchemaMaster*
- *becomeDomainMaster*
- *becomeRidMaster*
- *becomePdc*
- *becomeInfrastructureMaster*

For more information about the root DSE operational attributes, see "Active Directory Data Storage" in this book.

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:

```
Set dse = GetObject("LDAP://localhost/RootDSE")
dse.Put "becomeDomainMaster", 1
dse.SetInfo
```

Scripting Role Seizures

Active Directory role seizures are exposed as an LDAP update operation 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. If the seizure fails, it prints an error message.

```
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
```

For more information about Visual Basic Script programs, see *Windows 2000 Resource Kit* Tools Help, which is included on the *Windows 2000 Resource Kit* companion CD.

Examining Operations Master Technical Details

Use the following list to obtain more complete technical explanations concerning the management of operations masters:

- *Why must the RID master be available when creating a large number of security principal objects?*

  When a domain controller creates a security principal object, it attaches a unique Windows NT Security ID (SID) to the object. A SID consists of a domain SID that is the same for all SIDs created in a domain, and a relative ID (RID) that is different for each SID created in a domain.

  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.

- *Why must cross-domain object moves originate on the RID master?*

  In Active Directory, you can move an object from one domain to another. You can only move an object out of its domain on the domain's RID master. This prevents Active Directory from creating two objects in different domains with the same unique identifier. (This scenario could happen if an object were simultaneously moved from two domain controllers to two different domains.)

- *Why must the infrastructure master not be a Global Catalog server?*

  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 GUID, the SID (for references to 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 cross-domain, and its distinguished name always changes.

  The infrastructure master for a domain periodically examines the references, within its replica of the directory data, to 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 will never update anything, because it does not contain any references to objects that it does not hold. That is because a Global Catalog server holds a partial replica of every object in the forest.

- *Why must the domain naming master also be a Global Catalog server?*

  When the domain naming master creates an object representing a new domain, it must make sure that no other object—domain object or otherwise—has the same name. The domain naming master achieves this by running on a Global Catalog server, which contains a partial replica of every object in the forest.

- *What are the special considerations for role placement in mixed-mode domains?*

  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. By keeping both domain controllers in the same site, the system can avoid performing a full synchronization with the backup domain controllers in case you seize the PDC emulator role to the standby operations master domain controller.

- *What do you do if a role transfer is not completed?*

  When a role transfer takes place, it updates the current role owner before it updates the desired new role owner. If the desired new role owner fails before making its update, it does not yet hold the role. The desired new role owner can gain ownership of the role in the following ways:

  - Typically, you would repeat the role transfer attempt.
  - Allow replication to update the desired new role owner with the change made at the current role owner. (This does not require any action on your part, but it does take more time than repeating the role transfer attempt.)

- *What happens to operations master roles during backup and restore procedures?*

  When you back up a domain controller, you back up the roles it owns.

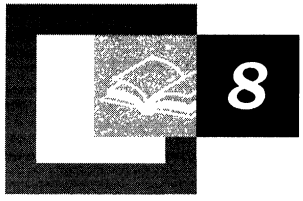
  So, when you restore a domain controller from backup media, you restore the roles it owns.

- *What happens to operations master roles during the demotion process?*

  When you remove Active Directory 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 not succeed.

  Do not rely on the transfer feature when removing Active Directory from a domain controller. Instead, transfer any roles before you begin the removal process so that role placements are as they should be.

CHAPTER 8



Monitoring Performance in Active Directory

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 to verify 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, and Microsoft® Visual Basic® scripts.

In This Chapter

Performance Tools

By using a combination of MMC snap-ins and Performance Monitoring tools, such as command-line tools and Visual Basic scripts, you can monitor many activities in Active Directory, the Windows 2000 directory service. For example, activities such as monitoring replication topology, Domain Name System (DNS) functionality, latency, connection times, and allocation of relative identifiers (RIDs) can each be monitored by using these tools.

MMC Snap-ins

The Windows 2000 MMC provides several snap-ins that are available from the **Administrative Tools** menu. The Active Directory Sites and Services snap-in enables you to view information that is related to sites and replication topology. The Performance snap-in provides a graphical way to view performance of Active Directory according to measurements, or counters, that you select. Also, you can log activity according to those counters and print them or view them online. Finally, Event Viewer enables you to view log files and error messages sent by applications.

Performance

The Performance snap-in enables you to monitor the performance of local and remote computers anywhere in your network and summarize the performance at selected intervals. The Performance snap-in uses a range of counters for monitoring real-time resource usage. The Performance snap-in can log results to a file so that you can analyze the performance history of a computer. You can monitor resource usage of other Windows 2000 servers on the network providing you have the right permission. You can also use the Performance snap-in to collect baseline performance data and send alerts to the event log about exceptions to the baseline that you configure.

The Performance snap-in is a command on the **Administrative Tools** menu, and it has two tools: System Monitor and Performance Logs and Alerts.

For more information about performance monitoring tools, see "Overview of Performance Monitoring" in the *Microsoft Windows 2000 Server Resource Kit Server Operations Guide*.

System Monitor

The System Monitor tool enables you to display performance data in real-time charts and reports, collect data in files, and generate alerts that warn you when critical events occur. This information is classified as a performance object by the component (whether it is a service, computer, or mechanism) that generates the data. Examples of performance objects include Processor, Print Queue, and SMTP service. To monitor Active Directory, you monitor the activity reported by the **NTDS** performance object.

By using System Monitor, you can track the activity of performance objects through the use of counters. Performance counters measure the parameters that apply to a performance object. For example, if you need to discover whether a server is receiving directory replication updates and applying the updates in a timely fashion, you can select one or more counters from the **NTDS** object and then view the current activity by using System Monitor. For more information about using System Monitor, see System Monitor Help.

For example, to ensure a timely replication of all network servers, you can use the DRA Pending Replication Synchronizations counter from the **NTDS** object to check the number of directory synchronizations that are queued for a server but not yet processed. This counter provides a visual representation of how quickly objects are being applied to the server, which helps you to determine whether a server is replicating normally.

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 service Lightweight Directory Access Protocol (LDAP), Key Distribution Center (KDC), Security Accounts Manager (SAM), Local Security Authority (LSA), and Net Logon service. 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 tool. To produce transaction-level costing information trace data, you must use the trace application programming interfaces (APIs). For more information about trace APIs for third-party systems management vendors, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Similar to the System Monitor tool, the Performance tool 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.

For more information about using the Performance snap-in, see Windows 2000 Server Help.

Event Viewer

Windows 2000 provides the Event Viewer snap-in as a way to monitor system events, such as application or system errors and the successful starting of services. These events are recorded in event logs. For example, if you need detailed information about when 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. For more information about how to use the Event Viewer snap-in, see Windows 2000 Server Help.

Active Directory Performance Tools

This section describes the command-line tools and Visual Basic scripts, provided on the *Windows 2000 Resource Kit* companion CD, that enable you to monitor performance of Active Directory, including monitoring differences between directory partitions, replication schedules and topology, networking, channels, and local and remote computer performance, and the allocation of RID pools. They are located in the directory *<Root Directory of the CD-ROM>*\Support\Reskit. The performance monitor (Pmon) tool and two Visual Basic (VBS) scripts, DisplayRID and DisplayOld, are furthered described in this chapter.

Pmon

You can use the Process Monitor (Pmon.exe) command-line tool to monitor process resource usage by tracking CPU and memory usage for each running process. Pmon.exe also measures paged and nonpaged pool usage, and identifies kernel-mode memory leaks. For Active Directory purposes, you can use Pmon.exe to monitor the memory usage of the Local Security Authority by viewing the Lsass.exe process.

The Process Monitor tool provides a keyboard interface, so you can use the UP ARROW and DOWN ARROW to scroll up and down the list of currently running processes. To quit Pmon.exe, press ESC or type **q**. Use any other key to refresh the Pmon.exe display.

For more information about paged and nonpaged pool usage, see "Evaluating Memory and Cache Usage" in the *Microsoft Windows 2000 Server Resource Kit Server Operations Guide*.

DisplayRID

You use the Visual Basic script DisplayRID to monitor the allocation and use of RID pools by domain controllers. For example, to monitor the ResKit domain's allocation and use of its RID pool, type:

```
cscript displayrid.vbs /D:ResKit /U:Sales\Administrator /W:mypass1
```

The general syntax of the command is as follows:

Cscript displayrid.vbs [/U:*Username***] [/W:***Password***] [/D:***Domain***] [/S:***Server***] [/Q]**

DisplayOld

You use the Visual Basic script DisplayOld to query Active Directory to find computer or user accounts that have not logged on for a specified period of time. The output can be optionally archived to a file in LDAP Data Interchange Format (LDIF) so that you can delete those objects from Active Directory. For example, to find all user accounts that have not logged on for the past 60 days in the Sales organizational unit (OU) within the Reskit.com domain, type the following command:

```
cscript displayold.vbs /N:60 /C:user /D:ResKit /U:Sales\Administrator
/W:mypass1
```

The general syntax of the command is as follows:

Cscript displayold.vbs /N:*Number_of_days* **[/U:***User_name***] [/W:***Password***] [/D:***Domain***] [/S:***Server***] [/C:***Computer|User***] [/F:***File_name***] [/Q]**

Note that only one parameter, **/N:***Number_of_days*, is required.

Trace Logging

Trace logging enables you to measure the performance cost of an operation. Active Directory provides instrumentation to trace the operations of directory services, such as LDAP Primitives, Kerberos, SAM, LSA, and Net Logon. When enabled, tracing captures key events such as logons, authentications, LDAP operations, and SAM operations. It also records parameters such as CPU time, timestamps, and thread identifiers. By using the event logging services of the Performance Logs and Alerts service, you can enable or disable trace logging. To produce transaction-level costing information trace data, you must use the trace APIs. For more information about trace APIs, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. The Microsoft Platform SDK and Resource Kit also includes sample programs and tools to post-process trace data.

System Monitor Counters

When the Active Directory Installation Wizard installs Active Directory, the wizard configures performance counters in the **NTDS** object that provide statistics about directory activity. These counters apply to the entire directory, including global catalogs. By using the Performance snap-in, you can graphically view current activity, and also send the activity to a log file.

NTDS Object

The **NTDS** object contains performance counters that provide statistics about Active Directory performance. For example, there are several counters associated with the directory replication agent (DRA), which monitors replication activity. Because there are many performance counters from which to choose, you first need to determine the statistics you want to monitor and then find the matching performance counter.

These counters can also help during capacity planning by providing some baseline analysis. For example, you can create events that run whenever performance exceeds your baseline criteria. Typically, counters that are suited for capacity planning contain the word "Total" in their name, and come in three types:

- *Statistic.* Shows total actions per second. For example, the counter DRA Inbound Properties Total/Sec shows the total number of object properties received from inbound replication partners.
- *Ratio.* Shows the percentage of total activity. For example, the counter DS % Writes from LDAP shows the percentage of directory writes coming from LDAP queries.
- *Accumulative.* Shows activity since the last start of the computer. For example, the counter DRA Inbound Bytes Total Since Boot shows the total number of inbound bytes replicated, the sum of the number of bytes of uncompressed data (never compressed) and compressed data (after compression).

Each counter has its own guidelines and limits. The counters listed in Table 8.1 are important to performance monitoring for the reasons described.

Table 8.1 Active Directory System Monitor Counters on the NTDS Object

Counter	Description
DRA Inbound Bytes Compressed (Between Sites, After Compression)/sec	The compressed size (in bytes) of compressed replication data inbound from directory system agents (DSAs) in other sites (per second).
DRA Inbound Bytes Compressed (Between Sites, Before Compression)/sec	The uncompressed size (in bytes) of compressed replication data inbound from DSAs in other sites (per second).
DRA Inbound Bytes Not Compressed (Within Site)/sec	The uncompressed size (in bytes) of replication data that was not compressed at the source—that is, inbound from other DSAs in the same site (per second).
DRA Inbound Bytes Total/sec	The total number of bytes (per second) received through replication. It is the sum of the number of bytes of uncompressed data (never compressed) and compressed data (after compression).
DRA Inbound Full Sync Objects Remaining	The number of objects remaining until the full synchronization process is completed.
DRA Inbound Objects/sec	The number of objects received (per second) through inbound replication from replication partners.

(continued)

Table 8.1 Active Directory System Monitor Counters on the NTDS Object *(continued)*

Counter	Description
DRA Inbound Objects Applied/sec	The number of objects received (per second) from replication partners and applied by the local directory service. This counter excludes changes that are received but not applied (for example, when the update is already made). This counter indicates how many replication updates are occurring on the server as a result of changes generated on other servers.
DRA Inbound Objects Filtered/sec	The number of objects received (per second) from replication partners that contained no updates that needed to be applied.
DRA Inbound Object Updates Remaining in Packet	The number of object updates received in the current directory replication update packet that have not yet been applied to the local server. This counter tells you whether the monitored server is receiving changes, but is taking a long time applying them to the database.
DRA Inbound Properties Applied/sec	The number of changes (per second) to object properties that are applied through inbound replication as a result of reconciliation logic.
DRA Inbound Properties Filtered/sec	The number of changes (per second) to object properties received during the replication that are already made.
DRA Inbound Properties Total/sec	The total number of changes (per second) to object properties received from replication partners.
DRA Inbound Values (DNs only)/sec	The number of values of object properties received (per second) from replication partners in which the values are for object properties that belong to distinguished names. This number includes objects that reference other objects. Values for distinguished names, such as group or distribution list memberships, are more expensive to apply than other kinds of values because a group or distribution list object can include hundreds or thousands of members. In contrast, a simple object might have only one or two attributes. A high number from this counter might explain why inbound changes are slow to be applied to the database.
DRA Inbound Values Total/sec	The total number of values of object properties received (per second) from replication partners. Each inbound object has one or more properties, and each property has zero or more values. A value of zero indicates that the property is to be removed.

(continued)

Table 8.1 Active Directory System Monitor Counters on the NTDS Object

Counter	Description
DRA Outbound Bytes Compressed (Between Sites, After Compression)/sec	The compressed size (in bytes) of compressed replication data that is outbound to DSAs in other sites (per second).
DRA Outbound Bytes Compressed (Between Sites, Before Compression)/sec	The uncompressed size (in bytes) of compressed replication data outbound to DSAs in other sites (per second).
DRA Outbound Bytes Not Compressed (Within Site)/sec	The uncompressed size (in bytes) of outbound replication data that was not compressed—that is, outbound to DSAs in the same site—per second.
DRA Outbound Bytes Total/sec	The total number of bytes sent per second. It is the sum of the number of bytes of uncompressed data (never compressed) and compressed data
DRA Outbound Objects Filtered/sec	The number of objects (per second) acknowledged by outbound replication partners that required no updates. This counter includes objects that the outbound partner did not already have.
DRA Outbound Objects/sec	The number of objects sent (per second) though outbound replication to replication partners.
DRA Outbound Properties/sec	The number of properties sent per second. This counter tells you whether a source server is returning objects or not. Sometimes, the server might stop working correctly and not return objects quickly or at all.
DRA Outbound Values (DNs only)/sec	The number values of object properties sent (per second), to replication partners in which the values are for object properties that belong to distinguished names. Values for distinguished names, such as group or distribution list memberships, are more expensive to apply than other kinds of values because a group or distribution list object can include hundreds or thousands of members. In contrast, a simple object might have only one or two attributes.
DRA Outbound Values Total/sec	The total number of values of object properties sent (per second), to replication partners.

(continued)

Table 8.1 Active Directory System Monitor Counters on the NTDS Object *(continued)*

Counter	Description
DRA Remaining Replication Updates	The number of changes to objects that have been received in the current directory replication update packet for the DRA that have not yet been applied to the local server. A sharp decline in the rate at which objects are applied to the database indicates normal operation, while a gradual decline indicates that complex objects are being applied. This counter is a gauge of whether a server is slow to replicate.
DRA Pending Replication Synchronizations	The number of directory synchronizations that are queued for this server that are not yet processed. This counter helps in determining replication backlog—the larger the number, the larger the backlog.
DRA Sync Requests Made	The number of synchronization requests made to replication partners since computer was last restarted.
DS Security Descriptor Suboperations/sec	The number of suboperations (per second) of security descriptor propagation. One operation of security descriptor propagation comprises many suboperations. There is approximately one suboperation for each object that the propagation operation causes the propagator to examine.
DS Security Descriptor Propagation Events	The number of events of Security Descriptor Propagation that are queued but not yet processed.
DS Threads in Use	The current number of threads in use by the directory service (different from the number of threads in the directory service process). This counter represents the number of threads currently servicing API calls by clients, and you can use it to determine whether additional CPUs would be beneficial.
LDAP Client Sessions	The number of sessions of connected LDAP clients.
LDAP Bind Time	The time (in milliseconds) required for the completion of the last successful LDAP binding.
Kerberos Authentications/sec	The number of times per second that clients use a client ticket to this domain controller to authenticate to this domain controller.
NTLM Authentications/sec	The number of NTLM authentications (per second) serviced by this domain controller.
LDAP Successful Binds/sec	The number LDAP bindings (per second) that occurred successfully.
LDAP Searches/sec	The number of search operations per second performed by LDAP clients.

Manually Loading and Unloading NTDS Counters

In rare cases, the **NTDS** object can fail to load. If this happens, the **NTDS** object is not available when you open System Monitor.

▶ **To load the NTDS object manually**

1. Change directory to *%SystemRoot%*\System32.
2. To load the counter information into the registry, at the command prompt, type **lodctr.exe ntdsctrs.ini**.
3. To enable collection of performance data for Active Directory, restart your computer.

After these steps are finished, you can use the System Monitor tool to view the counters for the **NTDS** object.

▶ **To unload the NTDS object manually**

1. Change directory to *SystemRoot*\System32.
2. To unload the counter information from the registry at the command prompt, type **unlodctr.exe ntds**.

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. The counters on the Database object enable you to perform advanced tuning of Active Directory. You can also use some of the counters to help determine whether you need more disk drives for storage of logs or database.

Currently, there is no automated way to install the performance dynamic-link library (DLL), Esentprf.dll, in Windows 2000.

Caution Do not use a registry editor to edit the registry directly unless you have no alternative. The registry editors bypass the standard safeguards provided by administrative tools. These safeguards prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your system. Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. To configure or customize Windows 2000, use the programs in Control Panel or Microsoft Management Console (MMC) whenever possible.

▶ To load the Database object

1. Copy the performance DLL (Esentprf.dll) located in *SystemRoot*\System32 to any directory (for example, C:\Perf).
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 these subkeys do not exist, you need to create them.

   For more information about creating registry subkeys, see Windows 2000 Server Help.
3. Make sure that, under the Performance subkey, the registry values that have the following settings exist:
   - Open : data type REG_SZ : **OpenPerformanceData**
   - Collect : data type REG_SZ : **CollectPerformanceData**
   - Close : data type REG_SZ : **ClosePerformanceData**
   - Library : 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 generated when Eseperfnt.dll was compiled.
5. (Optional) To verify that previous counter information is not present in the registry, at the command prompt, type **unlodctr.exe ESENT**.
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.

Table 8.2 lists the performance counters for the Database object.

Table 8.2 Additional Active Directory System Monitor Counters Under the Database Object

Counter	Description
Cache % Hit	The percentage of page requests for the database file that were fulfilled by the database cache without causing a file operation. If this percentage is too low, the size of the database cache might be too small. You can increase the size of the database cache by adding more RAM.
Cache Page Fault Stalls/sec	The number of page faults (per second) that cannot be serviced because there are no pages available for allocation from the database cache. If this counter is nonzero most of the time, the clean threshold might be too low. You can increase the size of the database cache by adding more RAM.
Cache Page Faults/sec	The number of page requests (per second) for the database file that require the database cache manager to allocate a new page from the database cache. If this number is too high, the size of the database cache might be too small. You can increase the size of the database cache by adding more RAM.
Cache Size	The amount of system memory used by the database cache manager to hold commonly used information from the database file or files to prevent file operations. If the size of the database cache seems to be too small for optimal performance and there is very little system memory available on the system, adding more RAM to the system might increase performance. If there is plenty of system memory available and the database cache size is not growing, the database cache size might be restricted to an artificially low limit. You can increase the size of the database cache by adding more RAM.
File Bytes Read/sec	The number of bytes that are read (per second) from the database file or files into the database cache. If this rate is too high, the size of the database cache might be too small. You can increase the size of the database cache by adding more RAM.

(continued)

Table 8.2 Additional Active Directory System Monitor Counters Under the Database Object

Counter	Description
File Bytes Written/sec	The number of bytes that are written (per second) to the database file or files from the database cache. If this rate is too high, the size of the database cache might be too small. You can increase the size of the database cache by adding more RAM.
File Operations Pending	The number of reads and writes issued by the database cache manager to the database file or files that the operating system is currently processing. A large number of pending operations increases system throughput, but also increases the time required to process individual operations. Typically, a larger number means that file operations on the database file or files might be a bottleneck.
File Operations/sec	The number of reads and writes (per second) issued by the database cache manager to the database file or files. If this number is too high, the database cache size might be too small. You can increase the size of the database cache by adding more RAM.
Log Record Stalls/sec	The number of instances (per second) that a log record cannot be added to the log buffers because the buffers are full. If this counter is not zero most of the time, the size of the log buffer might be a bottleneck.
Log Threads Waiting	The number of threads waiting for their data to be written to the log in order to complete an update of the database. If this number is too high, the log might be a bottleneck.
Log Writes/sec	The number of instances (per second) that the log buffers are written to the log file(s). If this number approaches the maximum rate at which the medium that is storing the log file(s) can write data, the medium might be a bottleneck.
Table Open Cache % Hit	The percentage of database tables opened by using cached schema information. If this number is too low, the size of the table cache might be too small. You can increase the size of the database cache by adding more RAM.

(continued)

Table 8.2 Additional Active Directory System Monitor Counters Under the Database Object *(continued)*

Counter	Description
Table Open Cache Hits/sec	The number of database tables opened (per second) by using cached schema information. If this number is too low, the size of the table cache might be too small. You can increase the size of the database cache by adding more RAM.
Table Open Cache Misses/sec	The number of database tables opened (per second) without using cached schema information. If this number is too high, the size of the table cache might be too small. You can increase the size of the database cache by adding more RAM.
Table Opens/sec	The number of database tables opened per second.

Note When the description of a counter indicates that the number counted is "too high" or "too low," it is an indication of how to adjust the system configuration. For example, if you make a change that increases the rate of page faults for a cache, you know that you have made an incorrect adjustment, and you must change the settings in the opposite direction. Please note that you cannot adjust settings that cause the counter behavior to change in System Monitor. Settings are configured elsewhere, such as in the registry or in startup .inf files.

Using System Monitor to Select Performance Counters

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. Also, when you have a counter selected, you can view a description of that counter and tips on how to improve system performance by clicking the **Explain** button.

▶ **To select Active Directory performance counters to monitor**

1. On the **Start** menu, click **Programs,** point to **Administrative Tools**, and then click **Performance**. The Performance snap-in appears with a graphical chart displayed in the details pane.
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**. If you need a detailed explanation of the counter, click **Explain**.

When you are finished, click **Close**. The counters that you selected appear in the lower part of the screen. System Monitor displays each counter in a unique color.

Note When 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 where the console file is installed. For more information about creating consoles, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Additional Resources

- For more information about the Windows 2000 Platform SDK, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

CHAPTER 9

Active Directory Backup and Restore

Backing up and restoring data is critical to your computer environment. This chapter discusses procedures to back up and restore data only as they apply to Active Directory™, the directory service included with Microsoft® Windows® 2000.

In This Chapter

Related Information in the Resource Kit

- For more information about backing up data, see "Backup" in the *Microsoft® Windows® 2000 Server Resource Kit Server Operations Guide*.
- For more information about restore, recovery and repair, see "Repair, Recovery, and Restore" in the *Microsoft® Windows® 2000 Server Resource Kit Server Operations Guide*.

Introduction to Active Directory Backup and Restore

This chapter discusses how you can back up and restore Active Directory by using the graphical user interface (GUI) and command-line tools provided in Microsoft® Windows® 2000 Server. The GUI tool, Backup, is used to back up and restore Active Directory (as well as other data and services). The command-line tool, Ntdsutil, is used in conjunction with Backup and gives you finer control over which Active Directory objects you restore.

Active Directory Backup

The Backup tool has several features that make backing up Active Directory a straightforward task that you can integrate into your regular backup procedures without interrupting the network or the operation of the domain controller you are backing up. For example, by using the Windows 2000 Backup tool you can:

- Back up Active Directory while the domain controller is online.
- Back up Active Directory, along with other system and data files.
- Back up Active Directory by using batch file commands.
- Back up Active Directory to any removable media, any available network drive, or a file.

Although Backup provides several methods for backing up data, the only type of Backup supported by Active Directory is *normal backup*. A normal backup creates a backup of the entire system while the domain controller is online. A normal backup marks each file as having been backed up, which clears the Archive attribute of the file. A normal backup also truncates the log files of database applications. To restore a system from a normal backup requires a single restore from the backup media (by comparison, restoring a system from an incremental backup requires all incremental backups as well as the first normal backup).

In addition, when you back up Active Directory, the Backup tool also automatically backs up all of the system components and all of the distributed services upon which Active Directory is dependent. This dependent data, which includes Active Directory, is known collectively as the System State data.

On a Windows 2000 domain controller, the System State data encompasses the system startup files; the system registry; the class registration database of COM+ (an extension to the Component Object Model); File Replication service (the SYSVOL directory); Certificate Services database (if it is installed); Domain Name System (if it is installed); Cluster service (if it is installed); and Active Directory. From a practical standpoint, this means that when you use the Backup tool to back up Active Directory, you cannot back up Active Directory by itself. It is recommended that you schedule and perform regular backups as a normal practice.

Active Directory Restore

There are two methods for restoring replicated data on a domain controller. You can reinstall Windows 2000, reconfigure the domain controller, and then let the normal replication process repopulate the new domain controller with data from its replica partners, or, you can use the Backup tool to restore replicated data from backup media without reinstalling the operating system or reconfiguring the domain controller. In addition, there are two general methods for restoring replicated data from backup media: nonauthoritative and authoritative. Since Active Directory is replicated data, these methods apply to Active Directory restores.

During nonauthoritative restore, the distributed services on a domain controller are restored from backup media and the restored data is then updated through normal replication. In short, each restored directory partition is updated with that of its replication partners. Nonauthoritative restore is typically performed when a domain controller has completely failed due to hardware or software problems.

Authoritative restore occurs after nonauthoritative restore has been performed. During authoritative restore, an entire directory, a subtree, or individual objects can be designated to take precedence over any other instances of those objects on domain controllers. So, through normal replication, the restored domain controller becomes authoritative in relation to its replication partners. Authoritative restore is typically used to restore a system to a previously known state, for example before Active Directory objects were erroneously deleted. The Ntdsutil command-line tool allows you to authoritatively restore the entire directory, a subtree, or individual objects provided they are leaf objects.

Permissions and User Rights

To back up or nonauthoritatively restore Active Directory, you must have the following permissions and user rights:

- To back up the System State data, the person performing the backup procedure must be either a Backup Operator or an Administrator.
- To restore the System State data, the person performing the procedure must be a Local Administrator.

Backup Operator, Administrator, and Local Administrator are Windows 2000 built-in groups. They are already set up and have the necessary permissions and user rights already defined.

Support for Third-Party Backup Tools

The Backup tool can read tapes created by third-party vendor backup tools. This can be useful in situations in which the third-party backup software is not readily available for some reason, such as when a computer is rebuilt and your third-party backup software is not yet installed. Also, third-party tools permit the use of remote tape devices, which are useful when performing enterprise-wide backups. For the Backup tool to read the tape successfully, the software used to create the tape must use the Microsoft tape format (MTF), and it must not compress the data.

For more information about specific procedures for performing backup and restore by using the Backup tool, see Windows 2000 Server Help.

Backing Up Active Directory

This section describes how to back up Active Directory using the Microsoft Windows 2000 backup and restore tool, Backup. Backup is a system tools accessory in the GUI. You can also access Backup at the command prompt by typing **Ntbackup**. This tool is used to back up and restore Active Directory (as well as other services) so that you can restore data or system components in the event of some unforeseen or inadvertent failure. Specifically, the Backup tool allows you to back up and restore the following:

- Entire server
- Selected files
- System State data

As mentioned earlier in this chapter, the System State data includes Active Directory and all other system components and services on which Active Directory is dependent. On a Windows 2000 domain controller, the System State data encompasses the system startup files, system registry, COM+ class registration database, File Replication service (the SYSVOL directory), Certificate Services database (if it is installed), Domain Name System (if it is installed), Cluster service (if it is installed) and Active Directory. The DNS data includes DNS zone information that is Active Directory–integrated. The Cluster service data includes any registry checkpoints and the quorum log, which contains the most recent cluster database information. Active Directory includes the following files:

- *Ntds.dit*. The Active Directory database.
- *Edb.chk*. The checkpoint file.
- *Edb*.log*. The transaction logs; each 10 megabytes (MB) in size.
- *Res1.log* and *Res2.log*. Reserved transaction logs.

Note By default, Active Directory is located in the directory Winnt\Ntds. However, you can designate a different location when you promote a server to a domain controller.

The Backup tool has a Backup wizard that guides you through the backup process, or you can use the GUI to manually complete the process. Procedures for backing up the System State data are described below.

▶ **To back up System State data using the Backup Wizard**

1. From the **Start** menu, click **Run**, and then type **Ntbackup**.
2. On the **Tools** menu, click **Backup Wizard**.
3. 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, click **Next**, and then click **Finish**.
5. When you are done setting options, click **Finish**.

You can also set advanced backup options using the Backup Wizard by clicking **Advanced** on the final wizard screen. This allows you to set or configure several parameters, including: data verification, hardware compression, media labels, whether you want the backup job appended to a previous job, and whether you want to schedule the backup to run unattended at another time. Data verification is particularly useful. If you choose this option, Backup checks to see whether there are differences between the files it backed up from the domain controller and those copied to the backup media. The results of the verification are reported in the Event Viewer. If there are differences in the files, the event type is "Error." Otherwise, the event type is "Information." For more information about errors that might be encountered while creating a backup, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. For more information about how to use the Backup tool, including information about backup options, see the Windows 2000 Server Help.

Important For full disaster recovery, back up all of the drives and the System State data. You can do this by running the **Backup** tool and choosing **Back up everything on my computer** on the **What to Back Up** screen from **Backup Wizard**.

▶ **To back up System State data manually by using the GUI**

1. From the **Start** menu, click **Run**, and then type **Ntbackup**.
2. On the **Backup** tab, under **Click to select the check box for any drive, folder, or file that you want to back up**, click the check box next to **System State**. **System State** appears in the tree view 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. In the **Backup media or file name** box, choose a file name or a tape name that you want to use to save the System State data.
5. Click **Start Backup**, edit any backup job information that you want to, and then click **Start Backup** again.

Note the following when using the Backup tool to back up System State data and other files:

- You must be an Administrator or Backup Operator to back up the System State data.
- System State data does not contain Active Directory unless the server on which you're backing up System State is a domain controller.
- You can back up System State data by itself, or you can back up System State data with other files as part of your regular backup procedures.
- You can back up System State data to a disk, tape, or a network share *while the domain controller is online.*
- If you're backing up to tape, you might have to use Removable Storage to add a tape to the Backup media pool or else the tape will not be available for Backup to use.

Important Because the Backup tool only supports local backups of Active Directory, you must perform a backup on *every* domain controller in the enterprise to entirely back up Active Directory. (Active Directory cannot be backed up on a remote computer.) This is a limitation of the Windows 2000 Backup tool; many third-party backup programs remotely back up and restore Active Directory.

For more information about resolving problems encountered during backup and about using Event Viewer, see "Active Directory Diagnostics, Troubleshooting, and Recovery" in this book.

Restoring Active Directory

There are two ways to restore Active Directory. You can reinstall Windows 2000, and then let normal replication repopulate Active Directory through the normal replication process, or you can restore Active Directory from a backup. The first method restores Active Directory to the current state with respect to its current replica partners. The second method restores Active Directory to a previously known state (whatever it was when you last backed up).

Restoring Active Directory Through Reinstallation and Replication

You can restore a domain controller by reinstalling Windows 2000 Server on the damaged system, making it a domain controller, and allowing the correct information to be copied to it automatically by Active Directory. To do this, first use the Sites and Services snap-in on an existing domain controller to delete any references to the old domain controller.

To open the Sites and Services snap-in, from the **Start** menu, point to **Programs** and **Administrative Tools**, and then click **Active Directory Sites and Services**.

Then, on the damaged system, reinstall Windows 2000 Server. On that server, use the Active Directory Installation Wizard to reinstall Active Directory, and promote the server to be a domain controller. Active Directory and SYSVOL are brought up-to-date through replication from a domain controller.

For more information about the Active Directory Installation Wizard, see Windows 2000 Help. The Active Directory support tools are included in the Microsoft® Windows® 2000 Support folder on the Windows® 2000 operating CD.

Restoring Active Directory from Backup Media

You can also restore Active Directory information on a domain controller by restoring the System State data from backup media. This restores Active Directory as well as the other System State components on which Active Directory depends. There are two general methods for restoring Active Directory from backup media: nonauthoritative and authoritative.

Nonauthoritative restore means that when a domain controller is restored from backup media, the restored data is updated through normal replication. Each restored directory partition is updated with that of its replication partners. One example of where you use nonauthoritative restore is a hard disk failure that requires replacement of the primary hard disk on a domain controller. In this case, you must format the new disk, recreate the partitions as they were before the crash, reinstall Windows 2000 Server on the primary partition, restore whatever data files you had on the computer, and then restore your distributed services, including the entire Active Directory. Because you use the Backup tool to restore the distributed services (including Active Directory), the restore operation is nonauthoritative. Therefore, any replicated data on the computer you restored, such as Active Directory data, is automatically updated through replication with your other domain controllers. The Backup tool allows you to nonauthoritatively restore Active Directory.

Authoritative restore means that *after* Active Directory is restored nonauthoritatively from a backup medium that the entire directory, a subtree, or individual objects can be designated to take precedence over any other instances of those objects on replicated domain controllers. So, through normal replication, the restored domain controller becomes authoritative in relation to its replication partners. Even if the authoritatively restored replica set is older than the current replicas, the *older* data is replicated to all of its replication partners. Authoritative restore is typically used to restore a system to a previously known state, for example before objects were erroneously deleted. The Ntdsutil command-line tool allows you to authoritatively restore the entire directory, a subtree, or individual objects provided they are leaf objects.

When you restore Active Directory from backup media, note the following:

- If the domain controller computer has been replaced because of malfunction, or if the network adapters have been replaced, you might need to reconfigure the network settings manually.
- If the domain controller has a substantial hardware problem and must be rebuilt, make sure the number and size of disk volumes is the same or larger than the previous system. If you must rebuild a system by starting with an empty hard disk, first install Windows 2000 Server (on the same disk as before), then recreate the partitions and volumes as they were on the damaged system, and finally restore the Active Directory information.

Nonauthoritatively Restoring Active Directory by Using the Backup Tool

By default, the Backup tool operates in nonauthoritative restore mode, which is to say, when you restore data by using the Backup tool *and no other tool*, you are restoring data nonauthoritatively. When the domain controller is brought online after a nonauthoritative restore, it detects that the restored data hasn't been updated since the backup was performed, and then it begins receiving and applying updates through normal replication with its replication partners. Therefore, any directory updates that occurred after the backup was created are applied after restore as part of the normal replication process. Replication reconstructs the replication metadata for the updates that originated on the restored domain controller between the time the server was last backed up and the time at which it is restored from backup. For more information about how metadata is reconstructed, see "Active Directory Replication" in this book.

Using the Backup Tool to Restore Active Directory

You can restore a domain controller from backup media only while Active Directory is offline. You can take Active Directory offline by placing the domain controller in Directory Services Restore Mode.

▶ **To put a domain controller into Directory Services Restore Mode**

1. Restart the domain controller.
2. When you are asked to choose which operating system you want to start, press F8.
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 an Administrator. To do this you must use an account name and password that is stored in the local security account database known as the Security Accounts Manager (SAM). You *cannot* use the name and password of the Active Directory administrator. This is because Active Directory is offline, and account verification cannot occur. Rather, the SAM accounts database is used to control access to Active Directory on the local computer while Active Directory is offline.

After taking the domain controller offline, you can use the Restore Wizard in the Backup tool to restore the System State data, or you can use the graphical user interface to manually restore the System State data. The procedure for restoring System State data by using the wizard is described below. For information about restoring System State data manually using the GUI, see Windows 2000 Server Help.

▶ **To restore System State using the Restore Wizard**

1. From the **Start** menu, click **Run**, and then type **Ntbackup**.
2. On the **Tools** menu, click **Restore Wizard**.
3. Click **Next**, select the backup set from which you want to restore, select **System State,** and then click **Next**.
4. Click **Finish**.

By default, the System State data is restored to the system root. Furthermore, by default, the System State data from the backup media replaces existing System State data on the domain controller. By setting advanced restore options you can change this default behavior for some of the System State components, but not for Active Directory. To change the default restore location, click **Advanced** on the last screen of the Restore Wizard and then choose where you want the System State data restored. If you choose an alternate location for restoring the System State data, only the system boot files, registry files, SYSVOL directory files, and Cluster service database information is restored to the alternate location. Active Directory, the COM+ class registration database, and the Certificates Services files are not restored.

Important 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.

You cannot restore into a replicated enterprise a backup image that is older than the *tombstone lifetime* setting for the enterprise. Tombstone lifetime is the number of days that a deleted object is maintained before the garbage collection process permanently removes it from Active Directory. The tombstoneLifeTime attribute is set on the object cn=Directory Services, cn=WindowsNT, cn=Services, cn=Configuration, dc=DomainName. The default value is 60 days, which can be adjusted. For more information about how to set the tombstone lifetime, see "Active Directory Data Storage" in this book. If your only backup is older than that, you need to reinstall the server from the Windows 2000 Server operating system CD. For more information about the tombstoneLifeTime attribute see, "Active Directory Data Storage" in this book.

If a domain controller was restored from a backup that was older than the tombstone lifetime, then the domain controller might contain deleted objects, and because the tombstones are deleted from the replica, the deletion event does not replicate into the restored domain controller. This is why Backup does not allow you to restore data from a backup that is older than the tombstone lifetime. The tombstone lifetime is the length of time that an object lives as a tombstone in the directory before being collected as garbage. The default is 60 days.

For more information about specific procedures for restoring from a backup by using the Backup tool, see Windows 2000 Server Help.

Note If you backed up to an NTFS volume on a network share, it is recommended that you restore the data to a disk volume that is formatted for the version of NTFS used in Windows 2000, or you might lose data as well as some file and folder features. For example, permissions, Encrypting File System (EFS) settings, disk quota information, mounted drive information, and Remote Storage information are lost if you restore data backed up from such a disk volume to a disk volume that is formatted for FAT or for the version of NTFS used in Microsoft® Windows NT® version 4.0.

Because the Backup tool restores the database and registry settings, when it restores Active Directory, the Internet Protocol (IP) configuration is also restored. Additionally, the DNS, the certificate server database files, and File Replication service (FRS) are also restored. Completing restore has the following results:

- The File Replication service is reset so that it is in a state ready for replication from a FRS replication partner.
- The Active Directory database is verified for restore.

The server then restarts into normal operational mode and performs the following actions:

- Checks Active Directory database files for consistency and re-indexes them.
- Replicates FRS data with an FRS replication partner. (For more information about FRS, see "File Replication Service" in this book.)
- Restores the Certificate Services database. (For more information about Certificate Services, see "Windows 2000 Certificate Services and Public Key Infrastructure" in this book.)

Distributed Services Dependencies

Active Directory is interlaced with other distributed services, such as Certificate Services, FRS, Distributed file system (Dfs), system registry, and so on. Because of these dependencies that distributed services have on one another, restoring Active Directory is not an isolated operation. For example, Active Directory relies on Policies for associating policies with domains and organizational units. Unless all dependent services are restored in the same mode and from the same backup media, inconsistencies might result.

Note If you are only backing up a file server that is not replicated throughout the forest, then these dependencies do not exist.

Implications of a Nonauthoritative Restore

When you use the Backup tool to restore a Windows 2000 domain controller, it is considered nonauthoritative. This means that when the restored domain controller replicates with its replication partners in the enterprise, the restored directory partitions are updated through replication with another domain controller within the restored domain.

So, when a domain controller is restored from a backup medium, any directory updates that occurred after the backup are applied as part of the normal replication process. Replication reconstructs the replication metadata for the updates that originated on the restored domain controller between the time the server was last backed up and the time it gets restored. For more information about how metadata is reconstructed, see "Active Directory Replication" in this book.

When directory partitions are restored from a backup using the Backup tool, the domain controller might lose information about the directory updates it originated after it was backed up. For example, suppose you back up your domain controller on day 1. On day 2 you add some new objects or make modifications to Active Directory, but the new objects or modifications are not replicated to your other domain controllers because of a network problem. On day 3, a catastrophic event (unrelated to the modifications you made on day 2) forces you to restore the domain controller from your backup. In this case, the new objects or modifications that originated on the domain controller after the backup are lost because they were never replicated to your other domain controllers and, therefore, can't be applied to the restored domain controller.

If the domain controller begins originating new updates *before* it receives updates from its replication partners through replication, one of two things happens:

- The domain controller declares itself fully up-to-date with respect to its own changes. In this case, it never receives the remaining updates it originated after the backup.
- The domain controller does not declare itself fully up-to-date with respect to its own changes. In this case, any new updates it originates replicates back to itself.

Problems can happen when the domain controller tries to determine where to restart in the update sequence number (USN) sequence. If it reuses an existing USN, serious replication problems can result. Both problems arise because the restored domain controller does not have global knowledge, it only has knowledge of itself and its current replication sources.

The solution is to generate a new database globally unique identifier (GUID) for the domain controller. This happens automatically with the restore operation. The restored domain controller assumes a new GUID and is aware of the USN for its own changes, with respect to its previous identity at the time of the last backup. The domain controller saves this state information so that replication partners do not subsequently send changes originated locally before the backup.

The restored domain controller receives the changes it originated after the backup (and before the restore) through normal replication, just as it would receive changes made by any other domain controller. Other domain controllers similarly receive changes made by the restored domain controller just as they would receive changes from a new replica.

For more information about replication, see "Active Directory Replication" in this book.

Verifying the Nonauthoritative Restore

Verify the success of the restore process by checking that Active Directory, Certificate Services, and File Replication service are operational. Perform a basic check of Active Directory by browsing users, groups, and other objects that were present before the backup. You can perform an advanced verification of Active Directory by performing the following procedure:

Important You can only perform this procedure 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 tool restore**

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. Verify that Active Directory is in a state consistent with having been recovered from backup by checking for a specific registry entry. Use a registry editor by typing **regedit** or **regedt32** at the command prompt.
4. Locate the **RestoreInProgress** entry in the following subkey:
   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS

   This entry is automatically generated by the Backup tool when restore is successful. It notifies Active Directory to perform a consistency check and to re-index database files when Active Directory is restarted. Then, the **RestoreInProgress** entry is automatically deleted when the process is finished. *Do not add, delete, or change the value of* ***RestoreInProgress.***

   Note After the database is restored from backup media, it is not in a valid format. To cause Active Directory to make the database usable, the Backup tool adds **RestoreInProgress** to the NTDS registry subkey. Active Directory reads this entry during system initialization and then deletes the entry.

   ---

5. Close the registry editor, and run the Ntdsutil tool. Type **Files** to display the **Files** menu, and then type **Info** to display the **Information** menu. If the Active Directory database files are successfully recovered, Ntdsutil displays that information. (For more information about Ntdsutil, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.)
6. Restart the domain controller in normal mode.

When the computer is restarted in normal mode, Active Directory automatically detects the recovery, performs an integrity check, and re-indexes the database. Browse the directory, and then make sure that all of the objects that were present in the directory prior to backup are restored.

Caution Do not use a registry editor to edit the registry directly unless you have no alternative. The registry editors bypass the standard safeguards provided by administrative tools. These safeguards prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your system. Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. To configure or customize Windows 2000, use the programs in Control Panel or Microsoft Management Console (MMC) whenever possible.

Restoring Active Directory to Dissimilar Hardware

It is possible to restore Active Directory to a computer other than the original computer, both computers must have the same number of disk drives. Also, if the replacement domain controller has a different video adapter or multiple network adapters, uninstall them before you restore data. When you restart the computer; Plug and Play functionality makes the appropriate updates.

Authoritative Restore

An authoritative restore means that *after* Active Directory is restored nonauthoritatively from a backup, the entire directory, a subtree, or individual objects can be designated to take precedence over any other instances of those objects in the forest. So, through normal replication, the restored domain controller becomes authoritative in relation to its replication partners. Even if the authoritatively restored replica set is older than the current replicas, the *older* data is replicated to all of its replication partners.

You perform authoritative restore after you perform nonauthoritative restore using the Backup tool. After restoring from backup with the Backup tool, you use the Ntdsutil tool to mark restored objects as authoritative so that they are propagated by the process of replication, thereby updating existing copies of those objects throughout the forest.

Important 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. For example, if the schema was modified, and then objects of the new or modified classSchema object were created, subsequent authoritative restore might replace the new or modified classes causing serious data consistency problems.

Authoritative restore is often used in situations in which objects are inadvertently deleted from Active Directory, and the deletions are propagated to other domain controllers. To recover from such an event, you can make authoritative restore to Active Directory from a backup that was created before the objects were deleted. After the domain controller is restored, but before the domain controller is restarted, the previously deleted objects are designated as authoritative. When you bring the domain controller online, the authoritative objects are replicated to the other domain controllers through the normal replication process. The previously deleted objects are ignored during replication (that is, they are not replicated), because the authoritatively restored objects have a higher version number, which overrides the lower version numbers of previously deleted objects.

It is important to note that authoritative restore does not affect objects that were created after the backup from which you are restoring was created. When other domain controllers exist, and authoritative restore is made, any objects that were created in a directory partition after the backup remain in Active Directory. For example, if you back up the system on Monday, and then create a new user account called James Smith on Tuesday, which replicates to other domain controllers in the domain. On Wednesday, another user account, Amy Anderson, is accidentally deleted. To authoritatively restore Amy Anderson without re-entering information, you can restore the domain controller with the backup created on Monday. Then, using Ntdsutil, mark the Amy Anderson user account as authoritative. The result is that Amy Anderson is restored, without any effect on James Smith.

The authoritative restore feature of the Ntdsutil tool is meant to be used sparingly because it restores the directory to an earlier state and any updates that were made after that point are lost. You can use it to selectively modify individual objects, subtrees, organizational units, and even an entire forest, but do so only if you have identified a specific problem and you know restore can fix it.

Note Certificate Services is not affected by authoritative and nonauthoritative restore because the data is not replicated across distributed systems. File Replication service (FRS) *is* affected by authoritative and nonauthoritative restore. However, the Ntdsutil command-line tool is *not* used to authoritatively restore FRS. To restore FRS authoritatively, restore the FRS files to an alternate location, restart the domain controller, publish the SYSVOL share, and then copy the restored files from their alternate location to their original location. You must publish the SYSVOL share before you copy the files or the restore does not succeed and policies are broken.

Authoritatively Restoring Active Directory by Using Ntdsutil

To perform authoritative restore of Active Directory, back up and restore the domain controller by using the Backup tool. (For more information about Backup, see "Backing Up Active Directory" earlier in this chapter.) Then after the restore operation finishes—but before reconnecting to the network—carry out the Ntdsutil commands for authoritative restore.

▶ **To authoritatively restore Active Directory**

1. Perform nonauthoritative restore of Active Directory, and then restart the computer. During the phase of startup where the operating system is normally selected, press **F8** to display advanced startup options. In the Windows 2000 **Advanced Options** menu, select **Directory Services Restore Mode**. This ensures that the domain controller is offline and is not connected to the network.
2. At the command prompt, type **ntdsutil**, and then press ENTER. Type **authoritative restore**, and then press ENTER.
    1. To authoritatively restore the entire directory, type **Restore database** and press ENTER.
    2. To authoritatively restore a portion, or subtree of the directory, such as a grouping of organizational units (OUs), type **Restore subtree** <*subtree distinguished name*>. Press ENTER.

       For example, to restore the Marketing OU in the Reskit.com domain, the commands are as follows:

       ntdsutil

       authoritative restore

       Restore Subtree OU=Marketing,DC=Reskit,DC=COM
    3. To authoritatively restore the entire directory *and* override the version increase, type **Restore database verinc** <*version increase*>, and then press ENTER.
    4. To authoritatively restore a subtree of the directory *and* override the version increase, type **Restore subtree** <*subtree distinguished name*>**verinc** <*version increase*>, and then press ENTER.

    During authoritative restore, 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.
3. Type **quit**, and then press ENTER to exit the Ntdsutil tool.
4. Restart the domain controller in normal mode and connect the restored domain controller to the network.

When the restored domain controller is online and connected to the network, normal replication brings the restored domain controller up-to-date with any changes from the additional domain controllers that were not overridden by authoritative restore. Replication also propagates the authoritatively restored objects to other domain controllers in the forest. The deleted objects that were marked as authoritative are replicated from the restored domain controller to the additional domain controllers.

Because the objects that are restored have the same objectGUID and objectSID, security remains intact, and object dependencies are maintained.

For more information about using the Ntdsutil tool, see "Active Directory Diagnostics, Troubleshooting, and Recovery" in this book.

Note The process mentioned above is primarily for authoritative restoration of only Active Directory. However, certain Active Directory objects (for example, OUs, domains and Site objects) may have group policies associated with them, and these group policies are stored in the SYSVOL directory. To authoritatively restore the Active Directory and SYSVOL, see "Authoritative Restoration of the Entire Active Directory Database" later in this chapter.

Authoritative Restoration of 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 this by copying the SYSVOL directory from the alternate location over the existing SYSVOL directory. This is necessary to ensure the integrity of the Group Policy of the computer. Use the following procedure to restore an entire Active Directory database.

▶ To restore the entire Active Directory database

1. Back up the System State data by using the Backup tool.
2. Restart the computer in Directory Service Restore Mode.
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. You can verify that the copy is complete by checking the contents of the Sysvol\domain directory; when completed, it contains a Scripts and Policies folder.

Authoritative Restoration of Specific Active Directory Objects

When you make authoritative restore of a portion of the Active Directory database (including Policy objects), you also must perform an additional procedure (described below) 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 tool.
2. Restart the computer in Directory Service 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.

Important When authoritatively restoring either the entire Active Directory database or selected objects, it is important that you copy the SYSVOL and policy data from the alternate location, that this be done *after* the SYSVOL share is published.

If the computer is in a replicated domain, it can take several minutes before the SYSVOL share is published, because it needs to synchronize with its replication partners.

If all computers in the domain are authoritatively restored and restarted at the same time, then they each are waiting (indefinitely) to synchronize with each other. In this case, restore one of the domain controllers first, so that its SYSVOL share can be published. Then, restore the other computers nonauthoritatively.

Verifying the Authoritative Restore

Authoritative restore operation restores the attributes of existing objects or restores deleted objects. It sets every attribute involved to its current value. However, the attribute values themselves don't change, but their metadata does. The metadata is changed to indicate when and where each attribute involved was given its current value. During replication, other domain controllers see this as an update, which overrides current values. Each domain controller might have a different notion of the "current value." For the authoritatively restored domain controller, the current value is as of the time of the backup. For other domain controllers, the current value is as it exists after all changes are made after the backup.

You can use the Repadmin command-line tool to verify that the authoritative restore was successful by checking the version number increase on the directory or subtree. Do this by carrying out the **show metadata** command followed by the exact distinguished name of the directory or subtree that you authoritatively restored. For more information about using the Repadmin tool, see "Active Directory Diagnostics, Troubleshooting, and Recovery" in this book.

Performing an authoritative restore updates the following metadata variables as described in Table 9.1.

Table 9.1 Updated Metadata Variables in Authoritative Restore

Per Domain Controller	**Result**
Highest-Committed-USN attribute	Incremented by one for each object that was authoritatively restored because unique sequence numbers (USNs) are machine specific. That means they are compared only against other USNs that were issued by the current domain controller.
Per Object and Per Domain Controller	**Result**
USN-Changed attribute	Set to the current value of *Highest-Committed-USN* attribute. Two domain controllers typically have different USN-Changed attributes for the same object, even if they are synchronized.
Per Object	**Result**
When-Changed attribute	Set to the current time.
Per Attribute	**Result**
Originating-DC-GUID attribute	Set to the GUID of the current domain controller.
Originating-USN attribute	Set to the current value of *Highest-Committed-USN* attribute.
Version attribute	Set to a version number higher than the version number of any other copy of the object on any other domain controller. (See discussion later in this chapter.)
When-Changed attribute	Set to the current time.
Property-USN attribute	Set to the current value of *Originating-USN* attribute.
Is-Deleted attribute	A new metadata entry is made.

Updates made by the authoritative restore appear as any other update, and replicate like other updates, with one exception. The exception with authoritative restore updates is that the version number is set high enough so that the updates made by authoritative restore are higher in version number than the normal updates.

Further, the version number increases by one hundred thousand for each day after the original backup update. (You supply the number of days.) You can override the version number.

Impact of Authoritative Restore on Trust Relationships and Network Connections

Both parent and child trust relationships in Windows 2000 domains and Kerberos and NTLM trust relationships to other Windows NT 4.0 or Windows 2000 domains reside in the domain directory partition. Because trust relationship and computer account passwords are renegotiated at a specified interval, 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 values are different from the current values, trust relationships and computer accounts might be invalidated. In the case of trust relationships, this might negate communication with domain controllers from other domains. If an older computer account password is restored, it could negate communications between the member's workstation or server and the domain controller. If the objects that you are authoritatively restoring affect trust relationships or computer account passwords, you need to reset the passwords. Therefore, be very selective when choosing objects to authoritatively restore; restore only those portions of the domain directory partition that are absolutely necessary.

Note By default, passwords are reset every seven days; except for computer accounts. The previous password is also maintained. Therefore, performing authoritative restore with a backup that is older than 14 days can affect the trust relationships.

The extent to which trust relationships are affected by authoritative restore depends upon the scope of the restore—the more of the domain hierarchy included in the restore, the greater that trust relationships are affected. To minimize the effort involved with resetting trusts and rejoining computers, you need to perform regular backups. Yet, there are always some trusts, user, and computer accounts that require resetting.

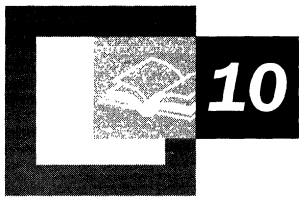
To reset Windows 2000 trust relationship passwords, use the Active Directory Domains and Trusts snap-in. For more information about specific procedures for resetting passwords, see Windows 2000 Server Help. To reset computer account passwords use the Active Directory Users and Computers snap-in. You can also use the Netdom command-line tool to reset trust relationship and computer account passwords. For more information about using the Netdom tool, see *Windows 2000 Resource Kit* Tools Help, which is included on the *Windows 2000 Resource Kit* companion CD, or see "Active Directory Diagnostics, Troubleshooting, and Recovery" in this book.

For more information about parent/child trust relationships, see "Active Directory Logical Structure" in this book.

Additional Resources

- For more information about the application programming interfaces (APIs) related to backing up and restoring data on Windows 2000, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.
- For more information about recovery and repair, see the the *Windows 2000 Professional Getting Started* book that came with your Windows 2000 CD.
- For more information about backup and restoration of data, see Windows 2000 Server Help.

CHAPTER 10



Active Directory Diagnostics, Troubleshooting, and Recovery

Diagnosing and troubleshooting Active Directory™, the directory service that is included with Microsoft® Windows® 2000, requires thorough familiarity with the content of the other Active Directory chapters in this book and knowledge of and proficiency in the use of the diagnostic tools that are included on the *Microsoft® Windows® 2000 Server Resource Kit* companion CD and Microsoft® Windows® 2000 Server operating system CD. Because Active Directory interacts with external services and protocols, such as DNS for name resolution, LDAP for directory access protocols, and TCP/IP for the transport protocol, it becomes more complex to accurately determine the cause of a problem and to apply a solution. Improper configuration of the services and protocols can create problems such as not being able to locate resources.This chapter assumes that you have already read and are thoroughly familiar with the content in the other Active Directory chapters of this book.

In This Chapter

Related Information in the Resource Kit

- For more information about diagnostics and troubleshooting, see the *Microsoft® Windows® 2000 Server Resource Kit Server Operations Guide*.
- For more information about troubleshooting software installation and maintenance, see "Troubleshooting Change and Configuration" in this book, and see the ResourceLink link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Summary of Active Directory Architecture

To determine the cause of an Active Directory problem accurately, you need to understand its architecture and the relationship of Active Directory to other network services and protocols. Active Directory is a database that not only stores information and resources but also extends the features of previous Microsoft® Windows–based directory services and adds new features. These new features make it easier to navigate and manage large amounts of information, which can generate savings for both administrators and end users. However, these features also increase the depth and complexity of the underlying architecture. So, to establish a starting point for Active Directory diagnostics and troubleshooting, it might be useful to briefly review how the system is structured., The following is a summary of Active Directory architecture and the protocols and services that interact with Active Directory. For more information about Active Directory architecture, see "Active Directory Data Storage" in this book.

Protocols

The main protocols that are used by Active Directory are Domain Name System (DNS), Transfer Control Protocol/Internet Protocol (TCP/IP), and Lightweight Directory Access Protocol (LDAP).

DNS

Domain Name System (DNS) is the de facto naming system for Internet Protocol (IP)-based networks and the naming service that is used to locate computers on the Internet. Windows 2000 uses DNS to locate computers and domain controllers (that is, to locate Active Directory). A workstation or member server finds a domain controller by querying DNS. For this reason, installing or upgrading to Windows® 2000 Server requires that a DNS infrastructure is in place or is installed simultaneously.

Every Windows 2000 domain has a DNS name (for example, reskit.com), and every Windows 2000–based computer has a DNS name (for example, Server1.reskit.com). Thus, domains and computers are represented both as objects in Active Directory and as nodes in DNS. For more information about DNS, see the *Microsoft® Windows® 2000 Server Resource Kit TCP/IP Core Networking Guide*.

TCP/IP

The required transport protocol for Active Directory is TCP/IP. For more information about TCP/IP, see the *TCP/IP Core Networking Guide*.

LDAP

LDAP is a structured protocol that is used to view and manipulate information that is stored in a hierarchical database. LDAP is defined by Request for Comments (RFC) 2251: "Lightweight Directory Access Protocol." Clients use LDAP for reading and updating the contents of Active Directory. Active Directory supports both LDAP version 2 (LDAP v2) and LDAP version 3 (LDAP v3).

The general model adopted by this protocol is one of clients performing protocol operations against servers. In this model, a client transmits a protocol request describing the operation to be performed to a server. The server is then responsible for performing the necessary operations in the directory. Upon completion of the operations, the server returns a response containing any results or errors to the requesting client.

Note Servers are required to return responses, but whenever such responses are defined in the protocol, there is no requirement for synchronous behavior on the part of either clients or servers. Requests and responses for multiple operations can be exchanged between a client and server in any order, provided the client eventually receives a response for every request that required one.

The ability to search a directory encompasses several operations that can be performed by a client. These include search, connect, bind, modify, add, and delete. Although it might be important for an administrator to be able to manipulate the information in Active Directory, greatest benefit to the end user is the ability to view information. The user, for example, might want to look up the telephone extension or room number of a coworker.

For more information about LDAP v3, see the Request For Comments link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources . Follow the links to RFC 2251.

Directory System Agent

The *directory system agent* (DSA) is the process that provides access to the *store*. The store is the physical store of directory information that is located on a hard disk. In Active Directory, the DSA is part of the Local System Authority (LSA) process in Windows 2000. The DSA manages the directory; therefore, it understands what each directory object and function represents. For example, when you create objects, the DSA knows how to check the Active Directory schema to identify the mandatory and optional attributes for that particular object.

The DSA also manages other relationships, such as replication topology, by identifying when events are going to force replication synchronization. Therefore, it implements the directory service itself. Clients gain access to the directory by using one of the following mechanisms supported by the DSA:

- LDAP clients connect to the DSA by using the LDAP protocol. Windows 2000–based clients, and Microsoft® Windows® 98–based or Microsoft® Windows® 95–based clients with the Active Directory client components installed, use LDAPv3 to connect to the DSA.
- Messaging application programming interface (MAPI) clients, such as Microsoft® Exchange Server version 5.5, connect to the DSA by using the MAPI remote procedure call (RPC) interface.
- Windows clients that use Microsoft® Windows NT® version 4.0 or earlier connect to the DSA by using the Security Accounts Manager (SAM) interface.
- Active Directory domain controllers connect to each other to perform replication by using a proprietary RPC implementation.

For more information about the DSA, see "Active Directory Data Storage" in this book.

Database Layer

The *database layer* provides an object view of database information by applying schema semantics to database records, thereby isolating the upper layers of the directory service from the underlying database system. The database layer is an internal interface that is not exposed to users. No database access calls are made directly to the Extensible Storage Engine; instead, all database access is routed through the database layer.

Active Directory provides a hierarchical namespace. Each object is uniquely identified in the database by its individual naming attribute, called the *relative distinguished name* (also known as the RDN). The relative distinguished name and the chain of successive parent object names make up the object's *distinguished name* (also known as the DN). The database stores the relative distinguished name for each object, as well as a reference to the parent object. The database layer follows these parent references and concatenates the successive relative distinguished names to form distinguished names.

Note Active Directory relative distinguished names are unique within a particular parent; that is, Active Directory does not permit two objects with the same relative distinguished name under the same parent container. The distinguished name identifies one object only and is unique (that is, no other object in the directory has its name).

A major function of the database layer is to translate each distinguished name into an integer structure called the *distinguished name tag*, which is used for all internal accesses. The database layer guarantees the uniqueness of the distinguished name tag for each database record.

All data that describes an object is held as a set of attributes, which are stored as columns in the database. The database layer is responsible for the creation, retrieval, and deletion of individual records, attributes within records, and values within attributes. To carry out these functions, the database layer uses the schema cache (an in-memory structure in the DSA) to get information about the attributes that it needs.

For more information about the schema cache, see "Active Directory Schema" in this book. For more information about distinguished names and relative distinguished names, see "Active Directory Logical Structure" in this book.

Extensible Storage Engine

Active Directory is implemented on top of an Indexed Sequential Access Method (ISAM) table manager, historically called "Jet." This same table manager is used by Exchange, File Replication service (FRS), the security configuration editor, Certificate Services, Windows Internet Name Service (WINS), and various other Windows components. In Windows 2000, this table manager is referred to as the Extensible Storage Engine (ESE).

The ESE (Esent.dll) database uses a concept of discrete transactions and log files to ensure the integrity of Active Directory. Each request to the DSA to add, modify, or delete an object or attribute is treated as an individual transaction. As these transactions occur on each domain controller, they are recorded in a series of log files that are associated with each Ntds.dit file. By default, the Active Directory database file is stored on *<drive>*\winnt\NTDS\Ntds.dit. Likewise by default, the log files are stored in the same directory.

For more information about ESE, see "Active Directory Data Storage" in this book. For more information about FRS, see "File Replication Service" in this book.

Domain Controller Locator

When an application requests access to Active Directory, an Active Directory server (domain controller) is located by a mechanism called the *domain controller locator (Locator)*. Locator is an algorithm that runs in the context of the Net Logon service. Locator can find domain controllers by using DNS names (for IP or DNS-compatible computers) or by using Network Basic Input/Output System (NetBIOS) names (for computers that are running Microsoft® Windows® 3.*x*, Microsoft® Windows® for Workgroups, Microsoft® Windows NT® version 3.5 or later, Windows 95, or Windows 98), or it can be used on a network where IP transport is not available.

The following sequence describes how the Locator is able to find a domain controller:

1. On the client (the computer locating the domain controller), the Locator is initiated as an RPC to the local Net Logon service. The Locator application programming interface (API) (DsGetDcName) is implemented by the Net Logon service.
2. The client collects the information that is needed to select a domain controller and passes the information to the Net Logon service by using the DsGetDcName API.
3. The Net Logon service on the client uses the collected information to look up a domain controller for the specified domain in one of two ways:
   - For a DNS name, Net Logon queries DNS by using the IP/DNS-compatible Locator—that is, DsGetDcName calls the DnsQuery API to read the Service Resource (SRV) records and A records from DNS, after it appends an appropriate string to the front of the domain name that specifies the SRV record.

     A workstation that is logging on to a Windows 2000 domain queries DNS for SRV records in the general form:

     _service._protocol.DnsDomainName

     Active Directory servers offer the LDAP service over the TCP protocol; therefore, clients find an LDAP server by querying DNS for a record of the form:

     _ldap._tcp.DnsDomainName

- For a NetBIOS name, Net Logon performs domain controller discovery by using the Microsoft® Windows NT® version 4.0–compatible Locator, that is, by using the transport-specific mechanism (for example, WINS).

Note In Windows NT 4.0 and earlier, "discovery" is a process for locating a domain controller for authentication in either the primary domain or a trusted domain.

---

4. The Net Logon service sends a datagram to (that is, pings) the computers that registered the name. For NetBIOS domain names, the datagram is implemented as a mailslot message. For DNS domain names, the datagram is implemented as an LDAP User Datagram Protocol (UDP) search. (UDP is the connectionless datagram transport protocol that is part of the TCP/IP protocol suite. TCP is a connection-oriented transport protocol.)

Note UDP allows an application on one computer to send a datagram to an application on another computer. UDP includes a protocol port number, which allows the sender to distinguish among multiple destinations (applications) on the remote computer.

---

5. Each available domain controller responds to the datagram to indicate that it is currently operational and returns the information to DsGetDcName.
6. The Net Logon service returns the information to the client from the domain controller that responds first.
7. The Net Logon service caches the domain controller information so that subsequent requests need not repeat the discovery process. Caching this information encourages consistent use of the same domain controller and, thus, a consistent view of Active Directory.

Note The debug log for the Net Logon service can be enabled by carrying out **nltest\dbflag:0x2000ffff** at the command prompt. Restart the computer, and then review entries in the [INIT] category of the Net Logon.log file that is located in the *%systemroot%*\Debug folder. Net Logon still uses the event log to notify administrators of "well known" problems that might occur, and it is recommended that you look in this place first.

In general to enable logging, it is not necessary to restart the computer. Setting the dbflag automatically enables logging. The restart is for purposes of viewing the [INIT] category of the Net Logon.log file.

Figure 10.1 illustrates the process of a client locating a domain controller.

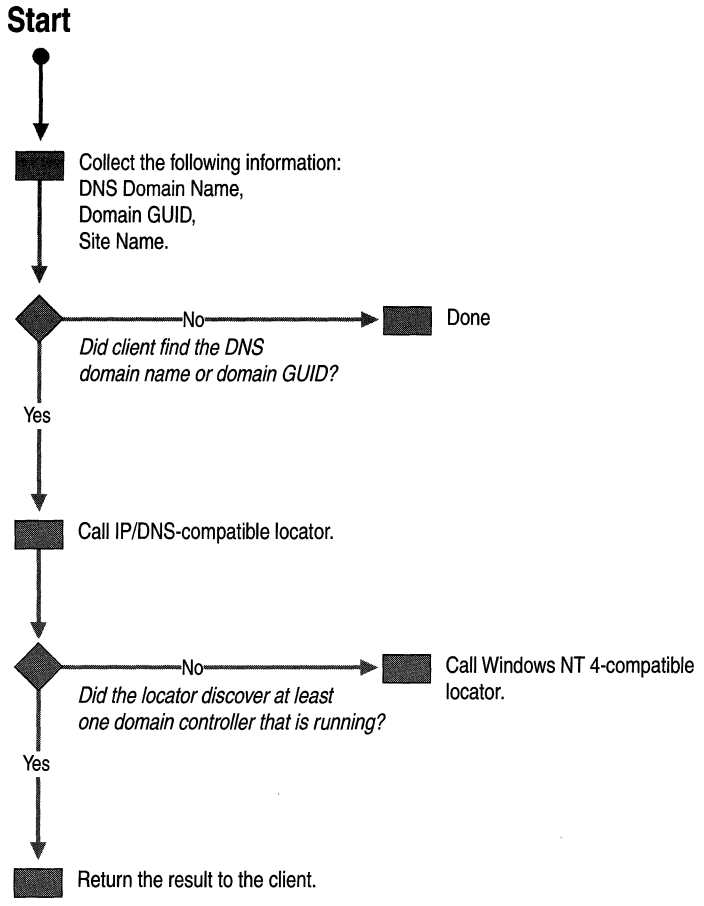


Figure 10.1 Domain Controller Locator Process

Note The locator can also be called by using a NetBIOS domain name, in which case it flows down to the Windows NT 4.0–compatible locator.

When a client logs on or joins to the network, it must be able to locate a domain controller. The client sends a DNS Lookup query to DNS to find domain controllers in the subnet of the client. Therefore, DNS finds the closest domain controller in its subnet.

After the client locates a domain controller, it establishes communication by using LDAP to gain access to Active Directory. As part of that negotiation, the domain controller identifies which site the client is in on the basis of the IP subnet of that client. If the client is communicating with a direct domain controller that is not in the closest (most optimal) site, it then receives the name of the site in which the client is located with a bit that indicates whether the current domain controller is in the closest site. If the client has already tried to find domain controllers in that site (for example, when the client sends a DNS Lookup query to DNS to find domain controllers in the client's subnet), the client uses the domain controller that isn't optimal. Otherwise, the client again does a site-specific DNS lookup with the new optimal site name. The domain controller uses some of the DSA information for identifying sites and subnets.

Note After the client locates a domain controller, the domain controller entry is cached. If the domain controller is not in the optimal site, the client flushes the cache after fifteen minutes, and discards the cache entry. It then attempts to find an optimal domain controller in the same site as the client.

After the client has established a communications path to the domain controller, it can establish the logon and authentication credentials and, if necessary for Windows 2000 platforms, set up a secure channel. Then the client is ready to perform the normal queries and search for information against the directory.

The client establishes an LDAP connection to a domain controller to log on. The logon process goes through the Security Accounts Manager. As the communications path goes through the LDAP interface and the client is authenticated through the DSA, the client account is verified and passed through the Security Accounts Manager to the DSA, the database layer, and, finally, to the database in the ESE. Therefore, there are a number of different component interactions. To effectively troubleshoot your system, you must be able to identify and diagnose problems that might occur in any of these different interactions.

For more information about Locator, see "Name Resolution in Active Directory" in this book.

Diagnosing and Troubleshooting Active Directory Problems

In terms of identifying, analyzing the cause of, and repairing Active Directory problems, there is a specific sequence of events to follow. This sequence serves as a roadmap to help you to accurately identify a situation, diagnose it, and then resolve it. Figure 10.2 illustrates the sequence of events to follow when troubleshooting Active Directory.

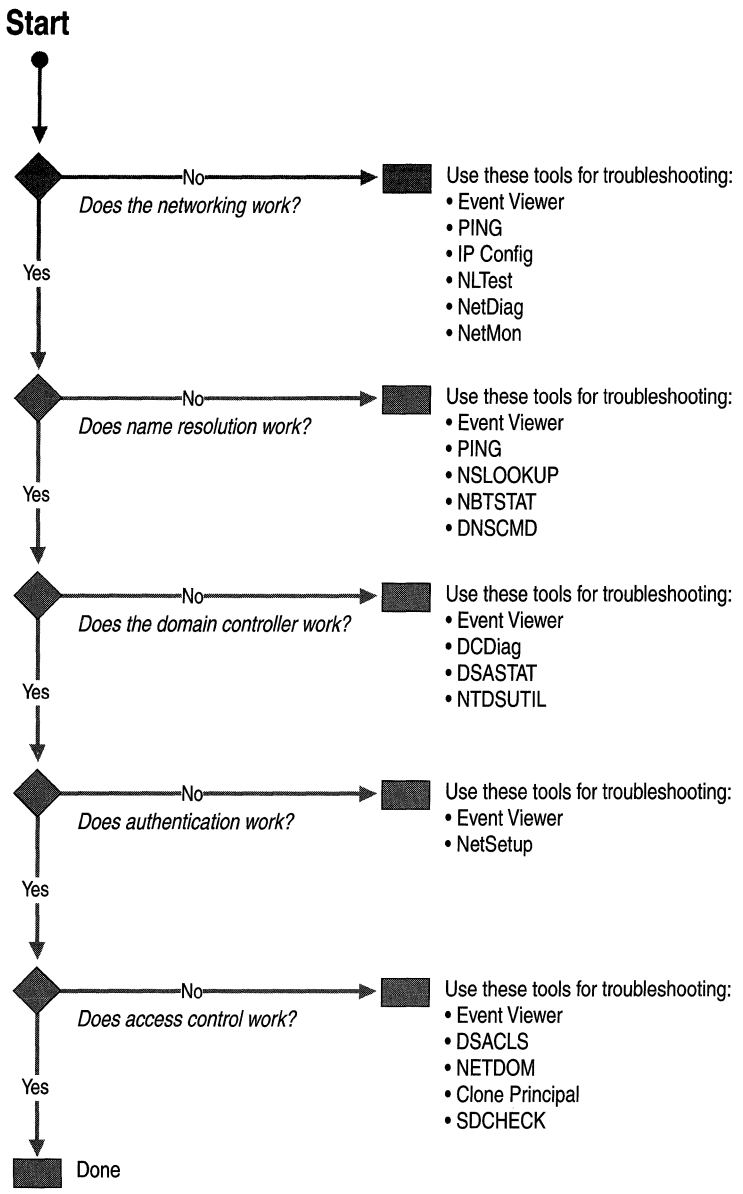


Figure 10.2 Active Directory Diagnostic and Troubleshooting Sequence

Important This chapter makes a best-effort attempt to provide examples of the types of problems you might encounter given the data available, describe the tools you can use to diagnose and identify those problems, and provide suggested solutions. Because Windows 2000 Active Directory is used on a more universal basis, more data will be available on the Microsoft Personal Online Support Web site link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Network Connectivity

The first step toward identifying and diagnosing Active Directory problems is to verify network connectivity. This section discusses diagnostic tools and gives examples of possible network connectivity problems, along with suggested solutions. Examine the following areas to determine whether the network is functioning properly.

Event Viewer

Event Viewer is one of the most useful tools you can use to identify not only networking problems, but also name resolution, directory service, and other types of problems. It categorizes error codes so that you can easily identify a problem, and then analyze the cause of it. Always check the event log to make sure that the directory service is not reporting any events that are indicators of future problems.

To identify network connectivity problems, check the System Log folder and analyze the types of errors and warnings listed. For each error or warning, go to the **Event Properties** page to view the description and the data returned. In the **Data** box, click **Words** and translate the hexadecimal code to decimal. If you see a number in the Event column for the error code, use the **net helpmsg** command to obtain a brief description of the error code.

For example, if the first four digits of the error code are 8007, this indicates a Microsoft® Win32® API or network error. You can then use the **net helpmsg** command to decode these types of errors. To decode the error, first convert the last four digits of the hexadecimal error code to decimal. Then at the command prompt, type the following:

net helpmsg ***<message number in decimal>***

where the *<message number in decimal>* is replaced with the return value you want to decode. The **net helpmsg** command returns a description of the error. For example, if Component Object Model (COM) returns the error 8007054B, convert the low order word, 054B, to decimal (1355). Then type the following:

```
net helpmsg 1355
```

For example, it is recommended that you look in the **Event Properties** page. Specifically, look at the description and the data that are returned. In the **Data** box, translate the hexadecimal code to decimal by clicking **Words**. Then, run **net helpmsg** ***<message number in decimal>***, as in the following example:

```
net helpmsg 1355 equates to "The specified domain either does not exist
or could not be contacted."
```

If you see error codes, such as "access denied" or "bad password," you probably have a security related problem, not a network connectivity problem. The error code "no logon servers" is usually indicative of the fact that you are not able to find a domain controller for that domain. The error code "No logon servers" have a source description of Net Logon. "Access Denied" might have a source description of SAM.

For more information about the Net Helpmsg command and error code explanations, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Hardware

Check that your hardware, such as the network hub, cables, and so on, are functioning properly. For example, if the **Local Area Connection** icon in the Network and Dial-up Connections properties in Control Panel is marked with a red "X," this usually implies that your network cable is disconnected. For more information about checking hardware functionality, see the *Server Operations Guide*.

As a minimum, check that your network adapters and drivers are functioning properly. There are many ways to check the functionality of devices, such as network adapters and drivers, through Control Panel. You can select the **Add/Remove Hardware** icon, and click Add/Troubleshooting a device. Or, you can select the **Hardware** tab from the **System** icon.

Another way of using Control Panel is to click **Hardware Wizard** on the **Hardware** tab of **System Properties** in Control Panel. Select a device from the **Devices** box, and then check to see whether the device is working properly. If you click **Finish**, the Troubleshooter starts when you quit the Add/Remove Hardware wizard. Examine the properties of each device that is displayed by double-clicking the device icon. The status of each device displays on the **General** tab. Click **Troubleshooter** for help if the device is not working properly.

Local Connectivity

Another aspect of verifying network connectivity involves a check of the local area connection. Ensure that you are connected to the network and that the Internet Protocol (IP) addresses are correct. Do this by using the IPConfig command-line tool. The IPConfig tool is used to view and modify IP configuration details used by the computer. With DNS dynamic updates, you can also use IPConfig to register the computer's entries in the DNS service.

▶ To view IP configuration details

1. Type **ipconfig /all** at the command prompt and then press ENTER.
2. Look through the output. Check the following:
   - Do you have an IP address?
   - Do you have a default gateway?
   - Do you have a DHCP server?
3. Use the Ping tool to determine whether you have network connectivity to the default gateway and to the DHCP server.

▶ To test a TCP/IP connectivity by using the ping command

1. At the command prompt, ping the loopback address by typing the following:

   127.0.0.1

   If the **ping** command fails, verify that the computer was restarted after TCP/IP was installed and configured.
2. Ping the IP address of the computer.

   If the **ping** command fails, restart the computer and check the routing table using the **route print** command.
3. Ping the IP address of the default gateway.

   If the **ping** command fails, verify that the default gateway IP address is correct and that the gateway (router) is operational.
4. Ping the IP address of a remote host (a host that is on a different subnet).

   If the **ping** command fails, verify that the remote host IP address is correct, that the remote host is operational, and that all gateways (routers) between this computer and the remote host are operational.
5. Ping the IP address of the DNS server.

If the **ping** command fails, verify that the DNS server IP address is correct, that the DNS server is operational, and that all gateways (routers) between this computer and the DNS server are operational.

Note Use the **ping** command to test TCP/IP connectivity and to determine whether there are network problems between different computers. A failure to connect to the server causes Ping to return a "Request timed out" message.

Example of a Local Area Network Without Network Connectivity

The following example displays an example of an *unsuccessful* TCP/IP configuration for the local area network, with the disabled components indicated in bold text. Also, notice that IP addresses are not displayed. The absence of IP addresses indicates that the local area network is not properly connected.

```
i:> ipconfig /all
Windows 2000 IP Configuration
    Host Name . . . . . . . . . . . . : SERVER1
    Primary DNS Suffix  . . . . . . . : reskit.com
    Node Type . . . . . . . . . . . . : Hybrid   IP Routing Enabled. . .
. . . . . : No  WINS Proxy Enabled. . . . . . . . : No  DNS Suffix
Search List. . . . . : reskit.com
                                        server1.reskit.com
Ethernet adapter Local Area Connection:
Media State . . . . . . . . . . . : Cable Disconnected
    Description . . . . . . . . . . . : 3Com EtherLink XL 10/100 PCI TX
NIC (3C905B-TX)
    Physical Address. . . . . . . . . : 00-10-5A-99-F7-15
```

Example of a Network That Has Network Connectivity

The following example shows a well-connected local area network. Notice that the IP addresses are displayed.

```
i:> ipconfig /all
Windows 2000 IP Configuration   Host Name . . . . . . . . . . . . :
Server1
    Primary DNS Suffix  . . . . . . . : reskit.com
    Node Type . . . . . . . . . . . . : Hybrid   IP Routing Enabled. . .
. . . . . : No  WINS Proxy Enabled. . . . . . . . : No  DNS Suffix
Search List. . . . . : reskit.com
                                        Server1.reskit.com
```

```
Ethernet adapter Local Area Connection:  Connection-specific DNS Suffix
. : Server1.reskit.com
    Description . . . . . . . . . . . : 3Com EtherLink XL 10/100 PCI TX
NIC (3C905B-TX)
    Physical Address. . . . . . . . . : 00-10-5A-99-F7-15    DHCP
Enabled. . . . . . . . . . . : No  IP Address. . . . . . . . . . . . :
172.25.128.19   Subnet Mask . . . . . . . . . . . : 255.255.252.0
    Default Gateway . . . . . . . . . : 172.25.128.1 DNS Servers . . . .
. . . . . . . : 172.26.128.19
    Primary WINS Server . . . . . . . : 172.25.254.203
```

Sending IP Configuration Data to a Text File

You might want to use the IP Configuration data of the local area connection that you obtained by using the IPConfig tool for further analysis. To make it easier to use, you can send the results to a text file. At the command line, type **ipconfig /all > *<local drive>*:*<document title>*.txt>** and then press ENTER. By default, the file is saved in the current directory. To view and modify the output, double-click the file that you created. For more information about TCP/IP troubleshooting, see the *TCP/IP Core Networking Guide*.

Client Connectivity

To determine whether your client is functional, you can use the Netdiag tool. The Netdiag tool helps to isolate networking and connectivity problems by performing a series of tests. These tests, and the key network status information they expose, give you a more direct means of identifying and isolating network problems. Moreover, because this tool does not require that parameters or switches be specified, you can focus on analyzing the output, rather than training users on tool usage.

Specifically, the Netdiag tool tests the following:

- Ndis - Netcard queries test
- IpConfig - IP config test
- Member - Domain membership test
- NetBT Transports - NetBT transports test
- Autonet - Automatic Private IP Addressing (APIPA) address test
- IpLoopBk - IP loopback ping test
- DefGw - Default gateway test
- NbtNm - NetBT name test
- WINS - WINS service test
- Winsock - Winsock test
- DNS - DNS test

- Browser - Redir and Browser test
- DsGetDc - DC discovery test
- DcList - DC list test
- Trust - Trust relationship test
- Kerberos - Kerberos test
- Ldap - LDAP test
- Route - Routing table test
- Netstat - Netstat information test
- Bindings - Bindings test
- WAN - WAN configuration test
- Modem - Modem diagnostics test
- NetWare - NetWare test
- IPX - IPX test

Run netdiag.exe at the command prompt and scan through the output, looking for words like "FATAL."

For more information about the Netdiag tool, see Windows 2000 Support Tools.

Example of Unsuccessful DNS Registrations and Secure Channel Verifications

By using the Netdiag tool, the following example shows failures during DNS registrations and secure channel verifications. (The failures are noted in bold text.)

```
Computer Name: Server1
DNS Host Name: Server1.reskit.reskit.com
System info : NT Server 5.0 (Build 2091)
Processor : x86 Family 6 Model 5 Stepping 2, GenuineIntel
List of installed hotfixes :
Q147222
Netcard queries test . . . . . : Passed
Per interface results:
    Adapter : Local Area Connection
Netcard queries test . . . : Passed
Host Name. . . . . : Server1.dns.reskit.com
IP Address . . . . . : 172.16.85.33
Subnet Mask. . . . . : 255.255.252.0
Default Gateway. . . . . : 172.16.84.1
Primary WINS Server. . . . : 172.16.254.201
Secondary WINS Server. . . : 172.16.254.202
Dns Servers. . . . . : 172.55.254.212
172.16.254.211
```

```
AutoConfiguration results. . . . . : Passed
Default gateway test . . . : Passed
NetBT name test. . . . . : Passed
WINS service test. . . . . : Passed
Global results:
Domain membership test . . . . . : Passed
NetBT transports test. . . . . : Passed
List of NetBt transports currently configured.
NetBT_Tcpip_{69F6A885-C07C-49E4-ABFF-D15FB4B678E8}
1 NetBt transport currently configured.
Autonet address test . . . . . : Passed
IP loopback ping test. . . . . : Passed
Default gateway test . . . . . : Passed
NetBT name test. . . . . : Passed
Winsock test . . . . . : Passed
DNS test . . . . . : Failed
[FATAL]: The DNS registration for Server1.reskit.reskit.com is incorrect
on all DNS servers.
Redir and Browser test . . . . . : Passed
List of NetBt transports currently bound to the Redir
NetBT_Tcpip_{69F6A885-C07C-49E4-ABFF-D15FB4B678E8}
The redir is bound to 1 NetBt transport.
List of NetBt transports currently bound to the browser
NetBT_Tcpip_{69F6A885-C07C-49E4-ABFF-D15FB4B678E8}
The browser is bound to 1 NetBt transport.
DC discovery test. . . . . : Passed
DC list test . . . . . : Failed
Trust relationship test. . . . . : Failed
 [FATAL] Secure channel to domain 'Reskit' is broken.
[ERROR_NO_TRUST_SAM_ACCOUNT]
Kerberos test. . . . . : Skipped
LDAP test. . . . . : Passed
Bindings test. . . . . : Passed
WAN configuration test . . . . . : Skipped
No active remote access connections.
Modem diagnostics test . . . . . : Passed
The command completed successfully
```

For more information about diagnosing and troubleshooting DNS registration problems, see "Name Resolution" later in this chapter. For more information about diagnosing and troubleshooting secure channel problems, see "Authentication" later in this chapter.

Example of Successful Network Connectivity

The following example shows successful client-server connectivity by using the Netdiag tool.

```
Computer Name: Server1
    DNS Host Name: Server1.Reskit.com
    Processor : x86 Family 6 Model 5 Stepping 1, GenuineIntel
    List of installed hotfixes :
        Q147222
Netcard queries test . . . . . : Passed
Per interface results:
Adapter : Local Area Connection
Netcard queries test . . . : Passed
Host Name. . . . . : Server1.Reskit.Reskit.com
        IP Address . . . . . : 172.16.128.19
        Subnet Mask. . . . . : 255.255.252.0
        Default Gateway. . . . . : 172.16.128.1
        Primary WINS Server. . . . : 172.16.254.203
        Dns Servers. . . . . : 172.16.128.19
AutoConfiguration results. . . . . : Passed
Default gateway test . . . : Passed
NetBT name test. . . . . : Passed
            No remote names have been found.
WINS service test. . . . . : Passed
Global results:
Domain membership test . . . . . : Passed
NetBT transports test. . . . . : Passed
    List of NetBt transports currently configured.
        NetBT_Tcpip_{F5A7E415-9D0B-444B-8028-11238D589BD0}
    1 NetBt transport currently configured.
Autonet address test . . . . . : Passed
IP loopback ping test. . . . . : Passed
Default gateway test . . . . . : Passed
NetBT name test. . . . . : Passed
Winsock test . . . . . : Passed
DNS test . . . . . : Passed
    PASS - All the DNS entries for DC are registered on DNS server
172.16.128.19.

Redir and Browser test . . . . . : Passed
    List of NetBt transports currently bound to the Redir
        NetBT_Tcpip_{F5A7E415-9D0B-444B-8028-11238D589BD0}
    The redir is bound to 1 NetBt transport.
List of NetBt transports currently bound to the browser
        NetBT_Tcpip_{F5A7E415-9D0B-444B-8028-11238D589BD0}
    The browser is bound to 1 NetBt transport.
DC discovery test. . . . . : Passed
DC list test . . . . . : Passed
Trust relationship test. . . . . : Skipped
```

```
Kerberos test. . . . . . . . . . . : Passed
LDAP test. . . . . . . . . . . . . : Passed
Bindings test. . . . . . . . . . . : Passed
WAN configuration test . . . . . . : Skipped
    No active remote access connections.
Modem diagnostics test . . . . . . : Passed
The command completed successfully
```

Sending Netdiag Data to a Text File

You might want to use the network client and server connection data that you obtained by using the Netdiag tool for further analysis. To make it easier to use, you can send it to a text file. From the command line, type **NetDiag.exe > <*local drive*>:\<*document title*.txt>**, and then press ENTER. By default, the file is saved to the current directory. To view and modify the output, double-click the file.

Domain Controller Connectivity

Verify that the domain controller is functional. To verify network connectivity for domain controllers, use the Ping tool to check your domain controller, as well as other domain controllers in the network. If they are connected, the IP addresses are going to be properly resolved.

For example, carry out the following commands:

```
ping <your domain controller>
ping <additional domain controller>
ping <additional domain controller>
```

Does at least one of the previous procedures succeed? Also verify that it resolves to the correct IP address for the computer. If it does, go to the next section.

Client-Domain Controller Trust Relationships

There are many reasons why the secure channel between a client and a domain might break. One example is if you don't have the appropriate access permissions, as shown in the following example:

```
[FATAL] Secure channel to domain 'RESKIT' is broken.
[ERROR_ACCESS_DENIED]

> nltest /sc_query:reskit
nltest /sc_query:reskit
Flags: 0
Trusted DC Name
Trusted DC Connection Status Status = 5 0x5 ERROR_ACCESS_DENIED
The command completed successfully
```

To validate trust connections, you normally test the secure channel:

- Nltest /sc_query is used to query the status of the secure channel.
- Nltest /sc_reset *<domain name>* can be used to force renegotiations of the secure channel.
- Nltest /sc_reset *<domain name>**<computer name>* can be used to force a secure channel onto a particular domain controller.

Note The results of an Nltest /sc_query are unreliable—it returns the status of the channel when it was used last time and not the current status. The recommended sequence of verifying the trust is to run nltest /sc_query. If that returns success, run nltest /sc_reset:*<domain>**<dcname returned by /sc_query>*.

▶ **To determine the cause of trust relationship problems**

1. Log on with a local account.
2. Set Net Logon flags by using the Nltest tool as follows:

   nltest /dbflag:0x2000ffff.
3. Run nltest as follows**: nltest /sc_reset:*<domain name to which you think your computer is joined>***.

   The *%windir%*\debug\netlogon.log explains why the secure channel setup is not possible. One possible reason is that SYSVOL isn't ready on the computer. By examining the Net Logon.log file, you can find the following error:

   ```
   08/30 10:15:19 [MAILSLOT] Returning paused to 'Reskit1' since: SysVol
   not ready
   ```

Common trust failures are the following:

- No SAM Trust Account - typically means that the computer account does not exist.
- Access denied—typically means that the trust passwords do not match. Be cautious when you get access denied—you get the same error back if you weren't granted permissions to run sc_query or sc_reset.

Note Installing computers that use the same computer name is often the reason for computer account problems, hence broken secure channels. The common way to get around this problem is to perform the join again.

Another example of client-domain controller trust relationship problems:

```
D:>nltest /sc_query:reskit
Flags: 0
Trusted DC Name
Trusted DC Connection Status Status = 1787 0x54b
ERROR_NO_SAM_TRUST_ACCOUNT
The command completed successfully
```

The preceding example implies that the client assumes it has joined the domain. However, the client is not able to find a computer account registered for itself in the domain controller.

For more information about trust relationships, see "Active Directory Logical Structure" in this book.

Trust Relationship Diagnostic Tools

The Nltest command-line tool enables you to check trust relationships, as well as the connectivity and traffic flow between a network client and a domain controller. Nltest checks the secure channel to make sure that both Windows 2000–based and Windows NT 4.0–based clients can connect to domain controllers. The tool also discovers domains and sites. Further, you can list the domain controllers and Global Catalog servers that are available. It supports user operations to identify which domain controllers are capable of logging on a specific user, as well as browsing specific user information.

To ensure that cached information is not being used when a Windows 2000–based client discovers a domain controller, carry out the **/force** command in the Nltest tool. At the command prompt, type **nltest /dsgetdc:<*your domain name*> /force** and then press ENTER.

Note Nltest /dsgetdc: is used to exercise the locator. Thus /dsgetdc:<*domain name*> tries to find the domain controller for the domain. Using the force flag forces domain controller location rather than using the cache. You can also specify options such as **/gc** or **/pdc** to locate a Global Catalog or a primary domain controller emulator. For finding the Global Catalog, you must specify a "tree name," which is the DNS domain name of the root domain.

If you receive the following error, ERROR_NO_LOGON_SERVERS while using the Nltest tool to query the secure channel, this is usually indicative of the inability to find a domain controller for that domain. Run **nltest /dsgetdc: <*DomainName*>**: to verify whether you can locate a domain controller. If you are unable to find a domain controller examine DNS registrations and network connectivity.

For more information about verifying DNS registrations, see "Name Resolution" later in this chapter.

The following example shows an unsuccessful attempt to find a domain controller for the domain:

```
>nltest /SC_QUERY:reskit
Flags: 0
Trusted DC Name
Trusted DC Connection Status Status = 1311 0x51f ERROR_NO_LOGON_SERVERS
The command completed successfully
```

The following example shows an unsuccessful attempt to locate the domain controller for the domain using **/dsgetdc** switch:

```
:\>nltest /dsgetdc:reskit /force
DsGetDcName failed: Status = 1355 0x54b ERROR_NO_SUCH_DOMAIN
```

The following example shows a successful attempt to find a domain controller for the domain:

```
H:\>nltest /dsgetdc:reskit /force
           DC: \\server1
      Address: \\172.16.132.197
     Dom Guid: ca21b03b-6dd3-11d1-8a7d-b8dfb156871f
     Dom Name: reskit
  Forest Name: reskit.com
 Dc Site Name: Default-First-Site-Name
Our Site Name: Default-First-Site-Name
        Flags: GC DS LDAP KDC TIMESERV WRITABLE DNS_FOREST CLOSE_SITE
The command completed successfully
```

DHCP Server Issues

To determine if the DHCP server is the problem, you can release your IP address, restart DHCP, and then renew your IP address by carrying out the following commands:

```
ipconfig /release
net stop dhcp
net start dhcp
ipconfig /renew
```

If you still cannot connect the client to the domain controller (even though you have a good IP address), a Network Monitor sniffer trace of the connection attempt might be useful.

For more information about DHCP, see "Dynamic Host Configuration Protocol" in the *TCP/IP Core Networking Guide*. For more information about DHCP Server, see Windows 2000 Server Help.

Using Network Monitor to Analyze Network Traffic Issues

Network Monitor sniffer traces can help you trace *all* of the traffic to and from a computer; as well as to and from the DHCP server that issues IP addresses. A "light" version is delivered with Windows 2000 Server. However, to use Network Monitor's full capabilities, you need the full version included with Microsoft® Systems Management Server.

▶ **To install Network Monitor**

- From the **Start** menu, point to the following:
  - **Settings**
  - **Control Panel**
  - **Add/Remove Programs**
  - **Add/Remove Windows Components**
  - **Management and Monitoring Tools**
  - **Details**
  - **Network Monitoring Tools**

As long as you have installed the full version available from Systems Management Server, you can capture and view every packet on the network. Network Monitor isolates the network layer where a problem occurred, or where an operation failed, and helps you determine the cause of the problem.

Note Run Network Monitor on the computer that is having the problems, or on another computer that feeds into the same microhub. For more information about Network Monitor, see the *Server Operations Guide*.

Because the Network Monitor sniffer trace captures the entire exchange that occurs on the wire, you can scan this quickly and determine the source of a particular problem. For example, if you have a reproducible problem, a sniffer trace (or capture) helps determine the actual operation that failed. It displays the speed of operations, the source to network traffic, if packets are being dropped or if processes are experiencing time-outs.

Example of Monitoring Network Traffic

A typical example of monitoring network traffic by using Network Monitor is one where you install Network Monitor on your main working computer. Assuming that all of your computers are connected to the same hub, you can use your main computer to sniff each of the other computers on the network. For example, to monitor another computer, obtain its address and add it, as an Ethernet address, with the name of the monitored computer. Next, you can filter the sniffer trace so that you capture activity for the monitored computer.

Note The Ethernet address (and not the IP address) is used for filtering when you want to see all traffic, be it IP or IPX. This is useful because delays can involve multiple transports.

When you are finished viewing the capture of the monitored computer, you can select another filter.

Example of a DNS Dynamic Update Protocol Frame Through Network Monitor

Windows 2000 includes the ability for clients to register DNS records automatically with DNS servers configured to accept these updates. The following example shows the captured network frame and indicates that the frames are client requests to dynamically update the DNS server.

```
DNS: 0x1B:Dyn Upd UPD records to MYSERVER.mycorp.com. of type Host Addr
         DNS: Query Identifier = 27 (0x1B)
         DNS: DNS Flags = Query, OpCode - Dyn Upd, RCode - No error
             DNS: 0.............. = Request
   -----> DNS: .0101.......... = Dynamic Update
             DNS: .....0..........  = Server not authority for domain
             DNS: ......0.........  = Message complete
             DNS: .......0........  = Iterative query desired
             DNS: ........0.......  = No recursive queries
             DNS: .........000....  = Reserved
             DNS: ............0000 = No error
         DNS: Zone Count = 1 (0x1)
         DNS: Prerequisite Section Entry Count = 0 (0x0)
         DNS: Update Section Entry Count = 3 (0x3)
         DNS: Additional Records Count = 0 (0x0)
         DNS: Update Zone: mycorp.com. of type SOA on class INET addr.
             DNS: Update Zone Name: mycorp.com.
             DNS: Update Zone Type = Start of zone of authority
             DNS: Update Zone Class = Internet address class
         DNS: Update: MYSERVER.mycorp.com. of type Host Addr on class
Req.
             for any(2 records present)
             DNS: Resource Record: MYSERVER.mycorp.com. of type Host
Addr
             on class Req. for any(2 records present)
                 DNS: Resource Name: MYSERVER.mycorp.com.
                 DNS: Resource Type = Host Address
                 DNS: Resource Class = Request for any class
                 DNS: Time To Live = 0 (0x0)
                 DNS: Resource Data Length = 0 (0x0)
```

This frame also includes the record to be written:

```
DNS: Resource Record: MYSERVER.mycorp.com. of type Host Addr
                  on class INET addr.
                 DNS: Resource Name: MYSERVER.mycorp.com.
                 DNS: Resource Type = Host Address
                 DNS: Resource Class = Internet address class
                 DNS: Time To Live = 1200 (0x4B0)
                 DNS: Resource Data Length = 4 (0x4)
                 DNS: IP address = 100.2.0.3  ---> example IP address
```

The version of Microsoft Network Monitor included with Windows 2000 Server parses these frames correctly and displays DNS dynamic update frames.

Note If you are using a third-party version or an earlier version of Network Monitor, you can identify DNS dynamic update frames by the four bits in the "DNS Flags" section of the frame.

The following example displays the four bits in the DNS Flags section:

```
DNS: 0x17:Std Qry for mycorp.com. of type SOA on class INET addr.
        DNS: Query Identifier = 23 (0x17)
        DNS: DNS Flags = Query, OpCode - Std Qry, RD Bits Set, RCode -
No
             error
            DNS: 0.............. = Query
   -----> DNS: .0101.......... = Reserved (a value of 5 (0101) here =
            Dynamic DNS Update Record)
            DNS: .....0..... = Server not authority for domain
            DNS: ......0..... = Message complete
            DNS: .......1..... = Recursive query desired
            DNS: ........0..... = No recursive queries
            DNS: .........000.... = Reserved
            DNS: ............0000 = No error
```

This frame also includes the record to be written:

```
        DNS: Authority Section: MYSERVER.mycorp.com. of type Host Addr
on
             class INET addr.
            DNS: Resource Name: MYSERVER.mycorp.com.
            DNS: Resource Type = Host Address
            DNS: Resource Class = Internet address class
            DNS: Time To Live = 3600 (0xE10)
            DNS: Resource Data Length = 4 (0x4)
            DNS: IP address = 100.2.0.3   ---> example IP address
```

When you use Network Monitor, be aware of the following:

- The Network Monitor user interface is context sensitive.
- The Network Monitor toolbars are helpful.
- Different menu behavior results when you select either the Capture Window or a capture window.
- You can save a capture to a *.cap file for later use.
- You can save a filter and reload it quickly.
- You can set Network Monitor to display the time from a previous frame (if you are doing performance analysis).
- If you have an unnamed address in one of the columns, you can right-click and name it for easier reading.
- You can set different colors for each protocol in a multiprotocol debug.
- Capture Buffer Settings lets you configure larger buffers, which is useful for long sniffs.
- For LDAP, the Global Catalog port (port 3268) is not recognized as an LDAP port. You need to add a line in parsers\tcpip.ini to enable this:

  3268 = LDAP (under the TCP_Handoffset section)
- Press **F8** to modify the filter. Let the direction arrow in the center remain <->. Choose **Any** on one side, and the name of your problem computer on the other. If the name of the computer doesn't appear, you must create a new entry (**Edit Addresses**, **Add**) for the computer, select the media access control (MAC) address. The MAC address is the Physical Address of the adapter when you run **ipconfig /all** (that is, 00C092FE1DAA). Click **OK**.
- Press **F10** to start capturing. Perform the operation that is not working correctly. The packets that are being captured are displayed at the bottom of the Network Monitor pane. Stop capturing, save the file, and attach it to the problem report.

For more information about the Network Monitor tool, see the *Server Operations Guide*. For more information about the DNS service, see "Windows 2000 DNS" in the *TCP/IP Core Networking Guide*.

Redirector Issues

To determine whether there is a problem with the redirector, type **net config rdr** at the command prompt, and then press ENTER.

If the workstation is not active on at least one transport, you see something similar to the example that follows. The **net config rdr** command shows how the redirector or workstation is currently configured on your computer.

```
Computer name                        \\Reskit
User name                            User1

Workstation active on                NetbiosSmb <000000000000>
Software version                     Windows 2000

Workstation domain                   NTWKSTA
Logon domain                         RESKIT

COM Open Timeout (sec)               0
COM Send Count (byte)                16
COM Send Timeout (msec)              250
The command completed successfully.
```

The workstation must be active on at least one transport. NetBT Tcpip, for example, as shown.

```
Computer name                        \\Reskit
User name                            User1

Workstation active on NetbiosSmb <000000000000>
NetBT_Tcpip_{24B6F8FC-0CE6-11D1-8F1A-A0BC38451EB2} (00C04FD8D37F)
Software version                     Windows 2000

Workstation domain                   NTWKSTA
Logon domain                         RESKIT

COM Open Timeout (sec)               0
COM Send Count (byte)                16
COM Send Timeout (msec)              250
The command completed successfully.
```

If not, you either have networking problems in the redirector, the transport, or in Plug and Play functionality. One main cause of not having at least one transport bound to the redirector or workstation is a duplicate name conflict.

Note You might experience a delay when you attempt to connect to network resources from a system with multiple redirectors installed. This delay happens only the first time that you attempt the connection.

For more information about the redirector, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Name Resolution

If you are having problems connecting to Active Directory and you have already successfully verified network connectivity, there might be a name resolution problem. If you cannot find other computers or network resources when you perform queries, this might mean that DNS domain names are not being resolved to IP addresses.

This section discusses diagnostic tools and gives examples of possible name resolution problems, along with suggested solutions. The first step toward identifying and diagnosing Active Directory name resolution problems is to review how a Windows 2000–based computer registers names and locates domain controllers.

To review, whenever you start up a Windows 2000 domain controller, it registers two types of names:

- A DNS domain name with the DNS service.
- A NetBIOS name with WINS or with another transport-specific service if the computer has NetBIOS enabled.

DNS records registered by domain controllers include multiple SRV records, A records, and CNAME records identifying the domain controllers' location in a specific domain and forest.

When a Windows 2000–based computer logs on to a domain, the computer does one of two things:

- Queries DNS to find a domain controller with which to authenticate (if the name of the logon domain is a DNS name).
- Sends a mailslot message to find a domain controller for the specified domain (if the name of the logon domain is a NetBIOS name).

After the computer finds a domain controller, the information is cached so that a new query is not required for subsequent domain controller discoveries.

For more information about Domain Controller Locator and discovery, see "Name Resolution in Active Directory" in this book.

An answer to the following question can help you determine whether domain controller names are being resolved properly by DNS.

> Can you look up names and addresses of network resources by using the Ping tool or the **net use** command?

Negative responses require further investigation. Begin by verifying your DNS configuration, followed by ensuring that DNS names are properly registered. Also, this section discusses a number of Resource Kit tools that can help you diagnose and repair name resolution problems.

DNS Registration and Consistency

A good practice following the installation of Active Directory is to verify that the DNS resource records for the domain controller are written to the DNS server. This is known as *registration*.

There are two specific types of registration; registration for the computer A and PTR records and registration for the domain controller SRV records, A records, and CNAME records in the DNS server. It is recommended you check both types of registrations.

Note If DNS records are not registered in the DNS server, no other computer or user is able to locate the domain controller. If DNS records of a computer are not registered, you see DNS errors in the System log in Event Viewer.

To review, the Net Logon service registers records when the domain controller is restarted and when the Net Logon service starts. The Net Logon service sends DNS dynamic update queries for its SRV records, A records, and CNAME records every hour to ensure that the DNS server always has these records registered. As described in RFC 2136, dynamic update is a recent addition to the DNS standard. It defines a protocol for updating a DNS server with new or changed records dynamically.

All Windows 2000 domain controllers must use DNS as their locator service. Every Windows 2000–based domain controller dynamically registers service (SRV) records in DNS, which allow servers to be located by service type (in this example, LDAP) and protocols (for example, TCP and UDP). In addition to registering LDAP-specific SRV records, Net Logon also registers Kerberos v5 authentication protocol–specific SRV records to enable locating servers that run the Kerberos Key Distribution Center (KDC) service.

For Active Directory–integrated zones, the DNS server stores all the records in the zone in Active Directory. It is possible that a record is updated in Active Directory, but has not replicated to all DNS servers loading the zone. This might cause consistency problems. By default, all DNS servers that load zones from Active Directory, poll Active Directory at a set interval—typically every five minutes—and update the directory for any incremental changes to the zone. In most cases, a DNS update takes no more than 20 minutes to replicate to all DNS servers used in an Active Directory domain environment employing default replication settings and reliable high-speed links. Thus, it is vital to ensure the consistency of directory-integrated zone data. In Windows 2000, DNS consistency plays a similar to the role of WINS in Windows NT 4.0 as the source of logon and trust relationship issues.

Tools Used for Diagnosing and Troubleshooting DNS Issues

The tools discussed in the following sections are useful for troubleshooting DNS problems.

Event Viewer

The DNS Server log in the Event Viewer Administrative Tool console is one of the primary tools you can use to identify DNS name resolution problems. To view messages about the DNS server, you need to check the DNS Server folder. To view messages about the DNS client check the System Log folder. For more information about DNS, see "Windows 2000 DNS"cncf_imp_VSIN in the *TCP/IP Core Networking Guide*.

Event Viewer logs errors with the Windows 2000 operating system and services such as the DNS service. If you are having problems with DNS, you can check Event Viewer for DNS-related events.

▶ **To open Event Viewer**

- From the **Start** menu, point to **Programs** and **Administrative Tools**, and then click **Event Viewer**.

  To view messages about the DNS server, click **DNS Server**.

  –Or–

  To view messages about the DNS client, click **System Log**.

For more information about Event Viewer, see Windows 2000 Help.

On a client, if you see DNS event errors in the System log, that is an indication that your client has a problem dynamically updating DNS records. On a domain controller, if you see the Netlogon event error 5781, that usually is an indication that your domain controller has a problem dynamically updating DNS records for the domain controller. Specific methods for troubleshooting these errors are be discussed in this chapter.

Using Nslookup for Name Resolution

You can use Nslookup to perform DNS queries and to examine the contents of zone files on local and remote servers.

To use Nslookup in interactive mode and to verify name resolution, at the command prompt, type the following:

```
NSLOOKUP
```

You might receive an error similar to the following:

```
DNS request timed out.
    timeout was 2 seconds.
*** Can't find server name for address 172.16.0.0: Timed out
Default Server:  UnKnown
Address:  172.16.0.0
```

The error " *** Can't find server name for address 172.16.0.0: Timed out" can be ignored. This error usually implies that there is no PTR record corresponding to the DNS server. Hence, if the Nslookup tool can't find a server name for the server's IP address, it uses Unknown as the server name but does not affect your queries.

For more information about the Nslookup tool and configuring a reverse lookup zone, see "Windows 2000 DNS" in the *TCP/IP Core Networking Guide*.

Using Netdiag to Verify DNS Registration

The Netdiag tool helps to isolate networking and connectivity problems by performing a series of tests. If you are unable to resolve a name, you might be experiencing DNS registration or consistency problems. To confirm this, answer the following questions:

When you run Netdiag, do you receive any DNS error messages? For example:

```
DNS test . . . . . . . . . . . . . : Failed
              [FATAL]: The DNS registration for SERVER1 in reskit.com is
incorrect on all DNS servers.
OR

DNS test . . . . . . . . . . . . . : Failed

.....[FATAL] No DNS servers have our DNS records for this DC
registered
```

If you receive this error refer to the methods used in this section to troubleshoot and resolve DNS registration failures.

Note To verify the DNS registration for your domain, the best tool to use is **netdiag /debug**, which must be run on all domain controllers.

To refresh all DHCP leases and re-register DNS names for computers, use the **ipconfig /registerdns** command. To refresh and re-register DNS names for domain controllers, stop and start the Net Logon service. By default, the Net Logon service automatically re-registers DNS names every hour. For information about DHCP, see "Dynamic Host Configuration Protocol" in the *TCP/IP Core Networking Guide*.

Using DNSCMD to Check Consistency

Dnscmd.exe is a command-line tool that you can use to view the properties of DNS servers, zones, and resource records. To be able to check your DNS server configuration, use the Dnscmd tool or the DNS Manager console to obtain information about the DNS server and obtain statistics about its performance.

Dnscmd is also used to manually modify DNS server properties, to create and delete zones and resource records, and to force replication events between DNS server physical memory and DNS databases and data files.

For more information about Dnscmd.exe, see "Dnscmd.exe: DNS Server Troubleshooting Tool" in *Windows 2000 Resource Kit* Tools Help on the Resource Kit companion CD.

Identifying and Verifying DNS Problems

There are three main scenarios that you might encounter:

- The user is not be able to log on.
- While running the Active Directory Installation Wizard, problems emerge when trying to find an existing domain controller in an existing forest or domain.
- A domain controller is not able to find another domain controller.

Verifying Your DNS Configuration

Because DNS locates network resources for Active Directory, you need to ensure that it is configured properly. For more information about DNS configuration, see "Name Resolution in Active Directory" in this book. However, begin by answering the following questions:

- Have you verified your DNS client configuration?

- Have you verified your DNS server configuration?
- Have you verified that needed records are registered in DNS and replicated to all authoritative DNS servers?

Before verifying the configuration of the DNS server and the existence of records, verify that your DNS client settings are correct.

▶ **To verify DNS client settings**

1. Right-click **My Network Places**, and then click **Properties**.
2. Right-click the connection for which you want to configure the DNS server, and then click **Properties**.
3. Click **Internet Protocol (TCP/IP)**, and then click **Properties**.
4. On the **Internet Protocol (TCP/IP) Properties** page, enter the IP address of the existing DNS server in the **Preferred DNS server** field. Add the IP address of an alternate DNS server in the **Alternate DNS server** field.
5. If you need to specify more than one alternate DNS server, click **Advanced**, click the **DNS** tab, and then enter the servers in the **DNS server addresses** box.

You can use the command-line tool Ipconfig to view your DNS client settings, to view and reset cached information used locally for resolving DNS name queries, and to register the resource records for a dynamic update client. If you use Ipconfig with no parameters, it displays DNS information for each adapter, including the domain name and DNS servers used for that adapter. Table 10.1 shows some command-line options available with Ipconfig.

Table 10.1 Ipconfig Command-Line Options

Command	Action
ipconfig /all	Displays additional information about DNS, including the FQDN and the DNS suffix search list.
ipconfig /flushdns	Flushes and resets the DNS resolver cache.
ipconfig /displaydns	Displays the contents of the DNS resolver cache.
ipconfig /registerdns	Refreshes all DHCP leases and registers any related DNS names. This option is available on Windows 2000–based computers unless the DHCP Client service is stopped.
ipconfig /release [*adapter*]	Releases all DHCP leases.
ipconfig /renew [*adapter*]	Refreshes all DHCP leases and dynamically updates DNS records. This option is available only on computer that are running the DHCP Client service.

Note In addition to flushing the cache by using Ipconfig, you can stop and flush the cache by stopping and starting the DNS Client service. For more information about flushing the cache, see "Windows 2000 DNS" in the *TCP/IP Core Networking Guide.*

After you confirmed that the client properly points to the primary and alternate DNS Servers, if the latter are not authoritative for the names to be resolved, confirm that they can recursively resolve the names that the client attempts to resolve. For more information on recursively resolving names, see Windows 2000 DNS" in the *TCP/IP Core Networking Guide.*

After you have verified that the client is properly configured and the preferred and alternate DNS servers are capable of the recursive name resolution, you need to verify that the DNS server contains the necessary records.

The following section discusses the list of resource records registered by the Net Logon service running on domain controllers.

Verifying DNS Registration from the Domain Controller

Besides A and PTR records that are registered by any Windows 2000 computer, the domain controllers also register additional records that indicate their role. Every time that the Net Logon service starts (including restarting the domain controller) the service attempts to register some or all SRV resource records as shown in the following example.

The SRV resource records are registered by starting the Net Logon service, which enlists the records in the Netlogon.dns file under the *%systemroot%*\System32\config folder.

Note To re-register the SRV resource records, at the command prompt, type **net stop netlogon,** and then type **net start netlogon**.

An example of a Netlogon.dns file:

```
reskit.com. 600 IN A 172.16.128.19
_ldap._tcp.reskit.com. 600 IN SRV 0 100 389 SERVER1.reskit.com.
_ldap._tcp.pdc._msdcs.reskit.com. 600 IN SRV 0 100 389
SERVER1.reskit.com.
_ldap._tcp.gc._msdcs.reskit.com. 600 IN SRV 0 100 3268
SERVER1.reskit.com.
_ldap._tcp.708b2ee5-a806-47c4-b6ee-
0dbe0e496b36.domains._msdcs.reskit.com. 600 IN SRV 0 100 389
SERVER1.reskit.com.
gc._msdcs.reskit.com. 600 IN A 172.16.128.19
11992d81-2208-4ff5-8641-b9c6a644064a._msdcs.reskit.com. 600 IN CNAME
SERVER1.reskit.com.
_kerberos._tcp.dc._msdcs.reskit.com. 600 IN SRV 0 100 88
SERVER1.reskit.com.
_ldap._tcp.dc._msdcs.reskit.com. 600 IN SRV 0 100 389
SERVER1.reskit.com.
_kerberos._tcp.reskit.com. 600 IN SRV 0 100 88 SERVER1.reskit.com.
_gc._tcp.reskit.com. 600 IN SRV 0 100 3268 SERVER1.reskit.com.
_kerberos._udp.reskit.com. 600 IN SRV 0 100 88 SERVER1.reskit.com.
_kpasswd._tcp.reskit.com. 600 IN SRV 0 100 464 SERVER1.reskit.com.
_kpasswd._udp.reskit.com. 600 IN SRV 0 100 464 SERVER1.reskit.com.
_ldap._tcp.Default-First-Site-Name._sites.reskit.com. 600 IN SRV 0 100
389 SERVER1.reskit.com.
_ldap._tcp.Default-First-Site-Name._sites.gc._msdcs.reskit.com. 600 IN
SRV 0 100 3268 SERVER1.reskit.com.
_kerberos._tcp.Default-First-Site-Name._sites.dc._msdcs.reskit.com. 600
IN SRV 0 100 88 SERVER1.reskit.com.
_ldap._tcp.Default-First-Site-Name._sites.dc._msdcs.reskit.com. 600 IN
SRV 0 100 389 SERVER1.reskit.com.
_kerberos._tcp.Default-First-Site-Name._sites.reskit.com. 600 IN SRV 0
100 88 SERVER1.reskit.com.
_gc._tcp.Default-First-Site-Name._sites.reskit.com. 600 IN SRV 0 100
3268 SERVER1.reskit.com.
```

To join a domain this record is used:

_ldap._tcp.dc._msdcs.*<existing domain the domain controller is joining>*

To join a tree this record is used:

_ldap._tcp.dc._msdcs.*<parent domain of the newly created domain in the existing tree>*

To join a forest, this record is used:

_ldap._tcp.dc._msdcs.*<ForestRoot>*

To confirm that appropriate records are registered in DNS, you can use the Nslookup tool or the DNS Management console.

The following example shows how to use an Nslookup query to verify that the generic records for the Reskit.com domain; _ldap._tcp.reskit.com exists in DNS:

```
C:\>nslookup
Default Server:  dc1.reskit.com
Address:  10.0.0.14
> set type=SRV
> _ldap._tcp.reskit.com
Server:  dc1.reskit.com
Address:  10.0.0.14
_ldap._tcp.reskit.com   SRV service location:
          priority       = 0
          weight         = 0
          port           = 389
          svr hostname   = dc1.reskit.com
_ldap._tcp.reskit.com   SRV service location:
          priority       = 0
          weight         = 0
          port           = 389
          svr hostname   = dc2.noam.reskit.com
dc1.reskit.com     internet address = 10.0.0.14
dc2.reskit.com     internet address = 10.0.0.15
```

Note Remember that for the Domain Control Locator to be successful, the client must resolve not only domain controller names through target hosts in the SRV resource records, but also the A records corresponding to the target host names. Usually these A records are returned in the additional section in the DNS server's response. If these records are not returned, use the Nslookup tool to verify their existence in DNS.

From the **nslookup** command prompt, type the host name of the record stored on the DNS server.

Note The host name that you type must be dot terminated.

Successful and unsuccessful query results might include the following:

```
> dc1.reskit.com.
Server:  my_DNS_servername
Address:  172.16.0.0

Name:    dc1.reskit.com
Addresses: 172.31.94.18
```

This means that DNS contains the A record and the server is responding back with the answer: 172.31.94.18. Next, you need to verify whether this IP address is the actual IP address for your computer, DC1. You can go to computer DC1 and type **ipconfig** to determine its real IP address, or you can use the Nbtstat tool and run the following command:

nbtstat -A 172.31.94.18.

The Nbstat tool is discussed in more detail later in this chapter.

If you detect that some of the records that must be registered are not registered, you need to troubleshoot your DNS record registrations.

Troubleshooting DNS Record Registration Failure

If you have problems with DNS record registration, verify the configuration of the DNS client on the domain controller and configuration of the zone authoritative for the records to be registered.

Verifying Registration of DNS Records for the Computer

Use the following steps to diagnose and troubleshoot your problem:

- Check whether you have any DNS and Net Logon event errors in the system. Log on to the computer that is responsible for registering the records.
- Run the Netdiag tool, and look for the expression [FATAL] in the results.

  For more information about using the Netdiag tool, see "Network Connectivity" earlier in this chapter.
- Verify whether any DNS server has the zone authoritative for the name to be registered and whether the zone allows dynamic update:
- Connect to the DNS server and open the DNS Manager console. Check whether you have that zone created on the DNS server. To do this, right-click **Zone**, click **Properties** to bring up the zone property, and then click the **General** tab. Check the **Allow Dynamic Update** field, and verify that it is not set to **No**. Click the **Start of Authority (SOA)** tab. Then check the **Primary server field**, and verify that the primary server field displays a valid Fully Qualified Domain Name (FQDN). For more information about primary server field, SOA, and zones, see "Windows 2000 DNS" in the *TCP/IP Core Networking Guide*.
- Verify that a computer that need to register DNS records is properly configured with the preferred and alternate DNS servers.
- Close the property page, and verify that DNS contains a correct A record for the FQDN name.

- Verify the configuration of the preferred and alternate DNS servers. For more information on preferred and alternate DNS servers, see "Windows 2000 DNS" in the *TCP/IP Core Networking Guide*.
- If the primary and alternate DNS servers are not authoritative for the names to be registered, verify that the primary and alternate DNS servers can recursively find the authoritative DNS server. For more information about how to verify that the primary and alternate DNS servers can recursively find the authoritative DNS server, see "Windows 2000 DNS" in the *TCP/IP Core Networking Guide*.

If all the preceding steps have been verified, the DNS server can receive dynamic updates from the clients, and then follow the troubleshooting steps in the following section.

Solving Problems with Dynamic Update

If dynamic update does not register a resource record properly, use the following process to troubleshoot your problem.

- If the client does not point to a valid DNS server (for example, you can find out which DNS servers you are pointing to by typing **ipconfig /all** from the command prompt), change the DNS server list. To change the list, right click **My Network Places**, and choose **Properties**. Right click **Local Area Connection**, and choose **Properties.** Click **TCP/IP**, and then click **Properties**. Change the DNS server list. Click **Use the following DNS server addresses**, and then type in the valid DNS servers.

  Force the client that is experiencing registration failures to renew its registration by typing the following:

  ipconfig /registerdns

- Wait approximately five minutes, check Event Viewer, and then check for any DNS events registered.

- Check whether dynamic update is enabled for the zone that is authoritative for the name the of the client that is attempting the update. Run the Netdiag tool to verify whether the registration failure has been corrected.

  Note You should see at least one DNS server has the DNS entry registered correctly. Other DNS servers still might not have the DNS entry registered because of replication latency from one DNS server to another.

  For more information about dynamic updates and secure dynamic updates, see "Windows 2000 DNS" in the *TCP/IP Core Networking Guide*.

DNS Troubleshooting Tips

The following suggestions will help you diagnose other problems you might have with DNS:

- To rule out other problems, check whether the dynamic update client lists the primary DNS server for the zone as its preferred DNS server.

  This is not necessary for dynamic update to work; however, if the client lists a preferred server other than the primary DNS server for the zone, many other problems could cause the failure, such as a network connectivity problem between the two servers or a prolonged recursive lookup for the primary server of the zone. To ascertain the preferred DNS server for the client, check the IP address configured in the TCP/IP properties for the client's network connection, or at the command prompt, type **ipconfig /all**. If the zone is directory-integrated, any DNS server that hosts a directory-integrated copy of the zone can process the updates.
- Check whether the zone is configured for secure dynamic update.

  If the zone is configured for secure dynamic update, the update can fail if zone or record security does not permit this client to make changes to the zone or record, or if the client does not have ownership of the name it is trying to update. To see whether the update failed for one of these reasons, check Event Viewer on the client.

For more information about secure dynamic updates, see "Windows 2000 DNS" in the *TCP/IP Core Networking Guide*

- The client-side DNS code has a cache for performance. If record data (for example an IP address or A record) changed in the last few minutes, the TTL (Time to Live) of cached data might not have expired yet. You can run either **ipconfig/flushdns** or **net stop dnscache** to stop the cache and eliminate this as a source of problems. The preferred method is **ipconfig /flushdns**, which purges the DNS Resolver cache.

  There are two ways to disable the DNS Caching Resolver:

  - Manually disable the Caching Resolver Service by typing **net stop dnscache** at the command prompt. This disables DNS server ordering, and support for Plug and Play adapters. The end result is Windows NT 4.0–like name resolution.

Caution Do not use a registry editor to edit the registry directly unless you have no alternative. The registry editors bypass the standard safeguards provided by administrative tools. These safeguards prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your computer. Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. To configure or customize Windows 2000, use the programs in Control Panel or Microsoft Management Console (MMC) whenever possible.

  - Set to zero the REG_DWORD MaxCacheEntryTtlLimit value under HKEY_LOCAL _MACHINE\System\CurrentControlSet\ Services\DnsCache\Parameters\ that specifies maximum limit of how long the positively answered lookup is cached. This effectively eliminates caching of any resource records, but does not disable DNS server ordering and support for Plug and Play (PnP).

For more information about DNS, see "Windows 2000 DNS" in the *TCP/IP Core Networking Guide*

Questionable IP Addresses

There might be cases when you question the validity of a returned IP address after you carry out the **ipconfig** command. For example, you would question an IP address if it was 0.0.0.0, which means that a DHCP server was unavailable and that you didn't assign a static IP address.

Determining the Name Resolution Method (DNS or WINS)

Unfortunately, if **ping** failed to reach a host, it doesn't provide a specific cause for the failure. It might be either name resolution (DNS or WINS) or connectivity problems. Even if Ping succeeds, there is no guarantee that DNS or WINS supplied you with the correct IP address. It is possible that another server is using the same address. For more information about WINS name resolution, see Windows 2000 Server Help.

There are several ways to determine name resolution paths:

- If an application calls the API gethostbyname (as in the case of Internet Explorer), then DNS name resolution is attempted first, and if that fails, only then is WINS name resolution attempted. The name passed to NetBT is the "computer-name <0x00>" name, which is the same name that the command **nbtstat -a computer-name** attempts to resolve. For more information on the Nbtstat tool, see the following section.
- For file-system calls (for example, calls processed through the redirector, such as net view, net use, etc), DNS name resolution is attempted in parallel with WINS name resolution. However, the name resolved by NetBT is the "computer-name <0x20>" name. For more information on NetBT, see "Identifying NerBIOS Name Resolution Problems" later in this chapter.
- Purge and display the DNS cache and WINS cache:

  For purging the DNS cache, type **ipconfig /flushdns** at the command prompt and for purging the WINS cache, type **nbtstat -R**. Then use the **ping** command to ping a name. For displaying the DNS cache , type **ipconfig /displaydns** and for displaying the WINS cache, type **nbtstat -c**.

Identifying NetBIOS Name Resolution Problems

A simple way to verify that you have the right IP address for a specific NetBIOS name is to use a tool that displays protocol statistics and TCP/IP connections using NBT (NetBIOS over TCP/IP). The tool is called Nbtstat and is mentioned in this section.

Note Nbtstat arguments are case sensitive. For example, **nbtstat -A** lists the remote computer name table when given its IP address, and **nbtstat -a** lists the remote computer name table when given its name.

Following are examples of NetBIOS name resolution problems:

- You can ping another computer, however Nbtstat believes it is a computer other than the one that you specified. This means that there is a problem with name to address mapping. (An Nbtstat result overrules a Ping result.)
- You cannot ping another computer, and you receive a "Bad IP Address" error. This means that the name cannot be found.
- You cannot ping and you receive a "Request timed out" error. This means that either there are name resolution or connectivity problems or that the server is not functioning.

Identifying IP Addresses in the NetBIOS Remote Cache Table

Another useful command is **nbtstat -c**. This command identifies the IP addresses that are in the NetBIOS/TCP remote cache table and displays the most recent NetBIOS names that were resolved.

▶ **To identify IP addresses that are in the NetBIOS/TCP remote cache table by using Nbtstat**

- At the command prompt, type the following, and then press ENTER:

  nbtstat -c

  The **-c** option lists NetBIOS/TCP's cache of remote computer names and their IP addresses.

Table 10.2 displays the most recent NetBIOS names that have been resolved:

Table 10.2 NetBIOS/TCP Remote Cache Names

Name	Type	Host Address	Life (sec)
User2	<20> UNIQUE	172.31.228.117	60
User2	<00> UNIQUE	172.31.226.28	120
PRINT	<20> UNIQUE	172.31.64.42	600
RESKIT	<1C> GROUP	172.31.128.9	480

Important Table 10.2 shows what is in the NetBIOS/TCP remote name cache, not the DNS cache. If name resolution is through WINS then you should purge the cache.

▶ **To purge the NetBIOS/TCP remote name cache table by using Nbtstat**

- At the command prompt, type the following, and then press ENTER:

  nbtstat -R

Using Nbtstat to Validate an IP Address for a NetBIOS Name

When validating an IP address for a NetBIOS name, the command you should use is **nbtstat -A**. This option lists the remote computer name table when given its IP address.

The following procedure assumes that you already ran **ping** for a domain called RESKIT and received its IP address of 172.16.80.200.)

▶ **To validate an IP address for a NetBIOS name by using Nbtstat**

- At the command prompt, type the following and press ENTER:

  nbtstat -A *<IP Address>* (for example, 172.16.80.200)

  The **-A** subcommand lists the remote computer's name table given its IP address. Table 10.3 lists the NetBIOS remote computer names.

Table 10.3 NetBIOS Remote Computer Names

Name	Type	Status
SERVER1	<00> UNIQUE	Registered
RESKIT	<00> GROUP	Registered
RESKIT	<1C> GROUP	Registered
SERVER1	<20> UNIQUE	Registered
RESKIT	<1B> UNIQUE	Registered
RESKIT	<1E> GROUP	Registered

(continued)

Table 10.3 NetBIOS Remote Computer Names *(continued)*

Name	Type	Status
SERVER1	<03> UNIQUE	Registered
RESKIT	<1D> UNIQUE	Registered
____MSBROWSE___	<01> GROUP	Registered
INet~Services	<1C> GROUP	Registered
IS~ SERVER1	<00> UNIQUE	Registered

The **nbtstat -A** command also resolves the MAC address from the IP address.

```
MAC Address = 08-00-2B-B9-FE-7C
```

Note the case of the switch; "**-A**" lists the remote computer's name table when given its name. As previously mentioned, Ping suggested that RESKIT is at the 172.16.80.200 IP address. Similarly, the Nbtstat -A command also- suggested that the IP address for RESKIT is 172.16.80.200.

Note The command **ping -a *<IP address>*** also results in a call into NetBT to do an IP-to-name lookup similar to what **nbtstat -A *<IP address>*** does, except that only one name is printed out.

Table 10.3 provides a key for understanding the NetBIOS types mentioned in Table 10.4.

Table 10.4 Explanations of NetBIOS Types

Name	Type	Usage
00	Unique	Workstation
00	Group	Domain
01	Unique	Messenger Service
01	Group	Master Browser
03	Unique	Logon Name/Computer Name /Messenger Service
20	Unique	Server
2F	Group	Lotus Notes
33	Group	Lotus Notes
1B	Unique	Domain Master Browser
1C	Group	Domain Controllers
1E	Group	Browser Service Elections

Note You can safely ignore the group names (typically domain or workgroup names).

Identifying NetBT Problems by Using Network Monitor

Following are examples of NBT unsuccessful and successful query requests and responses. It is recommended that you monitor these requests and responses to identify name resolution problems by using NetBIOS over TCP/IP. Carry out an NBT query request by running the **nbtstat -A *<ipaddress>*** command from the command prompt.

The following is an example of a successful NBT Query request:

```
+ Frame: Base frame properties
+ ETHERNET: ETYPE = 0x0800 : Protocol = IP:  DOD Internet Protocol
+ IP: ID = 0xF421; Proto = UDP; Len: 78
+ UDP: Src Port: NETBIOS Name Service, (137); Dst Port: NETBIOS Name
Service (137); Length = 58 (0x3A)
  NBT: NS: Query req. for BOGUSNAME      <00>
      NBT: Transaction ID = 37902 (0x940E)
      NBT: Flags Summary = 0x0100 - Req.; Query; Success
          NBT: 0.............. = Request
          NBT: .0000.......... = Query
          NBT: .....0......... = Non-authoritative Answer
          NBT: ......0........ = Datagram not truncated
          NBT: .......1....... = Recursion desired
          NBT: ........0...... = Recursion not available
          NBT: .........0..... = Reserved
          NBT: ..........0.... = Reserved
          NBT: ...........0... = Not a broadcast packet
          NBT: ............0000 = Success
      NBT: Question Count = 1 (0x1)
      NBT: Answer Count = 0 (0x0)
      NBT: Name Service Count = 0 (0x0)
      NBT: Additional Record Count = 0 (0x0)
      NBT: Question Name = BOGUSNAME      <00>
      NBT: Question Type = General Name Service
      NBT: Question Class = Internet Class
```

The following is an example of an unsuccessful NBT Query response from a Network Monitor sniffer trace:

```
+ Frame: Base frame properties
+ ETHERNET: ETYPE = 0x0800 : Protocol = IP:  DOD Internet Protocol
+ IP: ID = 0xCCFF; Proto = UDP; Len: 84
+ UDP: Src Port: NETBIOS Name Service, (137); Dst Port: NETBIOS Name
Service (137); Length = 64 (0x40)
NBT: NS: Query (Node Status) resp. for BOGUSNAME      <00>, Requested
name doesn't exist
NBT: Transaction ID = 37902 (0x940E)
NBT: Flags Summary = 0x8583 - Resp.; Query; Requested name doesn't
exist
NBT: 1............... = Response
NBT: .0000........... = Query
NBT: .....1.......... = Authoritative Answer
          NBT: ......0......... = Datagram not truncated
          NBT: .......1........ = Recursion desired
          NBT: ........1....... = Recursion available
          NBT: .........0...... = Reserved
          NBT: ..........0..... = Reserved
          NBT: ...........0.... = Not a broadcast packet
          NBT: ............0011 = Requested name doesn't exist
       NBT: Question Count = 0 (0x0)
       NBT: Answer Count = 0 (0x0)
       NBT: Name Service Count = 0 (0x0)
       NBT: Additional Record Count = 0 (0x0)
       NBT: Resource Record Name = BOGUSNAME      <00>
       NBT: Resource Record Type = Null
       NBT: Resource Record Class = Internet Class
       NBT: Time To Live(Seconds) = 0 (0x0)
       NBT: RDATA Length = 0 (0x0)
```

For more information about NBT, see "Windows 2000 TCP/IP" in the *TCP/IP Core Networking Guide*.

Using Nbtstat to Determine Possible NBT Name Conflict Errors

Following are examples of possible name conflict scenarios received when running the Nbtstat tool.

- If the computer name <00> is in conflict and you receive duplicate naming error messages, the most likely cause is that there is a WORKGROUP name with the same name as the computer name. The best way to resolve this name conflict is to re-name the computer.

- If both the server name <20> and computer name <00> are in conflict, it implies that there is a computer on the network that has the same name as this computer. In this case, do the following:
  - Check which computer is in conflict, and contact the user, or rename the computer.
  - If only the server name <20> is in conflict,. check the Event Viewer for specific error messages.
- If logon server name <03> or computer name <00> are in conflict, it means that the user is logged on in more than one computer at the same time.

Missing Name Errors

The following are missing name errors along with suggestions on how to resolve them:

- If the computer name <00> is the only name missing, this is most likely the same case as for duplicate name. Check Event Viewer for redirector errors or rename the computer.
- If logon server name <03> are missing (the computer and logon names), the Messenger Service is probably not running. Check Event Viewer for error messages, and try typing **net start messenger** at the command prompt.
- If the server name <20> is missing in conjunction with the computer name <00>, it is probably the result of a name conflict. Check Event Viewer to make sure. Then rename your computer.

RPC Name Resolution Problems

RPC errors generally mean that there is a problem with either networking or name resolution. The two most common causes are either the server is down, or that the name cannot be resolved.

Note It is important to understand what name is being used for the specific RPC application. For example, Active Directory replication always refers to other domain controllers using the "guid-based name" of the domain controller. This name looks like the following:

<guid>._msdcs.*<forest-root-dns-name>*

It is recommended that you verify that this name is registered. If the target is a newly promoted domain controller, its name might not have been registered on all DNS servers. The Netdiag tool detects this when run on the target computer.

To determine whether there are name resolution problems, answer the following questions:

- Can you use the NSLookup tool to successfully query on A records and SRV records?
- Have you checked the appropriate event logs in Event Viewer for error and warning messages?

Generally, when you receive an "RPC Server not available" error message, this implies a name resolution or registration issue on the domain controller. Run the following Netdiag tool from the command prompt on both the domain controller and then on the client, as follows:

```
Netdiag /debug /fix
```

This might show some name conflicts or unregistered or unresolved names for the domain controller.

You can use the **/l** option to generate a log file. The Netdiag tool is in the Support Directory on the on the Windows 2000 Server operating system CD.

Server-based Task Errors

When you perform any of the following server-based tasks, you might receive an error that says the RPC server is unavailable:

- Replication
- Winlogon
- Enable trusted relationships
- Connect to domain controllers
- Connect to trusted domains
- User authentication

The "RPC server unavailable" error can occur for any of the following reasons:

- The RPC service is not active.
- You are unable to resolve a DNS or NetBIOS name.
- An RPC channel cannot be established.

▶ **To resolve the "RPC server is unavailable" error**

1. On the server, from the **Start** menu click **Run.**
2. Type the following line in the **Open** box:

   net start rpcss
3. Click **OK**.
4. Perform a test to determine whether you still receive an error. For example, test a connection to a domain controller. If you receive an error, continue to the next step.

   On the **Start** menu, point to **Programs** and **Accessories**, and then click **Command Prompt**. At the command prompt, type the following:

   ping <*servername*>

   where <*servername*> is the server, and NetBIOS, DNS, or GUID is the name that you want to test for connectivity. If there is a connection issue with one of these computers, contact your network administrator to resolve the issue. If the error still occurs, continue to the next step.
5. Use the Netdiag tool to determine whether the domain controller is working correctly. (You can perform a network trace by using the MSRPC, DNS, NBT, LDAP, or TCP protocols.) If there is an issue with the domain controller, contact your network administrator to resolve the error. If the error still occurs, continue to the next step.
6. Use the Netdom tool to verify network trust relationships and to reset or establish a connection to a server. If the domain controller for the domain cannot be found, the domain name is not being resolved properly. Contact your network administrator to resolve the issue. If the domain controller is found, the RPC communication channel is functioning. You can use the Netdom tool to reset or establish a connection to another server.

LDAP Verification

After you have verified that the network and DNS service are working correctly, you need to identify whether the LDAP interface is working properly.

Note The most important tool for diagnosing LDAP problems is the Ldp tool and the second most valuable tool is Network Monitor.

LDAP Diagnostic Tools

A number of tools are available to determine whether the LDAP service is available and whether it can send and receive queries.

- *Ldp*. First, there is a graphical command-line tool called Ldp. Ldp (Ldp.exe) is a graphical tool that allows users to perform LDAP operations, such as bind, search, modify, add, and delete, against any LDAP-compatible directory service, such as Active Directory. To use Ldp, install the Support Tools that are located in the Support\Tools folder on the Windows 2000 Server operating system CD. To install the tools, double-click the **Setup** icon in that folder. For more information about Ldp, see Windows 2000 Support Tools Help. Ldp can be invoked from the command prompt or, from the **Start** menu, **Run** command by typing **ldp**. It has a navigational view with a scope pane on the left, to be used for searching Active Directory, and a details pane on the right, to be used for displaying the results of the LDAP operations.

  Not all object properties stored in Active Directory are displayed by using the graphical tools that are included with the retail version of Windows 2000 Server. You can use Ldp to view these objects and their properties to assist in problem solving. Some object properties contain definitional data, called metadata, that provides information about other data that is managed within an application or environment. Ldp is valuable in that it allows you to see every object property in the directory service. You can also use Ldp to perform extended LDAP operations.

Note ADSI Edit is better suited to viewing and modifying property values because it displays the objects in a hierarchical view and allows modifications through the object properties pages.

  By using Ldp, you can perform the following LDAP functions:

  - Bind to and unbind from a domain controller.
  - Add objects to the directory.
  - Delete objects from the directory.
  - Modify object attributes.
  - Modify object relative distinguished names (RDNs).
  - Search the directory by specifying a search base and LDAP filter.
  - Compare the value of an object's attribute with a specified value.
  - Perform an extended LDAP operation.
  - View an object's security descriptor. (However, ADSI Edit is more convenient.)

- View replication metadata to identify whether objects have been updated and replicated between domain controllers. (However, the Repadmin tool is more convenient.)
- View a specific portion of the directory tree.
- View a graphical display of domains and domain controllers, including whether the domain controllers are online or offline.

For more information about using Ldp, ADSI Edit, and Repadmin, see Windows 2000 Support Tools Help.

- *Network Monitor.* Because Network Monitor is a protocol analyzer tool used to analyze and interpret network traffic off the wire, you can use this tool to capture sniffer traces of the LDAP protocol traffic. For more information about the Network Monitor tool, see the *Server Operations Guide.*
- *Netdiag.* You can use the Netdiag tool to check the different network components like LDAP, DNS, and so on. It also queries the LDAP service and ensures that it can actually connect, bind, and do a search operation against the domain controller.
- *Ntdsutil.* You can use the Ntdsutil tool to set admin limits, disconnection time-outs, and server limits. For more information about the Ntdsutil tool, see LDAP Requests for Commentss in this book.
- *ADSIEdit.* You can use the ADSIEdit MMC console to carry out LDAP operations against any of the directory partitions. If you can enable ADSIEdit to communicate to the directory, LDAP is working. Also, any of the Active Directory snap-ins can help you determine if DNS, the IP layer, and the directory service are working and available.
- *ADSI Scripts.* Finally, ADSI scripts read or write to objects in the directory. They can be used to test if the LDAP service is available.

Identifying LDAP Problems

The following sequence provides a logical pattern to diagnose and troubleshoot LDAP protocol issues. Begin by answering the following questions:

- Are you receiving any errors in the Directory Service log in Event Viewer?

  If you are having Directory access problems, the first place to check is the Directory Service log in Event Viewer. To identify directory access problems, search for NTDS LDAP error messages.
- Can an LDAP connection be established at all? Open LDP, and attempt a connection to port 389.

▶ To connect to a domain controller and view rootDSE attributes by using Ldp

1. In Ldp, on the **Connection** menu, click **Connect**.
2. In the **Server** box, either use the current domain controller name or type the name of the domain controller to which you want to connect.
3. In the **Port** box, type the port number that you want to use.

   Port 389 is the default port for LDAP; port 3268 is the default port for the Active Directory Global Catalog.
4. Click **OK**.

   The following is an example of a successful connection by using the Ldp tool:

```
d = ldap_open("SERVER1", 389);
Established connection to SERVER1.
Retrieving base DSA information...
Result <0>: (null)
Matched DNs:
Getting 1 entries:
>> Dn:
    1> currentTime: 10/18/1999 2:45:52 Pacific Standard Time Pacific
Daylight Time;
    1> subschemaSubentry:
CN=Aggregate,CN=Schema,CN=Configuration,DC=reskit,DC=com;
    1> dsServiceName: CN=NTDS
Settings,CN=SERVER1,CN=Servers,CN=Default-First-Site-
Name,CN=Sites,CN=Configuration,DC=reskit,DC=com;
    3> namingContexts: CN=Schema,CN=Configuration,DC=reskit,DC=com;
CN=Configuration,DC=reskit,DC=com; DC=reskit,DC=com;
    1> defaultNamingContext: DC=reskit,DC=com;
    1> schemaNamingContext:
CN=Schema,CN=Configuration,DC=reskit,DC=com;
    1> configurationNamingContext: CN=Configuration,DC=reskit,DC=com;
    1> rootDomainNamingContext: DC=reskit,DC=com;
    16> supportedControl: 1.2.840.113556.1.4.319;
1.2.840.113556.1.4.801; 1.2.840.113556.1.4.473;
1.2.840.113556.1.4.528; 1.2.840.113556.1.4.417;
1.2.840.113556.1.4.619; 1.2.840.113556.1.4.841;
1.2.840.113556.1.4.529; 1.2.840.113556.1.4.805;
1.2.840.113556.1.4.521; 1.2.840.113556.1.4.970;
1.2.840.113556.1.4.1338; 1.2.840.113556.1.4.474;
1.2.840.113556.1.4.1339; 1.2.840.113556.1.4.1340;
1.2.840.113556.1.4.1413;
    2> supportedLDAPVersion: 3; 2;
    11> supportedLDAPPolicies: InitRecvTimeout; MaxConnections;
MaxConnIdleTime; MaxActiveQueries; MaxNotificationPerConn;
MaxPageSize; MaxQueryDuration; MaxTempTableSize; MaxResultSetSize;
MaxPoolThreads; MaxDatagramRecv;
```

```
    1> highestCommittedUSN: 4696;
    2> supportedSASLMechanisms: GSSAPI; GSS-SPNEGO;
    1> dnsHostName: SERVER1.reskit.com;
    1> ldapServiceName: reskit.com:SERVER1$@RESKIT.COM;
    1> serverName: CN=SERVER1,CN=Servers,CN=Default-First-Site-
Name,CN=Sites,CN=Configuration,DC=reskit,DC=com;
    1> supportedCapabilities: 1.2.840.113556.1.4.800;
    1> isSynchronized: TRUE;
    1> isGlobalCatalogReady: TRUE;
-----------
```

If it fails with the DNS name, try using the IP address that the server reports that it is using, not the one that DNS reports for it.

- Are you able to ping the server?

  If a connection can not be established, the next step is to capture a sniffer trace to determine whether the server is responding at all.

- Is the server responding to other clients?
- Is there enough LDAP traffic that the server cannot keep up?

  If LDAP connections cannot be established at all, the client computer might be registered on the IP-Deny list. The Ntdsutil tool can be used to check this.

- If all of the preceding fails, determine whether other services also are failing on the server. Try **net view *server_name***.*
- Use Task Manager on the server to make sure that there is enough memory on the server and that the CPU utilization isn't reaching 100 percent.
- Increase the LDAP diagnostic logging level in the registry to level **3**, and check Event Viewer. For more information about Active Directory Diagnostic Logging, see "Advanced Troubleshooting" later in this chapter.
- Does the server respond to simple queries?

  It's not necessary to bind in order to check this. Use Ldp to establish a connection to the server. Then perform a synchronous search; leave the **Base Dn** field blank; set the filter to "(objectclass=*)"; and set the scope to "base." This is a special search of the rootDSE. This returns a list of information including the directory partitions of which this server is aware.

  If the search does not return anything, first check the event log, then get a sniffer trace and see whether the server is responding at all.

- Does the server carry out a bind?

  Because there are many ways to bind, attempt a generic security support provider interface (SSPI) bind. Try one set of credentials; if they don't work, try another set.

- Try a search after the successful bind.

  If you used administrator credentials, almost all objects in Active Directory should be visible. Other credentials result in some, or possibly most, objects not showing up in searches.

- Are you able to perform LDAP operations in the parent domain?

  If not, one probable cause is the lack of privileges because of being authenticated in the child domain.

Note There are two TCP/IP ports that are used for LDAP traffic; the regular port (389) and the Global Catalog port (3268). The Global Catalog port is enabled only when Active Directory has been installed successfully, the server becomes a domain controller, and the Global Catalog option is set. Some data is available on one port, and some on another. For example, read-only copies of data from other domains are available only from the Global Catalog port.

If all of the preceding are successful and the object of interest is still not being returned by LDAP, either the object does not exist or the credentials that are being used are not authorized to view that object. Try another set of credentials—administrator credentials are always a good test.

Confirm that the search is not hitting one of the limits on search time or number of returned objects or attributes. If limits are being hit, a paged search should solve the problem. For more information about LDAP administration limits, see "LDAP Administrative Limits and Query Policy" later in this chapter.

A review of how LDAP messages are sent, the format in which they are sent, and the supported operations can assist you in responding to these questions.

LDAP Functionality

A typical LDAP client application interacts with the LDAP server in the following ways:

- Connect to the server.
- Authenticate the client to the server.
- Modify a directory entry.
- Search the directory.
- Process search results.
- Handle errors.
- Manage memory.
- Close the connection.

Establishing a Connection

When an LDAP client connects to an LDAP server, an LDAP session is established. Options are available to affect the way in which the connection is established, such as setting a time-out value, connecting to a secure LDAP server, and verifying that a server is available.

Authenticating the Client (Binding)

The bind operation identifies the connecting person, device, or application to the server by providing a distinguished name and some type of authentication credential, such as a password. The exact credentials used depend on the authentication method being used.

Note LDAPv3 defines an extensible model based on SASL. SASL uses a layered architecture for using different security providers.

LDAPv3 allows the client to negotiate with the LDAP server to determine the best security package available. The Microsoft implementation of the LDAP API allows the NEGOTIATE flag to be used to allow the client to discover the best mechanism available, in which case basic/simple authentication is not used. For example, a SASL mechanism such as Kerberos v5 authentication or NTLM authentication might be used. An Active Directory server can be configured to accept anonymous connections.

The Windows 2000 implementation of LDAP includes these key authentication methods.

Plaintext Password This method (simple bind) authenticates by checking a plaintext password against the account password.

NTLM Authentication NTLM authentication allows clients that are running Windows NT 4.0 and earlier to authenticate themselves to LDAP servers by using NTLM. It also authenticates user logon names in a stand-alone environment.

Kerberos v5 Authentication The Kerberos authentication protocol is the default for network authentication for computers that are running Windows 2000.

Note Authentication within and between Windows 2000 domains is performed by using either the Kerberos protocol (the default method) or NTLM (for Windows NT). Other methods are available to other clients and external users connecting over the Internet.

Secure Sockets Layer (SSL) SSL is a public-domain protocol for encrypting private communications over the Internet. When a certificate infrastructure is in place, specifying server port 636 causes an SSL session to be set up. Options, methods, and functions are case-sensitive. (For more information about setting up certificates, see "Windows 2000 Certificate Services and Public Key Infrastructure" in this book.)

Simple Protected Negotiation (SPNEGO) SPNEGO enables the client and server to negotiate either through the NTLM or Kerberos v5 depending on the authentication mechanisms available to the particular client and server involved. In this case, both the server and client negotiate on a common secure authentication mechanism (for example, Kerberos authentication or NTLM authentication). This option should be used if the user cares only that the authentication mechanism is secure.

Modifying a Directory Object The LDAP API contains functions to add and delete directory objects and to compare and modify attribute values within existing objects. LDAPv3 provides extensions to the add, delete, and modify functions that enable using controls to perform these operations. Controls are described in RFC 2251 as a mechanism to extend the functionality of LDAP. Windows 2000 supports several extension controls that go beyond those identified by LDAPv3.

Searching the Directory Searching is the most common directory activity, and the LDAP APIs provide a variety of search criteria and result retrieval methods. The client searches the LDAP server by passing it a special set of parameters that describe the information in which the client is interested. These parameters describe where to search in the LDAP directory, how deep to search, and define the search criteria that a client needs. The client uses a search filter to describe the objects it wants. Search filters are defined in RFC 2254. Extensions to the base LDAP API, in the form of LDAPv3 controls, provide the ability to sort results and set various limits on the search operation. Search results can be processed by paging and by sorting. Paging and sorting are supported in Windows 2000 as new LDAPv3 control extensions for processing search results on the server.

Handling Errors All LDAP results return an error code as defined in RFC 2251. In addition, Windows 2000 domain controllers can return additional information in the form of a character string that describes the error, and the error value is translated to the closest Win32 error code.

Closing the Connection (Unbinding)

Unbinding closes the connection and disposes of the session handle. Call the unbind function when an LDAP client has finished communicating with a server. There is no server response to an unbind request.

LDAP Message Protocol Data Unit

For the purposes of protocol exchanges, all protocol operations are encapsulated in a common envelope. The *LDAPMessage* is encapsulated within the Protocol Data Unit(PDU) format. The LDAPMessage consists of protocol operations, such as LDAP Bind Request, LDAP Bind Response, LDAP Search Request, and LDAP Search Response operations. By understanding these operations, you are better able to diagnose and troubleshoot LDAP protocol issues.

LDAPMessage protocol data units are mapped directly to the TCP data stream. The LDAP ports that are used by Active Directory clients are the following:

- Port 389. In accordance with RFC 2251, Active Directory uses port 389 as the default port for domain controller communications.
- Port 636. Active Directory supports port 636 for LDAP SSL communications.
- Port 3268 and port 3269. The Global Catalog listens for LDAP communications on port 3268; it listens for LDAP SSL communications on port 3269.

For more information about LDAP operations, see the Internet Engineering Task Force link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

LDAP Bind Request

According to RFC 2251, the Bind Request has the following parameters:

- *Version*: A version number indicating the version of the protocol to be used in this protocol session. Note that there is no version negotiation, and the client sets this parameter to the appropriate version.
- *Name*: The name of the directory object that the client wants to bind. This field can take on a null value (a zero length string) for the purposes of anonymous binds, when authentication has been performed at a lower layer, or when using SASL credentials.
- *Authentication*: Information used to authenticate the name, if any, provided in the Bind Request.

When receiving a Bind Request, a server authenticates the requesting client, if necessary. The server then returns a Bind Response to the client indicating the status of the authentication.

The following is an example of an LDAP Bind Request as shown by Network Monitor:

```
LDAP: ProtocolOp: BindRequest (0)
      LDAP: MessageID = 11 (0xB)
      LDAP: ProtocolOp = BindRequest
          LDAP: Version = 3 (0x3)
          LDAP: Name =
          LDAP: Authentication Type = Sasl
                  LDAP: Sasl Mechanism = GSS-SPNEGO
                  LDAP: Sasl Credentials
```

LDAP Bind Response

An LDAP Bind Response is an indication from the server as to the status of a request for authentication of the client. If the bind is successful, the result code is "success." Otherwise, according to RFC 2251, the error is one of the following:

- *operationsError*: Server encountered an internal error.
- *protocolError*: Unrecognized version number.
- *authMethodNotSupported*: Unrecognized SASL mechanism name.
- *strongAuthRequired*: The server requires that authentication be performed with a SASL mechanism.
- *referral*: This server cannot accept this bind, and it is recommended that the client try another.
- *saslBindInProgress*: The server requires the client to send a new bind request, with the same (SASL) mechanism, to continue the authentication process.
- *inappropriateAuthentication*: The server requires that the client that had attempted to bind anonymously or without supplying credentials provide some form of credentials.
- *invalidCredentials*: The wrong password was supplied or the SASL credentials cannot be processed.
- *unavailable*: The server is shutting down.

Note The serverSaslCreds are used as part of a SASL-defined bind mechanism to allow the client to authenticate the server to which it is communicating or to perform "challenge-response" authentication.

The following is an example of an LDAP Bind Response as shown by Network Monitor:

```
LDAP: ProtocolOp: BindResponse (1)
      LDAP: MessageID = 18 (0x12)
      LDAP: ProtocolOp = BindResponse
          LDAP: Result Code = Success
          LDAP: Matched DN =
          LDAP: Error Message =
          LDAP: Sasl Mechanism = GSSAPI
          LDAP: Sasl Credentials
```

LDAP Search

A client uses the LDAP Search operation to request that a search be performed on its behalf by a server. This can be used to read attributes from a single entry, from entries immediately following a particular entry or a whole subtree of entries.

According to RFC 2251, the Search Request has the following parameters:

- *baseObject*: An LDAP distinguished name that is the base object entry relative to which the search is to be performed.
- *scope*: Indicates the scope of the search to be performed. The semantics of the possible values of this field are identical to the semantics of the scope field in the X.511 Search Operation.
- *derefAliases*: Indicates how alias objects (as defined in X.501 specification) are to be handled while searching. The semantics of the possible values of this field are:
  - *neverDerefAliases*: Do not dereference aliases while searching or while locating the base object of the search.
  - *derefInSearching*: Dereference aliases in subordinates of the base object while searching, but not while locating the base object of the search.
  - *derefFindingBaseObj*: Dereference aliases while locating the base object of the search, but not when searching subordinates of the base object.
  - *derefAlways*: Dereference aliases both when searching and when locating the base object of the search.
- *sizelimit*: Restricts the maximum number of entries to be returned as a result of the search. A value of 0 in this field indicates that no client-requested sizelimit restrictions are in effect for the search. Servers can enforce a maximum number of entries to return.
- *timelimit*: Restricts the maximum time (in seconds) allowed for a search. A value of 0 in this field indicates that no client-requested timelimit restrictions are in effect for the search.

- *typesOnly*: Indicates whether search results are going to contain both attribute types and values, or only attribute types. Setting this field to TRUE causes only attribute types (no values) to be returned. Setting this field to FALSE causes both attribute types and values to be returned.
- *filter*: A filter that defines the conditions that must be fulfilled for the search to match a specific entry.
- *attributes*: A list of the attributes to be returned from each entry that matches the search filter. There are two special values that can be used: an empty list with no attributes, and the attribute description string "*."Both of these signify that all user attributes are to be returned. (The "*" allows the client to request all user attributes in addition to specific operational attributes.)

The following is an example of an LDAP Search Request:

```
LDAP: ProtocolOp: SearchRequest (3)
      LDAP: MessageID = 1 (0x1)
      LDAP: ProtocolOp = SearchRequest
          LDAP: Base Object =
          LDAP: Scope = Base Object
          LDAP: Deref Aliases = Never Deref Aliases
          LDAP: Size Limit = No Limit
          LDAP: Time Limit = No Limit
          LDAP: Attrs Only = 0 (0x0)
          LDAP: Filter Type = Present
              LDAP: Attribute Type = objectClass
```

LDAP Search Result

The results of an LDAP Search by the server upon receipt of a Search Request are returned in Search Responses, which are LDAP messages containing either SearchResultEntry, SearchResultReference, ExtendedResponse or SearchResultDone data types.

If the LDAP session is running TCP, the server returns to the client a sequence of responses in separate LDAP messages. There might be zero or more responses containing SearchResultEntry, one for each entry found during the search.

As indicated in RFC 2251, each entry returned in a SearchResultEntry contains all attributes, complete with associated values if necessary, as specified in the attributes field of the Search Request. Return of attributes is subject to access control and other administrative policy. Some attributes might be returned in binary format (indicated by the AttributeDescription in the response having the binary option present).

For more information about LDAP Search Result, see the Internet Engineering Task Force link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

The following is an example of an LDAP Search Response as shown by Network Monitor:

```
LDAP: ProtocolOp: SearchResponse (4)
      LDAP: MessageID = 1 (0x1)
      LDAP: ProtocolOp = SearchResponse
          LDAP: Object Name =
        + LDAP: Attribute Type = currentTime
        + LDAP: Attribute Type = subschemaSubentry
        + LDAP: Attribute Type = dsServiceName
        + LDAP: Attribute Type = namingContexts
        + LDAP: Attribute Type = defaultNamingContext
        + LDAP: Attribute Type = schemaNamingContext
        + LDAP: Attribute Type = configurationNamingContext
        + LDAP: Attribute Type = rootDomainNamingContext
        + LDAP: Attribute Type = supportedControl
        + LDAP: Attribute Type = supportedLDAPVersion
        + LDAP: Attribute Type = supportedLDAPPolicies
        + LDAP: Attribute Type = highestCommittedUSN
        + LDAP: Attribute Type = supportedSASLMechanisms
          LDAP: Attribute Type = dnsHostName
```

For more information about the LDAP v3 protocol, see the Internet Engineering Task Force link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources

The following is an example of an unsuccessful LDAP Bind Response Sniffer Trace:

```
  LDAP: ProtocolOp: BindResponse (1)
      LDAP: MessageID = 8 (0x8)
      LDAP: ProtocolOp = BindResponse
          LDAP: Result Code = Invalid Credentials
          LDAP: Matched DN =
          LDAP: Error Message =
```

The following is an example of a successful LDAP Bind Response Sniffer Trace:

```
LDAP: ProtocolOp: BindResponse (1)
      LDAP: MessageID = 18 (0x12)
      LDAP: ProtocolOp = BindResponse
          LDAP: Result Code = Success
          LDAP: Matched DN =
          LDAP: Error Message =
          LDAP: Sasl Mechanism = GSSAPI
          LDAP: Sasl Credentials
```

LDAP Administrative Limits and Query Policy

LDAP administrative limits constitute the LDAP query policy, and are stored as a multivalued attribute on query policy objects. LDAP administrative limits allow you to tune working set size and CPU consumption of a particular server or set of servers based on the query workload presented. For example, a "bridgehead" server for a particular domain might disallow sorting and paged results, freeing memory, and CPU cycles to handle the intersite replication workload. A memory-rich server with limited CPU bandwidth might allow for large result sets but a small number of active queries.

Query policy applies to the following LDAP query-related operations:

- Search. An LDAP search might cover a small part of a single directory service store or span every directory service store in the forest (and beyond, through support for external cross-references). A search can generate a significant amount of disk activity, take a long time, and return a large volume of data.
- Search with Paged Results. Because a search can return a large volume of data, the client can ask the server to hold the result set and return it in "pages." The server must hold the result set until the client releases it or unbinds.
- Search with sorted results. A client can request a result set in a particular order. Sorting requires storage and CPU cycles at the server. The resources consumed are directly proportional to the size of the result set.
- Search Page size. The administrator can specify the maximum number of attribute values that can be returned per request.
- Change notify. A client can request change notification on particular objects in the directory. The mechanism used to post a change notify request is the asynchronous LDAP query.

Because server size and CPU consumption might vary, query policies need to be tested in a laboratory environment, and then managed on an individual server basis.

Configuring parameters for LDAP administrative limits can both restrict and make server resources available to clients for basic queries and queries with paged or sorted results. Also, they determine how many connections are allowed for a server, how long it can be idle, and so on. Finally, they can access to a server through an IP host address or subnet mask.

Support for LDAPv3 extensions for querying, paging, and sorting places demands on the memory and computational resources of the Active Directory server. It is prudent practice to perform load balance testing on LDAP servers before you deploy them. Only then can you develop a set of baseline measurements from which to make adjustments.

Limits can be set on the server resources that are available to clients requesting LDAP queries, paged result sets, and sorted result sets. These limits constitute the LDAP query policy, and are stored as a multivalue attribute on query policy objects. Because workload and resources of a particular server varies, the query policy is configurable at the server level.

The Ntdsutil tool can be used to view or modify the query policy of a domain controller. The Active Directory Sites and Services console can be used to assign query policies to domain controllers but not to sites. Additionally, the Modifyldap.vbs script can be used to create, delete, assign, or modify query policy objects. This script can be installed from the Support\Reskit directory on the *Windows® 2000 Resource Kit* companion CD.

Query policy objects are stored in the container cn=Query-Policies, cn=Directory Service, cn=Windows NT, cn=Services in the configuration partition.

Default Query Policy Settings

In the absence of any other assigned policies, all domain controllers use the default query policy. If a site policy is assigned, the domain controller uses this policy. If a specific policy has been assigned to a domain controller, this policy takes precedence over any site policy.

The administrative limits and values can be viewed by using the Ntdsutil command-line tool. Table 10.5 shows the administrative limits that are in effect for the default query policy.

Table 10.5 Default Query Policy Settings

LDAP Administrative Limits	Default Values	Description/Search Behavior
InitRecvTimeout	120	Initial Receive Timeout. The maximum time in seconds that the server waits for the initial request before the connection closes. If a connection is idle for more than the stated limit, the LDAP server returns a LDAP disconnect notification and closes the connection.
MaxConnections	5000	Maximum Connections. The maximum number of concurrent LDAP connections allowed on the server. If the stated limit is reached, the LDAP server and closes the connection.

(continued)

Table 10.5 Default Query Policy Settings *(continued)*

LDAP Administrative Limits	Default Values	Description/Search Behavior
MaxConnIdleTime	900	Maximum Connection Idle Time. The maximum time in seconds that the client is allowed to be idle before the connection is closed. If a connection is idle for more than the stated limit, the LDAP server closes the connection.
MaxActiveQueries	20	Maximum Active Queries. The maximum number of concurrent search operations allowed on the server. When the stated limit is reached, the LDAP server returns a busy notification.
MaxNotificationPerConn	5	Maximum Notifications per Connection. The maximum number of concurrent notification requests allowed per connection on the server. When the stated limit is reached, the server returns a busy notification.
MaxReceiveBuffer	10485760	Maximum Receive Buffer. The maximum size LDAP request in bytes that the server will attempt to process. If the server receives a request that is larger then this value, it will close the connection.
MaxPageSize	1000	Maximum Page Size. The largest page size allowed by the server. The server returns the number of rows specified by MaxPageSize. If the paged results were requested, the client can retrieve additional pages until all results are returned.
MaxQueryDuration	120	Maximum Query Duration. The maximum elapsed time (in seconds) allowed for a query to complete. If paged results are requested, the client can continue the query if the timer expires before the query completes. When the stated limit is reached, the server returns the timeLimitExceeded error.

(continued)

Table 10.5 Default Query Policy Settings *(continued)*

LDAP Administrative Limits	Default Values	Description/Search Behavior
MaxTempTableSize	10000	Maximum Temporary Table Size. The upper limit, in candidate objects, on the temporary table. If the temporary table maximum limit is reached by an "OR" query optimization, the optimization is abandoned and replaced with a direct table scan. This limit can also be reached when the server sorts (for example, by the server side sort control,) results for the client. If the server reaches this limit while sorting results it will abandon the sort and return results unsorted.
MaxResultSetSize	262144	Maximum Result Set Storage. The maximum storage that the server can hold for all paged result sets. If the stated limit is reached, the oldest result sets are discarded.
MaxPoolThreads	4	The number of threads per processor allocated to answer LDAP requests. This value can be exceeded by the server only under certain circumstances Note: If it takes a long time to bind, increase the count to 6 or 8.
MaxDatagramRecv	1024	Maximum Datagram Receive. The maximum size of datagrams that can be received by the server. The server pre-allocates datagram buffers and cannot receive datagrams with a size larger than the stated limit.

For more information about using Ntdsutil, see the Support directory on the Windows 2000 Server operating system CD. For more information about virtual containers, see "Active Directory Data Storage" in this book.

Domain Controller Issues

Among the most important features of Windows 2000 include the facts that all domain controllers in the same domain are peers of one another and *any* domain controller can make directory updates.

However, given the way in which directory updates are replicated from one domain controller to another, it is possible that difficulties can arise. For example, if the necessary domain controllers are not connected by a replication topology, the appropriate domain controllers do not receive directory updates when replication occurs.

Also, in order for the (Domain Controller) Locator to find a domain controller, it must have accurate information so that it can properly locate the resource. If a domain controller is incorrectly advertised, the Locator is not going to find it.

Note In addition to the DNS and NetBIOS broadcast being used to find servers, each server must be "advertising" a role in order for the locator to return that server as a candidate. You can use the Nltest tool to show what roles are being advertised. Furthermore, a server does not advertise itself in some roles until it has finished initializing. Thus, if a server is stuck or having problems starting, it might be excluded from the list of available servers, making the other servers more heavily loaded. If a server runs out of disk space, it stops advertising itself as an LDAP server.

Also be aware that FRS might prevent a computer from advertising.

This section discusses diagnostic tools and gives examples of possible domain controller consistency problems, along with suggested solutions.

Event Viewer

In Event Viewer, there is a separate directory service log for the all the directory events that are written to it. For example, domain controller consistency problems might be manifested in events such as Internal Processing, Inter-Site Messaging, Service Control, and Internal Configuration.

For information about the replication schedule of directory partitions, use Event Viewer, and increase the Replication Events logging level to level 2. You can adjust the logging level in the registry by changing the value of entries in the HKEY_LOCAL_MACHINE\ SYSTEM\CurrentControlSet\Services\NTDS \Diagnostics subkey.

Note You should check the event log first and not raise the logging level until you understand the problem and what you are looking for.

It is not recommended that you set the diagnostic level of Replication Events higher than 2. The user can be inundated with detail, especially for replication events.

For more information about adjusting Active Directory log levels, see "Advanced Troubleshooting" later in this chapter. Do not modify the registry until you have read this section.

Using Dcdiag to Diagnose Domain Controller Issues

The Domain Controller Diagnostic tool (Dcdiag) analyzes the state of domain controllers in a forest or enterprise and reports any problems. The tool is designed to be an end-user reporting program that encapsulates the detailed knowledge of how to identify abnormal behavior in the computer. The area of focus of this tool is domain controller functions and interactions across an entire enterprise.

Dcdiag consists of a framework for running tests, and a series of tests to verify different functional areas of Active Directory. The framework selects which domain controllers are tested according to scope directives given by the user, such as enterprise, site, or single server. The user can also select domain controllers holding a directory partition.

It is recommended that only severe errors be reported, and that they be reported in a way that informs the user of the consequences of the problem, and also suggests a course of action for the user. In the default mode, minimum output is displayed —successful confirmation of each test. In the verbose mode, the collected data for each test displays.

Note Note that Dcdiag is intended to perform a fully automatic analysis with little user intervention. It is essentially a read-only tool that does not affect the state of the enterprise. Although it allows specific tests to be run individually, it is not intended as a general toolbox of commands to perform specific tasks.

Use the Dcdiag tool to diagnose domain controller status for the following:

- Connectivity
- Replication
- Topology Integrity
- Directory Partition Head Permissions
- User Permissions
- Locator Functionality
- Inter-site Health
- Trust Verification
- Diagnose Replication Latencies
- Replication of Trust Objects
- File Replication Service
- Critical Services Check

Connectivity

To test for domain controller connectivity, use the Dcdiag tool to do the following:

- Verify that the DNS names for the server are registered.
- Verify that the server can be reached by means of IP at its IP address.
- Verify that the server can be reached by means of LDAP.
- Verify that the server can be reached by means of an RPC call.

Replication

To test for domain controller replication consistency, use the Dcdiag tool to do the following:

- Report any replication errors on incoming replica links to this computer. Normal errors, such as those generated because the source is deleted or a new source is added, are filtered out appropriately.
- Report if replications are late in being performed.
- Check if replication is disabled.

Topology Integrity

To test for domain controller topology integrity, use the Dcdiag tool to verify that all servers holding a specific directory partition are connected by the replication topology.

Directory Partition Head Permissions

Use the Dcdiag tool to test that the security descriptors on the directory partition heads, such as the Schema, Domain, or Configuration directory partitions, for the proper permissions.

User Permissions

To ensure that users have the necessary permissions, use the Dcdiag tool to do the following:

- Check that the necessary users have the proper network logon permissions to allow replication to proceed.
- Check for Authenticated Users.

Locator Functionality

To ensure that the Domain Controller Locator is properly functioning, use the Dcdiag tool to do the following:

- Verify that each server is being advertised to the (Domain Controller) Locator.

- Verify that the roles returned by the Locator for the computer match the roles for which that computer is capable.
- Verify that the server recognizes and can communicate with global role holders (operations masters).
- Verify that the Locator can find a Global Catalog server for the enterprise.
- Verify that the Locator can find a primary domain controller for the enterprise.

Inter-site Health

To ensure consistency of domain controllers among sites, use the Dcdiag tool to do the following:

- Identify the Inter-site Topology Generator for each site.
- Identify bridgeheads for a site and generate a bridgehead status report to determine which ones are not functioning.
- In the case where bridgeheads are not functioning, locate additional backup bridgeheads. Report how long it is going to be until a failed bridgehead is failed-over. Fail-over means that if a bridgehead server unexpectedly goes down, another delegated or preferred bridgehead server eventually takes the place of that bridgehead server.
- Identify which sites are not communicating with other sites in the network topology.

For more information about Inter-site Topology Generator, bridgeheads, and bridgehead failovers, see "Active Directory Replication" in this book.

Trust Verification

To check for trust verification, the recommended method is to use the Netdom tool. However, the Dcdiag tool can also be used to check explicit trust relationships. A trust verification is between two domains that enumerates all of the domain controllers in each domain. You can optionally scope this verification by site or by domain controller. You can check trust establishment, the secure channel setup, and ticket validity between each pair of domain controllers. By default, errors are flagged. In verbose mode, all of the successes are printed as well.

Note The Dcdiag tool only checks explicit trust relationships; it does not check Kerberos v5 trust relationships. To check the Kerberos v5 trust relationships, you would use the Netdom tool. For more information on the Netdom tool and how to check the Kerberos v5 trust relationships, see "Join and Authentication Issues" later in this chapter.

If the trust relationship fails between every pair of domain controllers, there is a very high probability that the problem is with the trust relationship. In this case, use the Nltest tool to further isolate the failure (for example, use the **/sc_query** and **/sc_reset** switches) and the Net Logon log to further investigate the problem.

Note The problem can be usually be resolved by recreating the trust relashionship through the Active Directory Domains and Trusts console.

If only a few pairs of domain controllers are experiencing the trust relationship problem and other pairs are not, it could be a replication or name resolution–related problem. In this case, check whether the trusted domain objects (in the System container) are up-to-date on all domain controllers.

For more information about trusted domain objects, see "Active Directory Logical Structure" in this book.

For each server that has a broken secure channel, the server's name is printed out along with a Win32 error message indicating the reason why the secure channel is not working. For each error, the next step is to examine the domain controller that is having the trouble—most likely the error is network connectivity based.

Following is an example of a secure channel failure while running the Dcdiag tool.

```
F:> dcdiag /v /s:dc5/test:outboundsecurechannels /testdomain:washington
/nositerestriction
DC Diagnosis

Performing initial setup:
   * Connecting to directory service on server dc5.
   * Collecting site info.
   * Identifying all servers.
   * Found 20 DC(s). Testing 1 of them.
   Done gathering initial info.

Doing initial non skippeable tests

   Testing server: Building1\DC5
      Starting test: Connectivity
         * Active Directory LDAP Services Check
         * Active Directory RPC Services Check
         ..................... DC5passed test Connectivity
```

```
Doing primary tests

   Testing server: Building1\DC5
      Test omitted by user request: Replications
      Test omitted by user request: Topology
      Test omitted by user request: NCSecDesc
      Test omitted by user request: NetLogons
      Test omitted by user request: LocatorGetDc
      Test omitted by user request: RidManager
      Test omitted by user request: MachineAccount
      Test omitted by user request: Services
      Starting test: OutboundSecureChannels
         * Secure channel from [DC-08] to [\\RED-DC-
11.washington.corp.micros
oft.com] is working properly.
         * [DC-08] has downlevel trust object for [washington]
         * [DC-08] has uplevel trust object for [washington]
         * Secure channel from [DC-07] to [\\RED-DC-
01.washington.corp.micros
oft.com] is working properly.
         * [DC-07] has downlevel trust object for [washington]
         * [DC-07] has uplevel trust object for [washington]
         * Secure channel from [NTDSDCB] to [\\RED-DC-
08.washington.reskit.com.
com] is working properly.
         * [NTDSDCB] has downlevel trust object for [washington]
         * [NTDSDCB] has uplevel trust object for [washington]
         [NTDSDC] LDAP connection failed with error 58,
         The specified server cannot perform the requested operation..
         [NTDSDC] LDAP bind failed with error 31. A device attached to
the system is not functioning.
* Secure channel from [DC5] to [\\RED-DC-12.washington.reskit.com.
com] is working properly.
         * [DC5] has downlevel trust object for [washington]
         * [DC5] has uplevel trust object for [washington]
         * Secure channel from [DC1] to [\\RED-DC-
03.washington.reskit.com.
com] is working properly.
         * [DC1] has downlevel trust object for [washington]
         * [DC1] has uplevel trust object for [washington]
         * Secure channel from [DC9] to [\\RED-DC-
07.washington.reskit.com.
com] is working properly.
         * [DC9] has downlevel trust object for [washington]
         * [DC9] has uplevel trust object for [washington]
```

```
         * Secure channel from [DCG] to [\\RED-DC-
08.washington.reskit.com.
com] is working properly.
         * [DCG] has downlevel trust object for [washington]
         * [DCG] has uplevel trust object for [washington]
         * Secure channel from [DC2] to [\\RED-DC-
06.washington.reskit.com.
com] is working properly.
         * [DC2] has downlevel trust object for [washington]
         * [DC2] has uplevel trust object for [washington]
         ..... NTDSDC failed test
OutboundSecureChannels
      Test omitted by user request: ObjectsReplicated

   Running enterprise tests on : reskit.com
      Test omitted by user request: Intersite
      Test omitted by user request: RolesHeld
```

In this example, NTDSDC is down.

For a specific secure channel problem, you might see the following:

```
* Secure channel from [DC5] to washington is working because "The RPC
server is unavailable."
```

In this case, it is recommended that the administrator run diagnostics on [DC5] to see whether it is having network problems.

Diagnose Replication Latencies

The checks are as follows:

- Check the status of a specific source partner for a destination, The test also checks that the source partner has a notification link back to that destination.

  For more information about notification links, see "Active Directory Replication" in this book.
- Analyze a particular incoming replication link for occurrences of zero failures if the time since its last success is unusually long. This means that the replication link is being delayed or preempted because of higher priority work.
- Report if the updated sequence number (USN) vector, which the destination keeps for a particular source partner, indicates that a full synchronization is in progress. This is not a failure, although it does indicate that new changes from that partner are delayed until the full synchronization process has completed.

- Check the queue of current and pending replication activities for indications of delay. There are three specific areas to investigate:
  - First, a replication job is taking a long time when there are no higher priority tasks waiting. This is not a failure, although, this could mean that the computer is not up to date. New changes from that source are delayed until the computer catches up.
  - Second, a replication job is taking a long time when there are higher priority tasks waiting. Theoretically, this can only happen until the current call completes, when the replication dispatcher causes the higher priority task to run. In practice, this can indicate either a stuck call at the server or a replication call that does not have a server-side time limit.
  - Third, look at the number of pending replication tasks. A large number means that the computer was delayed in the past, and a large number of replications requests are waiting.

Replication of Trust Objects

This option checks the following:

- Check that the computer account object has replicated to all additional domain controllers of the domain. Verification is done by comparing the object attribute metadata of all copies of the object.
- Verify that the DSA object has replicated to all replicas of the configuration directory partition.

File Replication Service

Verify that File Replication service (FRS) has started successfully on all servers. If FRS has not started, it delays the Net Logon service from advertising that domain controller.

Critical Services Check

Verifies that critical services are running on each domain controller. The services that are checked include: File Replication service, Intersite Messaging Service, Kerberos v5 Key Distribution Center Service, Server Service, Workstation Service, Remote Procedure Call Locator Service, Windows Time Service, Distributed Link Tracking Client Service, Distributed Link Tracking Server Service and the Net Logon service.

Sample output of Dcdiag.exe running all the previous tests in verbose mode:

```
C:\DS TOOLS>dcdiag /s:SERVER1 /c /v

DC Diagnosis

Performing initial setup:
   * Connecting to directory service on server SERVER1.
   * Collecting site info.
   * Identifying all servers.
   * Found 1 DC(s). Testing 1 of them.
   Done gathering initial info.

Doing initial non skippeable tests

   Testing server: Default-First-Site-Name\SERVER1
      Starting test: Connectivity
         * Active Directory LDAP Services Check
         * Active Directory RPC Services Check
         ..................... SERVER1 passed test Connectivity

Doing primary tests

   Testing server: Default-First-Site-Name\SERVER1
      Starting test: Replications
         * Replications Check
         ..................... SERVER1 passed test Replications
      Starting test: Topology
         * Configuration Topology Integrity Check
         * Analyzing the connection topology for
CN=Schema,CN=Configuration,DC=f
oobar,DC=com.
         * Performing upstream (of target) analysis.
         * Performing downstream (of target) analysis.
         * Analyzing the connection topology for
CN=Configuration,DC=reskit,DC=c
om.
         * Performing upstream (of target) analysis.
         * Performing downstream (of target) analysis.
         * Analyzing the connection topology for DC=reskit,DC=com.
         * Performing upstream (of target) analysis.
         * Performing downstream (of target) analysis.
         ..................... SERVER1 passed test Topology
```

```
      Starting test: CutoffServers
         * Configuration Topology Aliveness Check
         * Analyzing the alive system replication topology for
CN=Schema,CN=Conf
iguration,DC=reskit,DC=com.
         * Performing upstream (of target) analysis.
         * Performing downstream (of target) analysis.
         * Analyzing the alive system replication topology for
CN=Configuration,
DC=reskit,DC=com.
         * Performing upstream (of target) analysis.
         * Performing downstream (of target) analysis.
         * Analyzing the alive system replication topology for
DC=reskit,DC=com.

         * Performing upstream (of target) analysis.
         * Performing downstream (of target) analysis.
         ..... SERVER1 passed test CutoffServers
      Starting test: NCSecDesc
         * Security Permissions Check for
           CN=Schema,CN=Configuration,DC=reskit,DC=com
         * Security Permissions Check for
           CN=Configuration,DC=reskit,DC=com
         * Security Permissions Check for
           DC=reskit,DC=com
         ..... SERVER1 passed test NCSecDesc
      Starting test: NetLogons
         * Network Logons Privileges Check
         ..... SERVER1 passed test NetLogons
      Starting test: LocatorGetDc
         Role Schema Owner = CN=NTDS
Settings,CN=SERVER1,CN=Servers,CN=Default-F
irst-Site-Name,CN=Sites,CN=Configuration,DC=reskit,DC=com
         Role Domain Owner = CN=NTDS
Settings,CN=SERVER1,CN=Servers,CN=Default-F
irst-Site-Name,CN=Sites,CN=Configuration,DC=reskit,DC=com
         Role PDC Owner = CN=NTDS
Settings,CN=SERVER1,CN=Servers,CN=Default-Firs
t-Site-Name,CN=Sites,CN=Configuration,DC=reskit,DC=com
         Role Rid Owner = CN=NTDS
Settings,CN=SERVER1,CN=Servers,CN=Default-Firs
t-Site-Name,CN=Sites,CN=Configuration,DC=reskit,DC=com
         Role Infrastructure Update Owner = CN=NTDS
Settings,CN=SERVER1,CN=Serve
rs,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=reskit,DC=com
         ..... SERVER1 failed test LocatorGetDc
```

```
      Starting test: RidManager
         * Available RID Pool for the Domain is 1603 to 1073741823
         * SERVER1.reskit.com is the RID Master
         * DsBind with RID Master was successful
         * rIDAllocationPool is 1103 to 1602
         * rIDNextRID: 1106
         * rIDPreviousAllocationPool is 1103 to 1602
         ..... SERVER1 passed test RidManager
      Starting test: MachineAccount
         * SPN found :LDAP/SERVER1.reskit.com/reskit.com
         * SPN found :LDAP/SERVER1.reskit.com
         * SPN found :LDAP/SERVER1
         * SPN found :LDAP/SERVER1.reskit.com/RESKIT1
         * SPN found :LDAP/6cbd730e-b9ce-4154-8367-
45a8b469097b._msdcs.reskit.co
m
         * SPN found :E3514235-4B06-11D1-AB04-00C04FC2DCD2/6cbd730e-
b9ce-4154-83
67-45a8b469097b/reskit.com
         * SPN found :HOST/SERVER1.reskit.com/reskit.com
         * SPN found :HOST/SERVER1.reskit.com
         * SPN found :HOST/SERVER1
         * SPN found :HOST/SERVER1.reskit.com/RESKIT1
         * SPN found :GC/SERVER1.reskit.com/reskit.com
         ..... SERVER1 passed test MachineAccount
      Starting test: Services
         * Checking Service: Dnscache
         * Checking Service: NtFrs
         * Checking Service: IsmServ
         * Checking Service: kdc
         * Checking Service: SamSs
         * Checking Service: LanmanServer
         * Checking Service: LanmanWorkstation
         * Checking Service: RpcSs
         * Checking Service: RPCLOCATOR
         * Checking Service: w32time
         * Checking Service: TrkWks
         * Checking Service: TrkSvr
         * Checking Service: NETLOGON
         * Checking Service: Dnscache
         * Checking Service: NtFrs
         ..... SERVER1 passed test Services
      Starting test: OutboundSecureChannels
         ** Did not run test because /testdomain: was not entered
......
.................. SERVER1 passed test OutboundSecureChannels
```

```
      Starting test: ObjectsReplicated
         SERVER1 is in domain DC=reskit,DC=com
         Checking for CN=SERVER1,OU=Domain Controllers,DC=reskit,DC=com
in domai
n DC=reskit,DC=com on 1 servers
            Object is up-to-date on all servers.
         Checking for CN=NTDS Settings,CN=SERVER1,CN=Servers,CN=Default-
First-Si
te-Name,CN=Sites,CN=Configuration,DC=reskit,DC=com in domain
CN=Configuration,DC
=reskit,DC=com on 1 servers
            Object is up-to-date on all servers.
         ......................... SERVER1 passed test ObjectsReplicated
      Starting test: frssysvol
         * The File Replication Service Event log test
         The SYSVOL has been shared, and the AD is no longer
         prevented from starting by the File Replication Service.
         ......................... SERVER1 passed test frssysvol

   Running enterprise tests on : reskit.com
      Starting test: Intersite
         ......................... reskit.com passed test Intersite
      Starting test: RolesHeld
         GC Name: \\SERVER1.reskit.com
         Locator Flags: 0xe00001fd
         PDC Name: \\SERVER1.reskit.com
         Locator Flags: 0xe00001fd
         Time Server Name: \\SERVER1.reskit.com
         Locator Flags: 0xe00001fd
         Preferred Time Server Name: \\SERVER1.reskit.com
         Locator Flags: 0xe00001fd
         KDC Name: \\SERVER1.reskit.com
         Locator Flags: 0xe00001fd
         ......................... reskit.com passed test RolesHeld
```

Using Ntdsutil to Manage Domain Controller Consistency

Ntdsutil is a command-line tool that provides directory service management. It maintains the Active Directory store, manages and controls Flexible Single Master Operations master, and purges metadata left behind by abandoned domain controllers (which are removed from the network without being uninstalled). For more information about using Ntdsutil, see "Active Directory Diagnostic Tool (Ntdsutil.exe)" in this book.

By using Ntdsutil, you can diagnose and troubleshoot the following domain controller consistency-related issues:

- Remove orphaned domain controllers and domains.

  Note Netdom can also remove orphaned domains. For more information about removing orphaned domain controller, see "Active Directory Instsallation and Removal" later in this chapter.

- Connect to a specific domain controller.

  View directory partitions, sites, servers, domains, and operations master roles.
- View and set the values for the LDAP policies supported on a server.
- Manage operations master roles. (For more information about managing operations master roles, see "Managing Flexible Single-Master Operations" in this book.)

Identifying Windows 2000 Domain Controller Roles

There might be instances when you need to identify which domain controller holds the primary domain controller operations master role in a domain so that clients that are running earlier versions of Windows NT can be authenticated.

Note Clients running earlier versions of Windows NT can be authenticated at any domain controller. Unavailability of the PDC emulator prevents these clients from joining computers to the domain or changing their user password among other options.

Also, you might need to identify which domain controllers are Global Catalog servers so that you can verify that LDAP Search requests can be satisfied in the forest. Use the following methods to identify Windows 2000 domain controllers:

- The NTDS registry subkey appears in the HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services registry subkey.
- The SYSVOL and NETLOGON shares exist. For more information about SYSVOL and NETLOGON shares, see "File Replication Service" in this book. (The SYSVOL share and its contents exist after removing Active Directory.)
- By using the **nbtstat** command-line tool, you can check domain name registration. It shows that the 1C name (Domain) is registered. Type **nbtstat -n** at the command prompt, and note the presence of the 1C name.
- The computer role from the Net Accounts tool lists the computer role as "PRIMARY" and stand-alone servers as "SERVERS." Type **net accounts** at the command prompt.
- The **net start** command indicates that the Kerberos KDC service is running. Type **net start |more** at the command prompt for additional information.

- By using the **Connect to server %S** command in the Ntdsutil tool, you can connect to other Windows 2000–based domain controllers. (Note that Ntdsutil functions only with Windows 2000–based domain controllers.) The computer responds to LDAP queries (specifically, to port 389 or port 3289).
- The **Change** button on the **Network Identification** tab in **My Computer** is disabled when a Windows 2000–based server is configured as a domain controller. A note appears indicating this fact. (Domain controllers cannot be renamed. However, domain member and stand-alone computers can be renamed.)
- To identify the domain controller that holds the primary domain controller role for a domain, by running the Netdiag tool and observing the "Machine is a Primary Domain Controller" entry in the output. Type **netdiag /v** at the command prompt. Also, you can use the Nltest tool to obtain the same information, as shown in the following example:

```
nltest /dsgetdc:reskit /pdc
           DC: \\NTDSDC4
      Address: \\172.23.92.85
     Dom Guid: ca21b03b-6dd3-11d1-8a7d-b8dfb156871f
     Dom Name: RESKIT
  Forest Name: reskit.reskit.com.
 Dc Site Name: Red-Bldg26
Our Site Name: Red-Bldg26
        Flags: PDC DS KDC TIMESERV WRITABLE DNS_FOREST CLOSE_SITE 0x8
The command completed successfully
```

- To identify the domain controller that is also designated as the Global Catalog server for the forest, you can either examine the Global Catalog check box in the Active Directory Sites and Services console or by running the Nltest tool, check whether the Global Catalog flag is returned.

```
E:\nltest /dsgetdc:server1.reskit.com /gc
           DC: \\FE-DC-02.fareast.reskit.com.com
      Address: \\172.23.4.194
     Dom Guid: 0502fd7a-2b1e-11d3-a5ec-00805f9f21f5
     Dom Name: fareast.reskit.com.com
  Forest Name: reskit.com.com
 Dc Site Name: Default-First-Site-Name
        Flags: PDC GC DS LDAP KDC TIMESERV WRITABLE DNS_DC DNS_DOMAIN
DNS_FOREST
```

- To identify the FSMO roles. Through the Active Directory Users and Computers console, you must be able to select operation masters and it is going to show the holders of the three roles—PDC, RID, and Infrastructure.

Advertising as a Global Catalog Server

A domain controller does not advertise itself as a global catalog until it has replicated *in* the required domains. The following standards for Global Catalog promotion:

- There is a distinction between requesting that a computer be elected as a Global Catalog server, and that computer actually finishing promotion and advertising as a Global Catalog server. The server must successfully replicate in read only copies of the domains in the enterprise before that server will advertise as a Global Catalog. The way you request a domain controller to become a Global Catalog is to check the **Global Catalog** box in the Active Directory Sites and Services console.

Note Even though this box is checked does not necessarily imply that the computer has successfully become a Global Catalog and is advertising itself.

There are four ways to determine if a computer is advertising as a Global Catalog:

- Look in the Directory Service log in Event Viewer for a message indicating that the computer is advertising.
- Use the Ldp tool to view the *isGcReady* attribute from the rootDSE. When this is true, the computer is a Global Catalog server and is advertising itself.
- Use the Nltest tool to determine that the computer has the Global Catalog attribute set. If it does, then it is a Global Catalog server.
- Verify if the Global Catalog Promotion Complete registry entry stored in HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS \Parameters\ is set to the value 1. If it is, then the computer is advertising as a Global Catalog server.

- If a computer is successfully advertised as a Global Catalog and restarted, it immediately advertises itself even if there are additional domains in the enterprise that the Global Catalog doesn't have yet. This can be referred to as a "grandfather clause," which implies that after a computer is a Global Catalog, the Global Catalog is not diabled even if it doesn't hold all the domains in its list.
- A computer with the Global Catalog check box is selected, retries periodically (every 30 minutes) to see if it holds all the domains. You can decrease this time period by setting a registry key that is mentioned in the event log message.
- The default requirement is that the Global Catalog must hold copies of all domains that have a source in the Global Catalog's site. Thus if the Global Catalog is in site1, and there exist domain controllers for domains A, B, and C in site1, and domain controllers in domains D, E, and F in site2, then the Global Catalog must hold copies of A, B, and C before it advertises.

- Rebooting a computer that is trying to become a Global Catalog doesn't alter the behavior. When it restarts, it continues trying to become a Global Catalog.

Note During dcpromo, after a certain point, the user has the option of **finish replication later**. If this is selected and the computer rebooted, the system does not advertise until the first full synchonziation of the domain has occurred. Whether the computer considers itself synchronized can be tested by using the RootDSE attribute *isSynchronized*. This can be examined using Ldp.exe.

Using Dsastat to Detect Directory Partition Differences

If you want to examine the differences amongst a user-defined scope of objects on two different domain controllers, use the Dsastat tool.

The Dsastat command-line tool compares and detects differences between directory partitions on domain controllers. It retrieves capacity statistics such as megabytes per server, objects per server, and megabytes per object class. Then, it compares the attributes of replicated objects. It can be used to compare two directory trees across replicas within the same domain or, in the case of a Global Catalog, across different domains. You can use this to monitor replication status at a much higher level than monitoring detailed transactions.

Note The Dcdiag tool contains an option called "check objects" that analyzes and confirms that all copies of a server's computer account objects and a server's DSA objects are consistent. In general, if replication is up-to-date, all copies are consistent and there is no need for a detecting differences of all the copies. This is only needed if you suspect database corruption. If you have different views of your data, the most likely reason is replication failure. The Dcdiag "replication" test tells you about any replication failures.

For example, to perform a comparison of all users in the Sales organizational unit in the Reskit.com domain, with those in another directory partition, specify the following:

```
dsastat -s:reskitS1;reskitS2 -b:OU=Sales,DC=Reskit,DC=com -gcattrs:all -
sort:true -t:false -p:16 -
filter:"(&(objectclass=user)(!objectClass=computer))"
```

In this example you can determine whether both domain controllers agreed on the contents of the OU=Sales,DC=Reskit,DC=com subtree. It detects objects in one and not the other (for example, if a creation or deletion has not replicated) as well as differences in the values on objects that do exist on both.

This example specifies a base search path at a subtree of the domain. In this case, the organizational unit name is "Sales." The filter specifies that the comparison is concerned only with user objects, not computer objects.

Note Because computer objects are derived from user objects in the class hierarchy, a search filter specifying "objectclass = user" returns both user and computer objects.

Also, using the Dsastat tool, you can specify the target domain controllers and additional operational parameters from the command line or from an initialization file. The Dsastat tool determines whether domain controllers in a domain have a consistent and accurate image of their own domain. In the case of Global Catalogs, it checks whether the Global Catalog server has an image that is consistent with the domain controllers in other domains. It complements the other replication-monitoring tools, Repadmin and Replmon, by ensuring that domain controllers are up to date with one another.

Determining if Domain Controllers are Up To Date

If you see the error "DS paths have a different object count in them" in the Directory Service log of Event Viewer, you would use Dsastat, Repadmin, and Replmon to diagnose and resolve the problems.

For example:

```
LDAP::<DCName>.reskit.com/CN=Packages,CN=Class Store,CN={EF06ECF2-A8C9-
11D2-B575-0008C7457B4E},CN=Policies,CN=System,
DC=reskit,DC=microsoft,DC=com

For DCName=ntdsdc4 there are 77 objects in the tree while for
DCName=RESKIT-DC-08 there are 78 objects. The missing object is
CN={7cc10d6e-463f-4a65-8d4d-56d85fc823c1}
```

Resolution to the problem:

The object was created by dc1 about 4 P.M.:

```
C:\>repadmin /showmeta "CN=7cc10d6e-463f-4a65-8d4d-
56d85fc823c1,CN=Packages,CN=Class Store,CN=User,CN={EF06ECF2-A8C9-11D
2-B575-
0008C7457B4E},CN=Policies,CN=System,DC=reskit,DC=microsoft,DC=com"
reskit-dc-08

29 entries.
```

```
Loc.USN                    Originating DSA Org.USN       Org.Time/Date
Ver Attribute
=======                    =============== =======       =============
=== =========
12950240                 Bldg\DC1 7611643 1999-06-18 15:58.37    1
objectClass
12950240              Bldg\RESKIT-DC-0812950240 1999-06-18 16:14.59
1 cn
12950240                 Bldg\DC1 7611643 1999-06-18 15:58.37    1
instanceType
12950240                 Bldg\DC1 7611643 1999-06-18 15:58.37    1
whenCreated
12950240                 Bldg\DC1 7611643 1999-06-18 15:58.37    1
showInAdvancedViewOnly
12950240                 Bldg\DC1 7611643 1999-06-18 15:58.37    1
nTSecurityDescriptor
12950240                 Bldg\DC1 7611643 1999-06-18 15:58.37    1
name
12950240                 Bldg\DC1 7611643 1999-06-18 15:58.37    1
msiScriptPath
12950240                 Bldg\DC1 7611643 1999-06-18 15:58.37    1
cOMClassID
12950240                 Bldg\DC1 7611643 1999-06-18 15:58.37    1
cOMProgID
12950240                 Bldg\DC1 7611643 1999-06-18 15:58.37    1
localeID
12950240                 Bldg\DC1 7611643 1999-06-18 15:58.37    1
computerArchitecture
12950240                 Bldg\DC1 7611643 1999-06-18 15:58.37    1
revision
12950240                 Bldg\DC1 7611643 1999-06-18 15:58.37    1
packageType
12950240                 Bldg\DC1 7611643 1999-06-18 15:58.37    1
packageName
12950240                 Bldg\DC1 7612100 1999-06-18 16:01.02    2
packageFlags
12950240                 Bldg\DC1 7611643 1999-06-18 15:58.37    1
versionNumberHi
12950240                 Bldg\DC1 7611643 1999-06-18 15:58.37    1
versionNumberLo
12950240                 Bldg\DC1 7612100 1999-06-18 16:01.02    3
lastUpdateSequence
12950240                 Bldg\DC1 7611643 1999-06-18 15:58.37    1
msiFileList
12950240                 Bldg\DC1 7611643 1999-06-18 15:58.37    1
categories
12950240                 Bldg\DC1 7611643 1999-06-18 15:58.37    1 url
```

```
12950240                    Bldg\DC1 7611643 1999-06-18 15:58.37    1
objectCategory
12950240                    Bldg\DC1 7611643 1999-06-18 15:58.37    1
upgradeProductCode
12950240                    Bldg\DC1 7611643 1999-06-18 15:58.37    1
canUpgradeScript
12950240                    Bldg\DC1 7611643 1999-06-18 15:58.37    1
fileExtPriority
12950240                    Bldg\DC1 7611643 1999-06-18 15:58.37    1
productCode
12950240                    Bldg\DC1 7612100 1999-06-18 16:01.02    2
msiScriptName
12950240                    Bldg\DC1 7611643 1999-06-18 15:58.37    1
installUiLevel
```

Taking in to consideration the latencies in reskit.microsoft.com (computers being restarted, upgrades, new software installation, and so on), it might take more than an hour for a change to replicate.

The following example shows that the change has finally replicated:

```
C:\>repadmin /showmeta "CN=7cc10d6e-463f-4a65-8d4d-
56d85fc823c1,CN=Packages,CN=Class Store,CN=User,CN={EF06ECF2-
2-B575-
0008C7457B4E},CN=Policies,CN=System,DC=reskit,DC=microsoft,DC=com"
ntdsdc4

29 entries.

Loc.USN                     Originating DSA Org.USN       Org.Time/Date
Ver Attribute
=======                     =============== =======       =============
=== =========
7597742                   Bldg\DC1 7611643 1999-06-18 15:58.37    1
objectClass
7597742                   Bldg\DC4 7597742 1999-06-18 16:17.19    1 cn
7597742                   Bldg\DC1 7611643 1999-06-18 15:58.37    1
instanceType
7597742                   Bldg\DC1 7611643 1999-06-18 15:58.37    1
whenCreated
7597742                   Bldg\DC1 7611643 1999-06-18 15:58.37    1
showInAdvancedViewOnly
7597742                   Bldg\DC1 7611643 1999-06-18 15:58.37    1
nTSecurityDescriptor
7597742                   Bldg\DC1 7611643 1999-06-18 15:58.37    1 name
7597742                   Bldg\DC1 7611643 1999-06-18 15:58.37    1
msiScriptPath
```

```
7597742                 Bldg\DC1 7611643 1999-06-18 15:58.37    1
cOMClassID
7597742                 Bldg\DC1 7611643 1999-06-18 15:58.37    1
cOMProgID
7597742                 Bldg\DC1 7611643 1999-06-18 15:58.37    1
localeID
7597742                 Bldg\DC1 7611643 1999-06-18 15:58.37    1
computerArchitecture
7597742                 Bldg\DC1 7611643 1999-06-18 15:58.37    1
revision
7597742                 Bldg\DC1 7611643 1999-06-18 15:58.37    1
packageType
7597742                 Bldg\DC1 7611643 1999-06-18 15:58.37    1
packageName
7597742                 Bldg\DC1 7612100 1999-06-18 16:01.02    2
packageFlags
7597742                 Bldg\DC1 7611643 1999-06-18 15:58.37    1
versionNumberHi
7597742                 Bldg\DC1 7611643 1999-06-18 15:58.37    1
versionNumberLo
7597742                 Bldg\DC1 7612100 1999-06-18 16:01.02    3
lastUpdateSequence
7597742                 Bldg\DC1 7611643 1999-06-18 15:58.37    1
msiFileList
7597742                 Bldg\DC1 7611643 1999-06-18 15:58.37    1
categories
7597742                 Bldg\DC1 7611643 1999-06-18 15:58.37    1 url
7597742                 Bldg\DC1 7611643 1999-06-18 15:58.37    1
objectCategory
7597742                 Bldg\DC1 7611643 1999-06-18 15:58.37    1
upgradeProductCode
7597742                 Bldg\DC1 7611643 1999-06-18 15:58.37    1
canUpgradeScript
7597742                 Bldg\DC1 7611643 1999-06-18 15:58.37    1
fileExtPriority
7597742                 Bldg\DC1 7611643 1999-06-18 15:58.37    1
productCode
7597742                 Bldg\DC1 7612100 1999-06-18 16:01.02    2
msiScriptName
7597742                 Bldg\DC1 7611643 1999-06-18 15:58.37    1
installUiLevel
```

For monitoring replication, use the tools Repadmin, Replmon, and Dsastat in the /Support directory on the Windows 2000 operating system CD.

Join and Authentication Issues

If you can't join a computer to an Active Directory domain, or if a computer can't communicate with any other computer in the network, the situation might be the result of join and authentication problems.

This section discusses diagnostic tools and gives examples of possible authentication problems, along with suggested solutions. The first step toward identifying and diagnosing Active Directory join and authentication problems is to review how a Windows 2000–based computer joins a domain, what permissions are required by a user, and how a secure channel is established.

Joining a Computer to a Domain

To review, when you join either a Windows NT 4.0–based or a Windows 2000–based client to a domain, the following occurs:

- The domain name is validated.
- A domain controller in the domain is located through a call to DsGetDcName.
- A session is established with the domain controller under the security context of the passed-in credentials that are supplied in the **Network Identification** tab under **System Properties** in **Control Panel**.
- The computer account is enabled. If the flags are so specified (NETSETUP_ACCT_CREATE), the APIs create the computer account on the domain controller.
- The local password for this account is created in the Local Security Authority (LSA).
- The local primary domain information LSA policy is set to refer to the new domain. This includes the domain name and the domain SID.

Note For a Windows 2000–based client only, the LSA policy consists of the domain name, domain SID, DNS domain name, DNS forest name, and domain GUID.

- The name of the DNS name assigned to the local computer is updated.
- The local group membership is changed to add members of the Domain Admins group to the Local Accounts Administrators group.
- The Net Logon trusted domain cache is initialized to the trusted domains domain list.
- For Windows 2000–based clients only, the Windows Time Service is enabled and started.
- The Net Logon service is started.

Changes Occurring on Domain Controllers in the Domain That the Client is Joining

When a client joins a domain, the following changes occur on Windows NT 4.0–based and Windows 2000–based domain controllers:

- A computer object is created. The name of this object is generated by appending a dollar sign ($) to the name (uppercase letters) of the client.
- On Windows 2000–based domain controllers only, the Net Logon service creates Service Principle Names (SPNs) on the computer object.

Identifying Whether You Have a Problem Authenticating

You can identify whether you have a problem authenticating (or joining) a computer to a domain by verifying that the local workstation is working. Do this by running the Netdiag tool. Read the output from the top, and look for the words "ERROR" or "FATAL." (Many failures are not relevant to the domain itself; but you should follow up on them because they involve network connectivity issues.) If you don't find these words in the output, continue as follows:

- Run **netdiag /v** (verbose mode). Do you receive any specific error messages or FATAL errors?
- If the answer to the preceding question is "No," run **netdiag /debug**. Do you receive any specific error messages or FATAL errors?
- If Netdiag displays an error or failure with the domain itself, check the *%SystemRoot%*\debug\netsetup.log file for join errors.

Note If the local workstation is functional, examine the Netsetup.log file that is located in the *%SystemRoot%*\debug folder. (This is where the join process is logged.) Are any specific error messages logged?

Format of Netsetup.log File

A typical line in Netsetup.log is formatted as follows:

<time-stamp> <function-name>: <description of operation>: <status code in hexadecimal code>.

An example is the following:

```
08/11 14:08:29 NetpJoinDomain: status of connecting to dc '\\DC9': 0x0
```

The description of the join operation is usually self-explanatory. The status code is NET API_STATUS or a Win32 error code. A "0x0" code indicates success; any other code indicates an error.

Specific Join Issues

You might encounter problems when you join your computer to a domain. Even though these problems are reported as join problems, some of the most frequently reported ones are not related to the join process. Looking at the Netsetup.log is sufficient to quickly spot such cases.

The following are some of the most common errors that relate to join issues:

- Failure to find or to connect to a domain controller.
- Transient network conditions or having specified an incorrect domain name.
- Failure to create a computer account.

The error code shown in Table 10.6 comes under this category.

Table 10.6 "Failure to find a domain controller" Error Code

Description	Actual Error	Error Code
Failure to find or connect to a domain controller.	ERROR NO SUCH DOMAIN	1355

The following is an example of this error:

```
07/20 16:51:10 NetpDsGetDcName: trying to find DC in domain
'verylongdomain1', flags: 0x1020
07/20 16:51:11 NetpDsGetDcName: failed to find a DC having account 'A-
USHAS2-80C$': 0x525
07/20 16:51:11 NetpDsGetDcName: failed to find a DC in the specified
domain: 0x54b
07/20 16:51:11 NetpDoDomainJoin: status: 0x54b
```

The join process usually tries to find a domain controller that already has a computer account for the computer that is currently being joined to the domain. If such a domain controller is not found, it tries to find another domain controller. The preceding example shows that the join domain operation failed because a domain controller was not located for the specified domain.

To investigate further, run **nltest /dsgetdc:<*domain-name*>** and examine the output. If you still receive errors, either the domain really does not exist or there is a transient net error that is preventing domain controller discovery. By running Netdiag.exe and examining the output, you usually can determine the cause. A "Failure to connect to a domain controller" message usually means that transient net errors or insufficient credentials are the cause. Table 10.7 shows some error codes that come under this category.

Table 10.7 "Failure to connect to a domain controller" Error Codes

Description	Actual Error	Error Code
Bad credentials.	ERROR_LOGON_FAILURE	1326
Time skew that can cause failure of Kerberos authentication.	ERROR_TIME_SKEW	1398
Failure to connect to a domain controller.	ERROR_ACESS_DENIED	5
No domain controller found.	ERROR_NO_LOGON_SERVERS	1311

The following is an example of this type of error code:

```
07/20 14:47:34 NetpDsGetDcName: trying to find DC in domain 'reskit',
flags: 0x1020
07/20 14:47:50 NetpDsGetDcName: failed to find a DC having account
'TO_A$': 0x525
07/20 14:47:50 NetpDsGetDcName: found DC '\\reskit' in the specified
domain
07/20 14:47:50 NetUseAdd to \\reskit\IPC$ returned 1326
07/20 14:47:50 NetpJoinDomain: status of connecting to dc '\\reskit:
0x52e
07/20 14:47:50 NetpDoDomainJoin: status: 0x52e
```

The previous example shows a failed attempt to find a domain controller having the account "TO_A$". This is not a fatal error because the code then tries to find any domain controller in the specified domain. After a domain controller is found, an attempt is made to connect to it by using the credentials that are supplied. This attempt failed with error 0x52e (ERROR_LOGON_FAILURE). This indicates that the credentials that were supplied do not have sufficient access rights for connecting to the domain controller.

To investigate the problem of failing to find a domain controller, run an equivalent command from the command prompt to confirm the preceding analysis.

net use \\dcname\ipc$ /u:*<domain\user>* *<password>*

Note You need to perform the net use if you failed to connect to the domain controller. If you failed to find the domain controller, you should perform **nltest /dsgetdc:** to try to locate the domain controller.

If this fails with the same error, a Network Monitor sniffer trace of the join operation would be helpful in diagnosing the failure.

If you receive the error "Failure to create a computer account," it usually means that either the account already exists or that there are insufficient access rights available to the user who is trying to join. Table 10.8 shows the error codes that come under this category.

Table 10.8 "Failure to create a computer account" Error Codes

Description	Actual Error	Error Code
Computer account usually exists already, and security on that account does not allow you to join—usually because the computer was joined previously by using different computer account credentials.	ERROR_ACCESS_DENIED	5
The user has joined so many computers that he has exceeded the default per user computer quota (by default, 10).	ERROR_DS_MACHINE_ACCOUNT_QUOTA_EXCEEDED	8557
The specified user already exists.	ERROR_USER_EXISTS	2224

The following example indicates an access denied error.

```
08/11 14:08:30 NetpManageMachineAccountWithSid: NetUserAdd on '\\DC9'
for 'A-ERINCO-TBCB$' failed: 0x5
```

The following example indicates there is no error.

```
08/11 14:08:30 NetpManageMachineAccountWithSid: NetUserAdd on '\\DC9'
for 'A-ERINCO-TBCB$' failed: 0x8b0
08/11 14:08:30 NetpManageMachineAccountWithSid: status of attempting to
set password on '\\DC9' for 'A-ERINCO-TBCB$': 0x0
```

This is not an error because the NetUserAdd operation fails with 0x8b0 (NERR_UserExists), which indicates that the computer account already exists on that domain controller.

Note Failure usually occurs when the account already exists. Error 5 occurs if the user does not have access on the account, in which case an attempt is made to set a new password on the account that succeeds.

To investigate further, you have to acquire the security descriptor and view the permissions on the computer account object. You can use either the Active Directory User and Computers MMC console or the Ldp tool.

For more information about how to view permissions and access control entries on specific objects with the Active Directory User and Computers console, see Windows 2000 Server Help. For more information about access control entries and security descriptors, see "Access Control" in this book.

To investigate further, connect to the domain controller by using the Ldp tool. Acquire the security descriptor on the computer account and determine whether the user trying to join has sufficient permissions to gain access to the computer account.

▶ **To use Ldp to acquire the security descriptor**

1. From the **Start** menu, click **Run**, and then type the following:

   ldp

2. Connect and bind to a domain controller in the domain whose security descriptor you are searching for.
   - To connect, on the **Connection** menu, click **Connect**, and then type a server name.
   - To bind, on the **Connection** menu, click **Bind**, and then type an account name, password, and domain if you want to connect to a domain other than the domain to which you are currently logged on.
3. On the **Browse** menu, point to **Security**, and click **Security Descriptor**.
4. Provide distinguished name of the computer object whose security descriptor you are looking for.

Here is a sample output:

```
Revision:   1
Sbz1:       0
Control:    (0x8c04)
    SE_DACL_PRESENT
    SE_DACL_AUTO_INHERITED
    SE_SACL_AUTO_INHERITED
    SE_SELF_RELATIVE
Owner:
            S-1-0x000005--0x20-0x220
            BUILTIN\Administrators
Group:
            S-1-0x000005--0x20-0x220
            BUILTIN\Administrators
Dacl:
    Revision:   4
    Sbz1:       0
    Size:       972
    No of Aces: 24
    Sbz2:       0
    Ace[0]:
        Type: (0)
            ACCESS_ALLOWED_ACE_TYPE
        AceSize: 0x24
        AceFlags: (0x0)
        Mask: 0x000f01ff
        Sid:
            S-1-0x000005--0x15-0x3bdcf4dc-0x64495118-0x500cebdb-0x200
            DDS\Domain Admins
Ace[1]:
        Type: (5)
            ACCESS_ALLOWED_OBJECT_ACE_TYPE
        AceSize: 0x28
        AceFlags: (0x0)
        Mask: 0x00000010
        Flags: 0x1
        ACE_OBJECT_TYPE_PRESENT
        Object Type:
            (in HEX)(59ba2f42-79a2-11d0-90-20-00-c0-4f-c2-d3-cf)

            GUID_PS_GENERAL_INFO

       Sid:
            S-1-0x000005--0xb
            NT AUTHORITY\Authenticated Users
```

For more information about interpreting mask, ACE types and flags, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. Follow the links to ntsam.h.

The following example shows a successful attempt to join a computer to a domain in the Netsetup.log file:

```
NETSETUP.log file
07/30 13:58:35 NetpDoDomainJoin
07/30 13:58:35 NetpMachineValidToJoin: 'USER1'
07/30 13:58:35 NetpGetLsaPrimaryDomain: status: 0x0
07/30 13:58:35 NetpMachineValidToJoin: status: 0x0
07/30 13:58:35 NetpJoinDomain
07/30 13:58:35   Machine: USER1
07/30 13:58:35   Domain: RESKIT
07/30 13:58:35   MachineAccountOU: (NULL)
07/30 13:58:35   Account: RESKIT\reskit
07/30 13:58:35   Options: 0x40001
07/30 13:58:35   OS Version: 5.0
07/30 13:58:35   Build number: 2089
07/30 13:58:35 NetpCheckDomainNameIsValid [ Exists ] for 'RESKIT'
returned 0x0
07/30 13:58:35 NetpValidateName: name 'RESKIT' is valid for type 3
07/30 13:58:35 NetpDsGetDcName: trying to find DC in domain 'RESKIT',
flags: 0x1020
07/30 13:58:50 NetpDsGetDcName: failed to find a DC having account
'USER1$': 0x525
07/30 13:58:50 NetpDsGetDcName: found DC '\\RESKIT-DC-08' in the
specified domain
07/30 13:58:51 NetpJoinDomain: status of connecting to dc '\\RESKIT-
DC-08': 0x0
07/30 13:58:51 NetpGetLsaPrimaryDomain: status: 0x0
07/30 13:58:51 NetpLsaOpenSecret: status: 0xc0000034
07/30 13:58:52 NetpJoinDomain: status of setting machine password:
0x0
07/30 13:58:52 NetpJoinDomain: status of setting netlogon cache: 0x0
07/30 13:58:52 NetpGetLsaPrimaryDomain: status: 0x0
07/30 13:58:52 NetpSetLsaPrimaryDomain: for 'RESKIT' status: 0x0
07/30 13:58:52 NetpJoinDomain: status of setting LSA pri. domain: 0x0
07/30 13:58:53 NetpJoinDomain: status of managing local groups: 0x0
07/30 13:58:54 NetpJoinDomain: status of starting Netlogon: 0x0
07/30 20:58:55 NetpJoinDomain: status of setting
ComputerNamePhysicalDnsDomain 'reskit.reskit.com': 0x0
```

```
07/30 20:58:55 NetpDsSetSPN: Setting DnsHostName
'USER1.reskit.reskit.com' on
'CN=USER1,CN=Computers,DC=reskit,DC=microsoft,DC=com'
07/30 20:58:55 NetpDsSetSPN: Setting SPN
'HOST/USER1.reskit.reskit.com' on
'CN=USER1,CN=Computers,DC=reskit,DC=microsoft,DC=com'
07/30 20:58:55 NetpJoinDomain: status of disconnecting from
'\\RESKIT-DC-08': 0x0
07/30 20:58:55 NetpDoDomainJoin: status: 0x0
```

Permissions on Computer Account Objects

This section describes the security on domain computer accounts before and after an upgrade to Windows 2000 Server. This information can be used in troubleshooting permissions on computer account objects in Active Directory and in determining which user created the computer account before the upgrade.

The Discretionary ACL (DACL) contains access control entries (ACEs) that define permissions on a specific object. In Windows NT 4.0, when a computer account is created, the Domain Administrators local group becomes the owner of the computer account. The user who created the computer account is stored as part of its data, and the DACL on the computer account includes limited rights for the user (such as deleting the account).

When you upgrade a Windows 2000–based server, the following changes occur on each computer account:

- A computer account object is created in the default Computers container. The original owner (for example, administrator) of the computer account remains the same. The privileges that the original owner had on the computer object in Windows NT 4.0 are retained as part of the upgrade.
- The DACL on the computer account is reset to the default that is defined for objects of the *computer* class in the schema. This DACL includes an entry for Creator Owner and, when viewed with ACL Editor, displays the name of the appropriate user.

Note Note that other ACEs can be present if users or groups are added or if permissions are changed on parent containers in Active Directory, which results in additional inherited permissions.

The following default DACLs apply to new computer accounts:

- Self:
  - Create All Child Objects
  - Delete All Child Objects

- Authenticated Users:
  - Read
  - Read Public Information
- System:
  - (Full Control)
- Creator Owner:
  - (Full Control)
- Domain Administrators:
  - (Full Control)
- Cert Publishers:
  - (no permissions)
- Enterprise Administrators (inherited permissions):
  - Read
  - Write
  - Create All Child Objects
  - Change Password
  - Receive As
  - Reset Password
  - Send As
  - Read Public Information
  - Write Public Information
- Account Operators:
  - (Full Control)
- Print Operators:
  - (no permissions)
- Everyone:
  - Change Password

Note If the account is created by using the privilege *add workstations to the domain,* then the rights of the Creator Owner are limited. Specifically, the Creator Owner is not allowed to change the DACL nor to delete the account. In addition, a quota check limits the number of objects that can be created by the person who is using the quota.

For more information about Default DACLs, see "Access Control" in this book.

Secure Channel Issues

For each Windows 2000–based client or server that is a member of a domain, there is a discrete communication channel, known as the secure channel. This secure channel is used by the Net Logon service on the client and on the domain controller to communicate with each other. The Netdom tool is used to reset the secure channel. If the computer account's password and the local password are not synchronized, the Net Logon service logs one or both of the following errors messages:

```
The session setup from the computer DOMAINMEMBER failed to authenticate.
The name of the account referenced in the security database is
DOMAINMEMBER$.
 The following error occurred: Access is denied.
   NETLOGON Event ID 3210:
   Failed to authenticate with \\DOMAINDC, a Windows NT domain
controller for domain DOMAIN.
```

The Net Logon service on the domain controller logs the following error message when the password is not synchronized:

```
NETLOGON Event 5722:
   The session setup from the computer %1 failed to authenticate. The
name of the account referenced in the security database is %2. The
following error  occurred: %n%3
```

Resetting Secure Channels and Computer Accounts

The following tools are available to reset the secure channel and the computer account:

- Resource Kit tools:
  - Netdom.exe
  - Nltest.exe
- Active Directory Users and Computers console

Using Netdom to Reset the Secure Channel

By using the Netdom.exe command-line tool, which is provided in the Windows 2000 Resource Kit, you can reset the secure channel between the domain's member workstation and the domain controller. For example, suppose you have a domain member named DOMAINMEMBER. You can reset the member's secure channel by running the following command:

```
netdom reset member /domain:domain
```

You can run this command on the member DOMAINMEMBER. To run this command on any other member or domain controller in the domain, you must provide an account that has administrator access to DOMAINMEMBER.

For example:

```
Netdom reset member /domain:domain /usero:member-admin
/passwordo:member-pw
```

Adding a Workstation or Member Server to a Domain

To add a workstation or member server to a domain, do the following:

1. Add the workstation Work1 to the Windows NT 4.0 domain Domain1.
2. Netdom add /d:domain1 work1/ ud:domain1\admin /pd:password.
3. Add the workstation Work1 to the Windows 2000 domain reskit.com in the organizational unit my-computer, as shown here:

   ```
   Netdom add /work1 /d:reskit.com /OU:OU=my-computers,DC=reskit,DC=com
   ```

Note The /OU parameter requires a complete distinguished name as specified by RFC 1779. If the /OU parameter is not specified, the computer account is created in the Computers container.

Joining a Workstation or Member Server to a Domain

To join a workstation or member server to a domain, you can use the Netdom tool. For example, to join a workstation called Work1 to the reskit.com domain in the my-computers organizational unit, carry out the following:

```
Netdom join /d:reskit.com /OU:OU=my-computers,DC=reskit,DC=com
/reboot:120.
```

In addition to adding the computer account to the domain, the workstation is modified to contain the appropriate shared secret to complete the Join procedure. If the Join procedure can be completed, the /reboot switch causes the computer to be automatically shut down and restarted after giving the user two minutes to save work in progress.

Using Nltest to Reset the Computer Secure Channel

By using the Nltest.exe command-line tool, you can reset secure channels that computers have with domain controllers in their domains. Nltest.exe can be used to test the trust relationship between a computer that is running Windows 2000 and is a member of a domain and a domain controller on which its computer account resides, as shown in the following example:

```
C:\Ntreskit\Nltest.exe

   Usage: nltest [/OPTIONS]

   /SC_QUERY:<DomainName> - Query secure channel for <domain> on
   <ServerName>

   /SERVER:<ServerName>

   /SC_RESET:<DomainName> - Renegotiates the secure channel in the
specified domain for a local or remote workstation, server, or domain
controller

An example to reset the secure channel:

nltest /sc_query:reskit /server:Server22
Flags: 30
Connection Status = 0 0x0 NERR_Success
Trusted DC Name \\Server1.reskit.com
Trusted DC Connection Status Status = 0 0x0 NERR_Success
The command completed successfully


nltest /sc_reset:reskit /server:Server2
Flags: 30
Connection Status = 0 0x0 NERR_Success
Flags: 30 HAS_IP HAS_TIMESERV
   Trusted DC Name \\server.reskit.com
   Trusted DC Connection Status Status = 0 0x0 NERR_Success
   The command completed successfully
```

Using the Active Directory Users and Computers Console to Reset Computer Account Passwords

By using Windows 2000, you can also reset the computer account password in the Active Directory Users and Computers console. Right-click the computer object in the **Computers** folder or other appropriate container, and then click **Reset Account**. The **Reset Account** context menu resets the computer account password back to a starting password.

This is used only if the computer has been taken offline and been completely reinstalled. Resetting the account password allows the (rebuilt) computer to rejoin the domain using the same name. If this command is carried out when the computer has not been reinstalled, the computer cannot authenticate in the domain.

Note Resetting the password for domain controllers by using this method is not allowed.

Using Nltest to View Trusted Domains

Different data about the trust relationship is kept in several key attributes of each *trustedDomain* object. The following are the key attributes:

flatName. Contains the NetBIOS name of the domain for this trust relationship.

trustDirection. Contains the direction of the established trust relationship:

- 0=Disabled
- 1=Inbound (Trusting domain)
- 2=Outbound (Trusted domain)
- 3=Both (Trusted and trusting domains)

trustPartner. Contains a string that represents the DNS-style name of the domain if it is a Windows 2000 domain or the NetBIOS name of the domain if it is trust relationship between a Windows 2000 domain and a non-Windows 2000 domain.

trustType. Contains the type of trust relationship that has been established to the domain.

- 1=A trust relationship between a Windows 2000 domain and a Windows NT 4.0 or earlier domain.
- 2=A Windows 2000 trust relationship.
- 3=A trust relationship between a Windows 2000 domain and a non-Windows Kerberos realm.

By using the Nltest command-line tool, you can display the current list of trusted domains known by a specified server. Nltest.exe is available with Windows 2000 Server Support Tools. (To use Nltest, install the tools that are located in the Support\Tools folder on the Windows 2000 Server operating system CD. To install the tools, double-click the **Setup** icon in that folder. For more information about using Nltest, see Windows 2000 Support Tools Help.)

Use the **/domains_trusts** option to list the domains that have trust relationships with the current domain. For each domain listed in the results, the following data is displayed:

- Trust Index (a number that identifies an entry in the enumerated list of trusts).
- NetBIOS domain name of the trusted domain (for example, reskit).
- DNS domain name of the trusted domain (for example, reskit.com).
- Trust type (NT 4 for trust relationship with a Windows NT domain), NT 5 (for a trust relationship with a Windows 2000 domain), or MIT (for a trust relationship with a non-Windows Kerberos realm). For more information about types of trust relationships, see "Active Directory Logical Structure" in this book.
- In addition, the following values are reported where applicable:
  - Forest Tree Root: Identifies the forest root domain.
  - Forest Trust Index: Indicates the domain that is the forest root.
  - Primary Domain: Identifies the domain in which the contacted server is located.
  - Direct Outbound: Identifies the domain as being directly trusted by the primary domain.
  - Direct Inbound: Identifies the domain as directly trusting the primary domain.
  - Attr: Returns the bits specifying the value in the *trustAttributes* attribute on the trustedDomain object. This value determines, for example, whether the trust relationship is transitive or nontransitive.
  - Native: Identifies a primary domain that is running in native mode. Where no value is displayed for primary domain, the primary domain is running in mixed mode.

For example, the following Nltest command is executed on a computer that is a member of the noam.reskit.com domain returns.

```
D:\>nltest /domain_trusts
List of domain trusts:
    0: RESKIT reskit.com (NT 5) (Forest Tree Root) (Direct Outbound)
(Direct Inbound) ( Attr: 0x400000 )
    1: AVIONICS avionics.reskit.com (NT 5) (Forest: 0)
    2: EUROPE europe.reskit.com (NT 5) (Forest: 0)
    3: NOAM noam.reskit.com (NT 5) (Forest: 0) (Primary Domain) (Native)
The command completed successfully
```

This output indicates the following:

- Reskit.com is the forest root domain.
- All of the domains are in the same forest as reskit.com (identified by the index number 0).
- All of the trust relationships are Windows 2000 trust relationships (indicated by "NT 5").
- Noam.reskit.com is the domain of the server that is running Nltest.
- Noam.reskit.com, which is a primary domain, is running in native mode.

To run a query on a specific server, type **nltest /server:<*servername*>domain trusts**. For example, the "domain that is trusted" list might be displayed if a query is run on a domain controller in the root domain of the forest. (This example shows root.com as the root domain.)

```
   0: TESTDOMAIN testdomain.root.com (NT 5) (Forest: 3) (Direct
Outbound)
   1: CHILD child.root.com (NT 5) (Forest: 3) (Direct Outbound)
   2: GRANDCHILD grandchild.child.root.com (NT 5) (Forest: 1)
   3: ROOT root.com (NT 5) (Forest Tree Root) (Primary Domain)
   4: NT4DOMAIN (NT 4) (Direct Outbound)
   5: NEWROOT newroot.com (NT 5) (Forest Tree Root) (Direct Outbound) (
Attr:
   0x800000 )
```

Note Note that Nltest shows trusted domains with transitive trust relationships as Windows 2000 trust relationships without the Direct Outbound tag.

Another way to view domains and trust relationships is by using ADSI Edit.

▶ **To view trusted domains and trust relationship properties by using ADSI Edit**

1. In ADSI Edit, expand the domain directory partition node and navigate to the System container.
2. In the console details pane, use the **Class** column to identify all objects with the type *trustedDomain*.
3. To view properties, right-click the *trustedDomain* object, and then click **Properties**.
4. In the **Select which properties to view** box, click **Both** to view both optional and mandatory attributes.
5. In the **Select a property to view** box, select a property. Its value is displayed in the **Value(s)** box.

Checking Trust Relationships Authenticated By the Kerberos v5 Protocol

Use the Netdom tool to verify the Kerberos v5 authentication protocol between a client and a target domain. The Netdom tool trust verification option with the */Kerberos* switch allows you to obtain a session ticket from the Kerberos authentication service in the target domain. If successful, the conclusion is that Kerberos operations such as Key Distribution Center (KDC) referrals, are operating correctly between the workstation and the target domain. Upon failure, the list of referral tickets currently cached, are displayed..If you do not receive the session ticket, the cause of failure can be determined by tracing the list of referral tickets from the KDCs located on the path toward the target domain.

To verify the Kerberos authentication protocol issue the following command:

```
NETDOM TRUST <trusting_domain_name> /d: <name of the trusted domain>
/Kerberos /UserO:<User account for making the connection with the
trusted domain> /PasswordO:<Password of the user account specified by
/UserO > /UserD:<User account used to make the connection with the
domain specified by the /domain argument >
/PasswordD:<trusted_domain_user_password>
```

Note Both users must be specified because the command will attempt a Kerberos v5 authentication of those users.

The above command will verify the following:

- The trust passwords are correct (for example, determine if the passwords match).
- The users can be located in Active Directory.
- The users can be authenticated through the issuance of Kerberos v5 tickets.

For more information on the Netdom tool, see Windows 2000 Support Tools Help. For more information on Kerberos v5 authentication, see "*Authentication*" in this book.

Fail Logons in Absence of Global Catalog Servers

For Windows 2000 in native mode a Global Catalog is required for the logon process. If the domain controller cannot contact a Global Catalog server, the user is not be able to log on. An exception is made only for the administrator account in the domain (RID 0x1F4). This account is allowed to log on even without a Global Catalog, so that in an emergency situation a Global Catalog can be configured.

Specifically, group expansion during token creation when the user is logging onto a workstation is as follows:

1. Add the user's SID in the token.
2. Add the global groups that the user is part of in the token.
3. Add the universal groups to which the user's SID and the global groups belong in the token.
4. Add the domain local groups to which the preceding accounts belong to the token. This step is performed at a domain controller for the domain to which the workstationbelongs.

   Domain local groups are not added to the token, if this domain is in mixed mode.
5. Add the local and built-in local group memberships for the groups in the workstation of the set computed in steps 1 through 4. If the user is connecting to or logging on to a domain controller, this step addresses only the built-in local groups; if the domain local groups were evaluated in step 4.

Access Control

Access control problems can manifest themselves when you cannot access Active Directory objects. They can also happen when you get "Access Denied" errors when trying to map to another network resource, or when trying to run **net view** at the command prompt to, see a particular server.

This section discusses diagnostic tools and gives examples of possible access control problems, along with suggested solutions. The first step toward identifying and diagnosing Active Directory access control problems is to find out the security descriptor of the object being accessed, and then to look in Event Viewer.

Event Viewer

To view access control errors, you need to activate auditing. Then you can see events logged in the Security log.

Gaining Access to Other Computers

The Net view command-line tool enables you to determine whether you can gain access to other network computers. To view the computer, to determine whether you can access other computers, carry out the following:

net view \\<*computer name*>

Determining Whether You Can View Other Computers

If you can view other computers, can you use the computer to connect to your domain by using **net use \\server1\ipc$ /u:reskit*<user>* *<password>*? If you do have connectivity, can you establish a connection through a local domain account? If the first option fails, this indicates a problem with the trust links. To correct problems with the trust links, run Nltest.exe as shown in the following example:

```
nltest.exe /server:<server> /sc_query:<domain of server that is failing>
e.g. nltest /server:server1 /sc_query:reskit
```

- If the result from the Nltest.exe comes back with a specific server (server1) error, ensure that that domain controller has connectivity to the domain controller of the secure channel query. Test each server by using **ping** and **net view** commands to ensure server functionality.
  - If **ping** works, but **net view** doesn't, verify the type of name resolution (for example, WINS or DNS).
  - If **ping** does not work, verify transport connectivity or functionality status of the server in question. (Note that the server might not be running TCP/IP.)
  - If the domain controller and connectivity are good but, **net use** still fails, attempt to reset secure channel by using **nltest /server:*<server>* /sc_reset:*<domain of server that is failing>***.

Note You must be connected as an Administrator to the target *<server>*.

If you cannot connect to your domain and *cannot* establish a connection through a local domain account, the LSA and SAM might be the problems. If you receive an error message while running the **net view *<server>*** or **net use** command, do the following:

- When a computer is not able to be authenticated by any domain user, the first step is to try to ping the computer.
  - If there is no response to the ping, perform a local review. Do this by verifying whether your computer has bug checked or access violated. If not, verify the local IP address returned by DNS by using the **ping** command. (For example, is the IP address returned from the ping the same as the local **IPConfig/all** address?)

- If you can obtain the IP address of the computer by using **ping**, is the computer viewable by using the **net view** command. In other words, does it return "access denied" or the share list? If not, WINS or the SMB Services might be suspect. Perform a Network Monitor sniffer trace to determine which name server is responding and, with what address. Verify the address returned to the server in question. If they do not match, this indicates a name server inconsistency. If they do match, a sniffer trace might help identify the error.

Gaining Access to Active Directory Objects

If you have a problem gaining access to Active Directory objects, this is almost always because the ACL embedded in the security descriptor for the object does not explicitly grant access to the person performing the operation.

Using Dsacls to View an Object's Security Descriptor

To view the security descriptor of an Active Directory object, you can use the Dsacls command-line tool. By using Dsacls.exe, you can view the security descriptor for an object, which includes the ACL. The ACL contains the discretionary access control list (DACL) and the system access control list (SACL).

The DACL identifies the permissions or rights that particular security groups have with regard to the Active Directory object. If it is not obvious why the DACL doesn't grant the user the access privileges he or she desires, you have to reset the permissions and add permissions so that you can gain access to the appropriate resource.

Note There are other ways to view the ACL. For example, from the **Security** tab in the various consoles.

Resetting the permissions might fix the particular problem, but it might affect another area. It is more reliable to add the required permissions. Alternately, if the permissions on the object are set such that no meaningful access is possible, resetting them to default might be the only option.

Using SDCheck to Verify ACL Inheritance

To verify that ACL entries are being propagated correctly through parent-child relationships, use the **sdcheck** (Security Descriptor Check) command-line tool. You can determine whether ACLs are being inherited correctly and if ACL changes are being replicated from one domain controller to another. For more information about inheritance, see "Access Control" in this book.

To list all of the ACEs that are inherited from the parent to the child, for the child object that is having its password ACLs changed, you can use the **sdcheck/dumpall** command.

Viewing ACEs with the Ldp Tool

To view ACEs for particular Active Directory objects, use the Ldp.exe tool included in the Microsoft Windows 2000 Resource Kit companion CD. To use this tool select the security principal (object) of reference, such as cn=<user1>, cn=Users, cn=Domain, dc=<domain name>, dc=<root domain>, and so on. From the **Browse** menu, click the security descriptor for the object of reference to provide the default access control list and the system access control list in a low-level format.

Taking Ownership of an Object and Resetting the ACL

Another benefit here is that, if you have an administrator who inadvertently set an access control so that nobody can view an object in the directory, you can use ADSIEdit to take ownership of that object and reset the ACL to one where users can do that. For more information about ACLs, see "Access Control" in this book.

Using the Netdom Tool to Maintain Your Enterprise

You can use Netdom to manage computer accounts for member workstations and member servers. The options of Add, Remove, and Query, allow you to specify the OU for the computer account, move an existing computer account for a member workstation from one domain to another, and list the member workstations or servers in a domain.

Only certain objects are supported for each command. These objects are listed in Table 10.9.

Table 10.9 Netdom Tool Commands

Command	Valid object(s)	Example
add	Name of client or stand-alone server	**Netdom add /d:reskit.com mywksta**
join	Name of client or stand-alone server	**Netdom join /d:reskit.com mywksta**
move	Name of client or stand-alone server	**Netdom move /d:newdomain mywksta**
remove	Name of client or stand-alone server	**Netdom remove /d:reskit.com mywksta**
reset	Name of client or stand-alone server	**Netdom reset /d:reskit.com mywksta**
verify	Name of client or stand-alone server	**Netdom verify /d:resourcedom yourwksta**

(continued)

Table 10.9 Netdom Tool Commands *(continued)*

Command	Valid object(s)	Example
rename	Name of backup domain controller. This only applies to Windows NT 4.0 and Windows NT 3.51 backup domain controllers only.	**Netdom rename /d:newdomain BDC51**
query	Workstation, Server, Domain controller, Primary Domain Controler, Domain, FSMO, OU	**Netdom query /d:reskit.com DC**
time	Optional. Name of specific domain controller or Member	**Netdom time /d:masterdom**
trust	Name of a domain	**Netdom trust /d:masterdom resourcedom**

Auditing Policy

To be able to track user activity, you need to establish an auditing policy. An auditing policy defines the kinds of events that you want to monitor for a particular group of users. It is defined on the SACL.

The following includes the implementation sequence for establishing an auditing policy:

- Enable auditing policy for the domain controller
- Set an auditing SACL on the object of interest
- Force object access
- Check the audit log

Enable Auditing Policy for the Domain Controller

The first procedure to implement is an auditing policy to enable the domain controller to audit the actions of specified security principals. You must be logged on as a member of the Administrators group to perform this procedure.

▶ **To enable the auditing policy for the domain controller**

1. From the **Start** menu, select **Programs**, **Administrative Tools**, and **Active Directory Users and Computers**.
2. Locate the container holding the domain controller, and then right-click it to display the **Properties** page.
3. Click the **Group Policy** tab, and then click **Edit** to edit the **Default Domain Policy** (*<domain controller name>*) from the Group Policy window.
4. In the left pane, click **Computer Configuration**.

5. In the right pane, double-click each of the following in succession: **Windows Settings**, **Security Settings**, and **Local Policies**.
6. Double-click **Audit Policy**. The right pane includes an entry named **Audit Directory Service Access**.
7. Double-click **Audit Directory Service Access**, and then enable or disable successful or failed access attempts. Because domain controllers poll for policy changes every five minutes, it is best that these modifications take effect within five minutes. Then other domain controllers are updated at the next replication interval. Click **OK**.

Note For more information about specific auditing options, see "Group Policy" in this book.

Set an Auditing SACL on the Object of Interest

The next procedure in implementing an auditing policy is to set an Auditing SACL on the objects of interest. You must be logged on as a member of the Administrators group to perform his procedure.

▶ **To set an Auditing SACL on the object of interest**

1. From the **Start** menu, select **Programs**, **Administrative Tools, Active Directory Users and Computers**.
2. On the **View** menu, select **Advanced Features**.
3. Locate the container holding the objects, and right-click it to display the **Properties** page.
4. Open the container holding the object of interest by double-clicking it.
5. Right-click the object to display the **Properties** page, and select the **Security** tab.
6. Click **Advanced**, and then select the **Auditing** tab.
7. Click the **Add** button to display the **Select User, Computer, or Group** dialog box where you can specify whose actions to audit. Select a security principal name and click **OK**.
8. The **Auditing Entry for Administrator** dialog box displays with two tabs: **Object** and **Properties**.
   - The **Object** tab lets you identify generic and control rights to audit.
   - The **Properties** tab lets you identify property accesses to audit.

   Decide what the object access and rights apply to by choosing from the selections in the pull-down list. (The default is "this object and all child objects.") Click each tab you want to modify and check the boxes for the accesses or properties that you need to audit. When you are finished, click **Apply**, and then click **OK**.

9. At the **Access Control Settings** window, determine whether auditing entries must be inherited from the parent container to propagate this object. If so, check the **Allow inheritable auditing entriesfom parent to propagate to this object** box. When you are finished, click **Apply** and then click **OK**.
10. At the Properties window, determine whether auditing permissions must be inherited from the parent container to propagate this object. If so, check the box. When you are finished, click **Apply** and then click **OK**.

Access and Use Objects

After the domain controller and object auditing entries are made, perform a test to ensure the auditing policy is implemented according to your requirements. To do this, you must have a reference indicating the objects, security principals, types of access and entries that comprise the auditing policy. Create a test plan that lists a sampling of the accesses and actions for each security principal for which you have an auditing policy. Perform consecutive logons, logging in under a different security principal each time and performing the actions from auditing policy test plan.

Check the Audit Log

The last part of the implementation of an auditing policy is to verify that the auditing policy works as expected by reviewing the results of your test plan. To do this, use the Event Viewer MMC snap-in. Browse the Security log. To refresh the window to see the most recent access, press F5. The events are displayed according to the event category to which they pertain. Each entry provides an audit trail of who gained access to what at what time and whether they were denied access.

For more information about auditing, see Windows 2000 Server Help.

Advanced Troubleshooting

Advanced troubleshooting techniques apply to situations that go beyond the network connectivity, name resolution, domain controller, authentication, and access control situations previously discussed. To resolve the kinds of situations that warrant these advanced troubleshooting techniques, superior knowledge of Active Directory and expert proficiency in the use of Resource Kit Support tools are both required.

The situations that involve Advanced Troubleshooting include the following:

- Active directory diagnostic logging
- Active directory installation and removal issues
- Database issues
- Schema issues
- Flexible single-master operations
- Replication issues

Active Directory Diagnostic Logging

Active Directory records events in the directory services log in Event Viewer. You can use the log to monitor the activity level of Active Directory or to investigate problems.

By default, Active Directory records only critical error events. To instruct Active Directory to record other events in the directory service log, modify the registry. For more information about how to use the Windows 2000 registry editors, see the Windows 2000 Server Help.

Caution Do not use a registry editor to edit the registry directly unless you have no alternative. The registry editors bypass the standard safeguards provided by administrative tools. These safeguards prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your computer. Editing the registry directly can have serious, unexpected consequences that can prevent the computer from starting and require that you reinstall Windows 2000. To configure or customize Windows 2000, use the programs in Control Panel or MMC whenever possible.

The registry entries that manage diagnostic logging are stored in HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services \NTDS\Diagnostics. Each entry represents a type of event that Active Directory can log. The value of the entry determines the level of detail of the events that are logged and ranges from **0** (records default-level errors and standard verbosity) to **5** (most verbose and records all activity).Table 10.10 describes each of these values.

Table 10.10 Values for the Diagnostics Registry Entry

Option	Description
0 (None)	Only critical events and error events are logged. This is the default and should be changed only if a problem occurs.
1 (Minimal)	Very high-level events are recorded in the event log. These might include one message for each major task performed by the service. Use this setting to begin an investigation when the location of the problem is in doubt.
2 (Basic)	Events with a logging level of 2 or lower are logged.
3 (Extensive)	Events with a logging level of 3 or lower are logged.
	Messages are sent to the event log to record steps taken to run a task. This provides more information than the minimum level but not the detail of the maximum level. Use this when the problem has been narrowed to a service or group of categories
4 (Verbose)	Events with a logging level of 4 or lower are logged.
5 (Internal)	All events are logged, including debug strings and configuration changes received.
	Provides a complete log of the operation of the service. Use this level when the problem is traced to a particular category or a small set of categories.

All of the entries in the Diagnostics subkey have the REG_DWORD data type and a default value of **0**.

Note Logging levels should be set to **0** (None) unless a problem is being investigated.

All fatal and critical errors are logged at level **0**, and no user action is required to view them.

Increasing the level increases the detail of the messages and the number of messages emitted. Setting the value of entries in the Diagnostics subkey to greater then **3** can degrade server performance and is not recommended. The application event log fills up quickly when the logging level is increased.

Table 10.11 contains a list of registry entries in the Diagnostics subkey that store the directory service logging levels.

Table 10.11 Registry Entries in the Diagnostics Subkey

Registry Entry	Description
Knowledge Consistency Checker (KCC)	The KCC derives its input configuration from objects in the directory (for example, sites, servers and site links). The KCC reports if these objects are incorrect or missing. Events occurring during a run of the KCC. Messages fall into the following categories: KCC runtime errors, such as inconsistencies, resource errors or directory access problems. KCC output configuration problems. The new configuration cannot be built or is incomplete in some way. Perhaps too many servers are down to build a complete topology at this time.
Security Events	Events related to Windows 2000 Security, such as a user who tries to read or write an attribute with insufficient permissions, a user binding through MAPI, or a domain that has been changed to native mode.
ExDS Interface Events	Events related to communication between Active Directory and Exchange clients.

(continued)

Table 10.11 Registry Entries in the Diagnostics Subkey *(continued)*

Registry Entry	Description
MAPI Interface Events	Events related to communication between Active Directory and Exchange clients.
Replication Events	Events related to outbound replication, where changed objects are found and inbound replication, where these changes are applied to a local database. "Normal" errors during the course of replication, such as a domain controller being down, are not logged. They are kept as status and are available through the replication tools. The errors logged during replication are generally critical inconsistencies that require user intervention, as database errors. The other kind of events logged by the replication category are information about which objects and attributes were updated and why. Note that many attributes are updated each time replication occurs. Logging detail about attributes can generate a great deal of messages very quickly. A level of **1** is safe and might be informative as to the general types of operations occurring for replication. A level higher than level **2** can result in filling up the log file and performance degradation.
Garbage Collection	Events generated when objects marked for deletion are actually deleted.
Internal Configuration	Interpretation and display of the internal directory service operations.

(continued)

Table 10.11 Registry Entries in the Diagnostics Subkey *(continued)*

Registry Entry	Description
Directory Access	Reads and writes directory objects from all sources.
Internal Processing	Events related to the internal operation of Active Directory code such as processing security descriptor propagation. Error events in this category might be an indicator of serious problems in Active Directory. When the directory returns the status of "internal error," this category can be used to identify the problem for Microsoft support. Set this category to 1 on all computers involved (client and server) and reproduce the problem. Note the point in the code where the internal error was raised.
Performance Counters	Events related to loading and unloading the NTDS performance object and performance counters.
Initialization/Termination	Events related to starting and stopping Active Directory.
Service Control	Processes Active Directory service events.
Name Resolution	Resolution of addresses and Active Directory names.
Backup	Events related to the backup of Active Directory. Specifically, errors occurring when ESE database records are read or written for backup purposes. Generally only logged when a backup operation is underway.
Field Engineering	Internal debugging trace.
LDAP Interface Events	Events related to LDAP. An example of events logged include the following: the LDAP server closed a socket to a client, unable to initialize LDAP Simple Bind Authentication, and LDAP over SSL is now available.

(continued)

Table 10.11 Registry Entries in the Diagnostics Subkey *(continued)*

Registry Entry	Description
Setup	Events related to running the Active Directory Installation Wizard.
Global Catalog	Events related to Global Catalog. For example, "Promotion of this server to a Global Catalog will be delayed for *%1* minutes. This delay is necessary so that the required partitions can be made ready before the GC is advertised. The operations that occurs during this time include the KCC being run to generate the new topology, all read-only partitions in the enterprise being added to this server, and the contents of these partitions being replicated into this system. If you want to promote the GC immediately without enforcing this precondition, set the registry variable HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS\Parameters\GlobalCatalogDelayAdvertisement(sec) to a DWORD value of 0. The GC will be promoted on the next attempt to check preconditions. This value can also be set to the maximum number of seconds that the DSA will wait before promoting to a GC."
Inter-site Messaging	These messages are logged by the "Intersite Message" service, which is a separate service from the directory itself. There are two kinds of messages that are generated in this category: The ISM Service is responsible for transporting replication messages between sites. The ISM Service is also responsible for calculating site routes for the KCC to use. Note that the messages in this category are either fatal configuration errors, or informational messages about the amount of traffic being carried.

Summary of Log Files Used in Active Directory

Windows 2000 maintains specific log files that pertain to Active Directory. For example, when installing or removing Active Directory by using the Active Directory Installation Wizard (also known as dcpromo), several log files are created in the %SystemRoot%\Debug that you can use to investigate the actual process. You need to be familiar with the information provided in these files because they provide relevant facts about Active Directory performance and services. The default location for the log files is the *%SystemRoot%*\Debug folder. For more information about Windows 2000 log files, see the Microsoft TechNet Web link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. Search the Technical Support section of this site for Knowledge Base articles and other sources of technical information.

DCPromoUI.log

The DcpromoUI.log file contains a detailed progress report of the Active Directory installation and removal processes. Its default location is the *%SystemRoot%*\Debug folder on Windows 2000–based servers. Logging begins when the Active Directory Installation Wizard is opened and continues until the summary screen appears; regardless of whether it terminated prematurely or completed successfully. If the installation or removal failed, detailed error messages appear in the log immediately after the step that caused the failure. When the installation or removal process is successful, the log provides positive confirmation of that fact.

Additionally, the DcpromoUI.log file includes the following useful information, about the installation or removal of Active Directory:

- The name of the source domain controller for replication.
- The directory partitions that were replicated to the target server
- The number of items that were replicated in each directory partition
- The services configured on the target domain controller
- The access control entries (ACEs) set on the registry and files
- The SYSVOL directories
- Applicable error messages
- Applicable selections that were entered by the Administrator during the installation or removal process

For more information about the Dcpromoui.log, see "Active Directory Installation and Removal Issues" later in this chapter.

DCPromos.log

The %windir%\debug\dcpromos.log is created by the user interface during the graphical user interface mode setup when a Windows 3.*x*–based or Windows 4.0–based domain controller is promoted to a Windows 2000 domain controller.

DCPromo.log

The DCPromo.log file is created by using the Active Directory Installation Wizard. Its default location is the %SystemRoot%\Debug folder on Windows 2000–based servers. It also records settings used for the promotion or demotion, such as the site name, the path for the Active Directory database and log files, time synchronization, and information about the computer account. The DCPromo.log file captures the creation of the Active Directory database, SYSVOL trees and the installation and modification of services.

For more information about the Dcpromo.log see "Active Directory Installation and Removal Issues" later in this chapter.

Netsetup.log

When joining a computer to a Windows 2000 domain, the Networking Setup (NetSetup) installs all the necessary Microsoft supported networking components. The Netsetup.log file provides information about the attempts to join domains and records any errors that might be preventing the join from being successful. Also, to install networking components not directly supported by Microsoft, the NetSetup tool provides a way to connect into the setup process for third-party components.

For more information about Netsetup.log, see "Authentication" earlier in this chapter.

Netlogon.log

The Net Logon service responds to network logon requests. The Net Logon service dynamically creates records in the DNS database that are used to locate a server.

The Netlogon.log file is created whenever the service is used. For more information about the Net Logon service, see "Name Resolution in Active Directory" in this book. For more information about Netlogon.log, see "Active Directory Architecture" earlier in this chapter.

Ntfrsapi.log

The File Replication service (FRS) text-based log file is the Ntfrsapi.log file. It resides in the *%SystemRoot%*\Debug folder. It tracks replication problems and contains events that take place during the installation or removal of Active Directory, for example, creating the NTFRS registry keys. For more information about FRS and the Ntfrsapi.log file, see the "File Replication Service" in this book and the Microsoft Personal Online Support link on the Web Resource page at http://windows.microsoft.com/windows2000/reskit/webresources.

Userenv.log

The output of this log file can be helpful in troubleshooting problems with user profiles and Group Policy processing. The log file resides in the *%SystemRoot%*\Debug folder.

Following is an example of the userenv.log file showing a failure to return a string representing the user guid of the current user.

```
USERENV(b8.a0) 17:02:31:274 GetUserGuid: Failed to get user guid with
1332.
USERENV(b8.a0) 17:02:31:584 GetUserGuid: Failed to get user guid with
1332.
USERENV(b8.a0) 17:02:31:584 GetUserGuid: Failed to get user guid with
1332.
USERENV(b8.cc) 17:02:31:715 ProcessGPOs: Starting user Group Policy
processing...
USERENV(b8.cc) 17:02:31:765 ProcessGPOs: User Group Policy has been
applied.
USERENV(b8.c0) 18:43:31:980 ProcessGPOs: Starting user Group Policy
processing...
USERENV(b8.c0) 18:43:32:030 ProcessGPOs: User Group Policy has been
applied.
```

Active Directory Installation and Removal Issues

To install and remove Active Directory, the Active Directory Installation Wizard (Dcpromo) is used. It is important that certain requirements are met to prevent an unsuccessful installation and removal of Active Directory in an enterprise. These requirements include planning for sufficient storage capacity, time synchronization, domain controller availability, DNS configuration, and administrator access permissions. Also, there are specific troubleshooting techniques that you can follow if you do encounter errors when installing and removing Active Directory. Some of the common problems you might encounter are the following:

- You cannot reach the server from which you are installing, perhaps because the DNS name is not registered yet.
- The name of the domain you are authenticating against is incorrect or not available yet.
- The user name and password you supplied are incorrect.
- The DNS server settings are not configured correctly

The recommended ways to diagnose and resolve these problems :

- Check the Event Viewer for specific errors related to Active Directory Installation Wizard on all domain controllers within the domain. For more information on examining Event Viewer messages , see "Network Connectivity" in this chapter.
- Wait and allow enough time for replication to occur on all domain controllers within the domain. For more information on diagnosing replication issues, see "Replication Issues" in this chapter.
- Check network connectivity. For more information on diagnosing and resolving network connectivity problems, see "Network Connectivity" in this chapter.
- Check for name conflicts. For more information on diagnosing and resolving naming conflicts, see "Name Resolution" in this chapter.

Note Whenever you encounter errors running the Active Directory Installation Wizard (Dcpromo.exe), examine the Dcpromoxx.log files. These files (Dcpromo.log and Dcpromoui.log) are located in the *%SystemRoot%*\debug folder. For more information about examining Active Directory Installation log files , see "Troubleshooting Active Directory Installation Wizard Errors" in this chapter.

Ensuring Sufficient Disk Space

To avoid any problems with installing or removing Active Directory, it is important to confirm that you have sufficient disk space on the network drives that are going to be configured to host the directory information tree (DIT) and log files.

The Active Directory Installation Wizard requires 200 megabytes (MB) of disk space for the Active Directory database and 50 MB for the ESENT transaction log files. File size requirements for the Active Directory database and log files are calculated by the Dssize tool. This is dependent on the number and type of objects in the domain database or databases held by the forest if the computer is serving as a Global Catalog server.

Ensuring Time Synchronization

The Kerberos v5 authentication protocol time synchronization between domain controllers and stand-alone servers defaults to five minutes of each other. Use the **net time** command to synchronize the time of the server that is being promoted with the domain controller from which you are holding the directory partition.

Note By default, time synchronization is done automatically.

Active Directory Installation Operations Master Requirements

The Domain Naming Master server must be available when a Windows 2000 Server attempts to join an existing tree by creating a new domain.

Note For an additional domain controller installation, the domain naming operations master is not needed.

To determine operations master availability, use the Active Directory Users and Computers console and the Active Directory Domains and Trusts console. You can also use the Ntdsutil tool to determine operations master availability and server location.

For more information about operations master roles, see "Managing Flexible Single-Master Operations" in this book.

Confirming DNS Configuration

To test the DNS configuration paths in the Active Directory Installation Wizard, type **ipconfig /release** at the command prompt, and then start the Active Directory Installation Wizard. This causes the wizard to run as if DNS is not properly configured. You cannot use the wizard to continue until it detects a valid DNS setup. To simulate completion of the configuration, run **ipconfig /renew** in another process, and then return to the Active Directory Installation Wizard.

Privileges Needed to Add an Additional Domain Controller

To install an additional domain controller, an administrator who is a member of the Built-in [administrators] group (for example, Enterprise Admins and Domain Admins) on your domain controller must have the "enable computer and user accounts to be trusted for delegation" privilege. This is necessary so that during the installation of Active Directory, the computer account canbe trusted for delegation.

Note By default, the delegation privilege is provided to the Built-in [administrators] group.

The Active Directory Installation Wizard attempts to enable the computer account to be trusted for delegation for an install of an additional domain controller. However, there might be situations where the "enable computer and user accounts to be trusted for delegation" privilege is not provided by default to the Built-in [administrators] group. In that case, the security configuration engine must correct the this privilege problem before the Active Directory Installation Wizard can successfully run or you manually need to give the priviledge to a computer account.

▶ **To give the "enable computer and user accounts to be trusted for delegation" privilege to a computer account**

1. Open the Active Directory Users and Computers console.
2. Expand and right-click **Domain Controllers**, **Properties**, and the **Group Policy** tab.
3. Modify the "default domain controllers policy" Group Policy object, computer configuration, windows settings, security settings, local policies, user rights assignments and open "enable computer and user accounts to be trusted for delegation."
4. Add the account you want to use for the domain controller promotion process or group of which it is a member.

Operations Master Availability Requirements

During the installation of Active Directory, there are requirements that must be met by the servers that hold certain operations master roles.

Domain Naming Master Operations Master Role Holder

The Domain Naming Master operations master role must be available by RPC when installing a new domain in an existing forest.

Failure to access the Domain Naming Master operations master role holder during the installation of Active Directory for a new domain generates an error such as this:

```
To perform the requested operation, the Directory Service needs to
contact the
Domain Naming Master (server reskit.com). The attempt to contact it
failed.
"The RPC server is unavailable"
The text message is a win32 error message indicating why the network
operation to reskit.com failed
```

If you receive the preceding error message, it is recommended that you verify the following:

- Verify that network connectivity is present.
- Verify that the Domain Naming Master operations master role owner is available.
- Verify that LDAP records are correctly located on the DNS server for the domain controllers in the root domain.
- Verify that the domain controllers in the root domain are time synchronized within five minutes of each other.
- Verify that the IP configuration on domain controllers in the parent and child domains are correct and are using the same DNS server.

Note Flush the DNS cache to verify that the DNS entries are correct. The command to use is ipconfig /flushdns.

Removing Data in Active Directory After an Unsuccessful Removal of Active Directory

As part of the removal of Active Directory from a domain controller, the Active Directory Installation Wizard removes the configuration data for the domain controller from Active Directory. This data takes the form of the NTDS Settings object, which exists as a child of the server object (cn=NTDS Settings,cn=<*serverName*>,cn=Servers,cn=<*siteName*>,cn=Sites,cn=Configuration,dc=*forestRootDomain*). You can view these objects in the Sites container in the Active Directory Sites and Services console.

The attributes of the NTDS Settings object include data that represent how the domain controller is identified to its replication partners, the directory partitions that are maintained on the computer, whether or not the domain controller is a Global Catalog server. The NTDS Settings object is also a container that can have child objects that represent the domain controller's direct replication partners. This data is required for the domain controller to operate within the environment, but the NTDS Settings object is removed upon the removal of Active Directory.

If the NTDS Settings object is not properly removed during the process of removing Active Directory, the administrator can use the Ntdsutil tool to manually remove the NTDS Settings object. The following sections outline the procedure for removing the NTDS Settings object from Active Directory for a specific domain controller using the Ntdsutil tool. For more information about the available options, the administrator can type **help** on each Ntdstutil menu.

Caution Before manually removing the NTDS Settings object for any server, it is recommended that the administrator also check that replication has occurred because of the removal of Active Directory. Improper use of the Ntdsutil tool can result in partial or complete loss of Active Directory functionality.

For more information about the Ntdsutil tool, see Windows 2000 Support Tools on the Windows 2000 Server operating system CD.

Removing the Domain Controller Object

After you remove Active Directory from a domain controller, the object that represents the server in the Active Directory Sites and Services console remains.

This condition occurs because the server object is a container object that can hold child objects that represent configuration data for other services installed on your computer. For this reason, the wizard does not automatically remove the server object.

Warning If the server object contains any child objects named NTDS Settings, these objects represent the server as a domain controller and must be removed automatically when Active Directory is removed. If these objects are not removed automatically, or if removal of Active Directory cannot be performed (for example, on a computer that has malfunctioning hardware), these objects must be removed by using the Ntdsutil tool before you can delete the server object.

▶ **To remove the domain controller object**

1. In the Active Directory Sites and Services, double-click the Sites container to expand it, and then double-click the appropriate site object (the site in which the server resides) to expand the site object.
2. Double-click the Server container, right-click the server object, and then click **Delete**.
3. When you are prompted to confirm deleting the object, click **Yes**.

Note This process might not complete successfully for either of the following reasons:

If you receive a message that states the server is a container that contains other objects, before you continue verify that the appropriate services have been stopped.

If you receive a message that states the NTDS Settings object cannot be deleted, you might be attempting to delete an active domain controller. However, this message would only occur if the NTDS Settings object is the computer which you are trying to delete, otherwise the delete operation will succeed.

An administrator can safely delete the server object in the Active Directory Sites and Services console after all services have been removed and no child objects exist.

For more information about the Ntdsutil tool, see Windows 2000 Support Tools on the Windows 2000 Server operating system CD.

Troubleshooting Active Directory Installation Wizard Errors

When the Active Directory Installation Wizard performs the domain controller promotion process, it automatically creates its own log file: the Dcpromoui.log file. Specifically, it verifies and checks the following:

- The platform including the computer role (stand-alone client, member workstation, stand-alone server, member server, primary controller, or backup controller).
- The version number of the operating system.
- The role change of the computer. If the role change cannot occur, it returns an error message to the user; otherwise, it returns "0".
- Checks for the presence of at least one logical drive formatted with NTFS version 5. If the drive cannot be found, it returns an error message to the user.
- Checks if the user is a member of the Administrators group.

All important API calls are logged with the parameters and the error code returned. For example:

```
dcpromoui t:0x260 00325                Calling NetValidateName
dcpromoui t:0x260 00326                lpServer   : (null)
dcpromoui t:0x260 00327                lpName     : server.reskit.com
dcpromoui t:0x260 00328                lpAccount  : (null)
dcpromoui t:0x260 00329                lpPassword : (null)
dcpromoui t:0x260 00330                NameType   :
NetSetupNonExistentDomain
dcpromoui t:0x260 00331                Error 0x0 (!0 => error)
```

The error codes are typically Win32 error codes. For more information about the cause of each error according to API, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. Please note that not all error codes indicate a malfunction. In some cases, the error is the expected result, as in the following example:

```
dcpromoui t:0x260 00311                Calling DsGetDcName
dcpromoui t:0x260 00312                ComputerName : (null)
dcpromoui t:0x260 00313                DomainName   :
server.reskit.com
dcpromoui t:0x260 00314                DomainGuid   : (null)
dcpromoui t:0x260 00315                SiteGuid     : (null)
dcpromoui t:0x260 00316                Flags        : 0x1
dcpromoui t:0x260 00317                Error 0x54B (!0 => error)
dcpromoui t:0x260 00318                Trying again w/ rediscovery
dcpromoui t:0x260 00319                Error 0x54B (!0 => error)
```

This example shows that the error 0x54b is returned from two calls to DsGetDcName. 0x54b is ERROR_NO_SUCH_DOMAIN. This is a *good* result, because you are validating that the domain does not already exist in this context.

Most error conditions occur during the role change, because this is where high-dependency operations occur, such as DNS name resolution or Kerberos v5 authenticated network connections. The Active Directory Installation Wizard displays these errors. Errors returned from the API are divided as follows:

- The operation that was occurring at the time.
- The Win32 error text that occur during that specific operation.

For example:

```
The Directory Service failed to create the object
CN=Test,CN=Partitions,CN=Configuration,DC=server1,DC=reskit,DC=com.
Please check the event log for possible system errors.
The operation failed because:
The directory cannot validate the proposed directory partition name
because it does not hold a replica of the directory partition above the
proposed directory partition. "
```

In this example, the promotion operation was attempting to create a cross reference object in the directory partition for the new domain, but that operation failed because Active Directory cannot validate the specific domain name. The problem was that installation of the grandchild domain occurred before the child domain was replicated to the Global Catalog server. The corrective action is to force a replication to the Global Catalog server to allow the name validation to occur.

Usually, the problem is a network related issue. The first part of the error code, that is the operation that was happening at the time, helps you to isolate the problem. For example, you might see the message "can't open LDAP connection." The second part of the error code and the second text might help you understand why the error occurred, for example, "unable to authenticate."

For more information about cross reference objects, see "Name Resolution in Active Directory" in this book.

Examining the Dcpromo.log File

While Dcpromoui.log logs all the events from a graphical interface perspective , Dcpromo.log captures the creation and removal of Active Directory, SYSVOL trees and the installation, modification and removal of key services.

For more information about the Active Directory Installation and Removal process and why the following events are logged, see "Active Directory Data Storage" in this book.

Format of Dcpromo.log file

A typical line in Dcpromo.log is formatted as follows:

```
<time-stamp> <INFO field>: <description of operation>: <status code in
hexadecimal>
```

For example:

```
08/11 14:08:29 Request for promotion returning 0
```

The description of the promotion operation is usually self explanatory. The status code is NET API_STATUS or Win32 error code. A 0x0 indicates success, any other code indicates an error.

Examining the Dcpromo.log file from an installation of a child domain

During the gathering information phase, Dcpromo.log captures events that identify the DNS domain name, NetBIOS domain name, site name, and the location of the system volume.

```
08/16 16:21:07 [INFO] Promotion request for domain controller of new
domain
08/16 16:21:07 [INFO] DnsDomainName  user.reskit.com
08/16 16:21:07 [INFO]    FlatDomainName  USER0
08/16 16:21:07 [INFO]    SiteName  (NULL)
08/16 16:21:07 [INFO]    SystemVolumeRootPath  C:\WINNT\SYSVOL
08/16 16:21:07 [INFO]    DsDatabasePath  C:\WINNT\NTDS, DsLogPath
C:\WINNT\NTDS
08/16 16:21:07 [INFO]    ParentDnsDomainName  reskit.com
08/16 16:21:07 [INFO]    ParentServer  (NULL)
08/16 16:21:07 [INFO]    Account reskit\administrator
08/16 16:21:07 [INFO]    Options  2244
```

Verify the Ntds.dit file path and verify if SYSVOL is on a fixed drive and resides on an NTFS v5 volume.

```
08/16 16:21:07 [INFO] Validate supplied paths
08/16 16:21:07 [INFO] Validating path C:\WINNT\NTDS.
08/16 16:21:07 [INFO]    Path is a directory
08/16 16:21:07 [INFO]    Path is on a fixed disk drive.
08/16 16:21:07 [INFO] Validating path C:\WINNT\NTDS.
08/16 16:21:07 [INFO]    Path is a directory
08/16 16:21:07 [INFO]    Path is on a fixed disk drive.
08/16 16:21:07 [INFO] Validating path C:\WINNT\SYSVOL.
08/16 16:21:07 [INFO]    Path is on a fixed disk drive.
08/16 16:21:07 [INFO]    Path is on an NTFS volume
```

Ensure the name passed in is unique.

```
08/16 16:21:07 [INFO] Child domain creation -- check the new domain name
is child of parent domain name.
08/16 16:21:07 [INFO] Domain Creation -- check that the flat name is
unique.
```

Determine the site to place the domain controller and which domain controller to replicate from

```
08/16 16:21:22 [INFO] Start the worker task
08/16 16:21:23 [INFO] Request for promotion returning 0
08/16 16:21:23 [INFO] No source DC or no site name specified. Searching
for dc in domain reskit.com: ( DS_REQUIRED | WRITABLE )
08/16 16:21:23 [INFO] Searching for a domain controller for the domain
reskit.com

08/16 16:21:23 [INFO] Located domain controller reskit.com for domain
(null)

08/16 16:21:23 [INFO] No user specified source DC
08/16 16:21:23 [INFO] No user specified site
08/16 16:21:23 [INFO] Using site Default-First-Site-Name for server
reskit.com
```

Force a time synch so Kerberos v5 will authenticate successfully.

```
08/16 16:21:23 [INFO] Forcing a time synch with \\MARAK.reskit.com

08/16 16:21:17 [INFO] Reading domain policy from the domain controller
\\MARAK.reskit.com

08/16 16:21:17 [INFO] Stopping service NETLOGON

08/16 16:21:17 [INFO] Stopping service NETLOGON

08/16 16:21:17 [INFO] Configuring service NETLOGON to 1 returned 0
```

Prepare the SYSVOL.

```
08/16 16:21:17 [INFO] Creating the System Volume C:\WINNT\SYSVOL

08/16 16:21:17 [INFO] Deleting current sysvol path C:\WINNT\SYSVOL
08/16 16:21:22 [INFO] Preparing for system volume replication using root
C:\WINNT\SYSVOL
```

Ensure that the computer can be a member of the existing forest. If there is an existing forest, contact the Domain Name Master operations master role owner to verify that the domain does not already exist in the forest.

```
08/16 16:21:22 [INFO] Copying initial Directory Service database file
C:\WINNT\system32\ntds.dit to C:\WINNT\NTDS\ntds.dit

08/16 16:21:28 [INFO] Installing the Directory Service

08/16 16:21:28 [INFO] Calling NtdsInstall for user.reskit.com
08/16 16:21:28 [INFO] Starting the Directory Service installation
08/16 16:21:28 [INFO] Validating user supplied options
08/16 16:21:28 [INFO] Determining local site to enter
08/16 16:21:28 [INFO] Examining existing Enterprise Directory Service
08/16 16:21:30 [INFO] Configuring the local server to host the Directory
Service
```

Replicate the forest data

```
08/16 16:22:05 [INFO] Replicating the Directory Service schema container
08/16 16:22:09 [INFO] Replicating
CN=Schema,CN=Configuration,DC=reskit,DC=com: received 100 out of 1002
objects.
08/16 16:22:11 [INFO] Replicating
CN=Schema,CN=Configuration,DC=reskit,DC=com: received 199 out of 1002
objects.
08/16 16:22:13 [INFO] Replicating
CN=Schema,CN=Configuration,DC=reskit,DC=com: received 298 out of 1002
objects.
CN=Schema,CN=Configuration,DC=reskit,DC=com: received 1002 out of 1002
objects.

08/16 16:22:31 [INFO] Replicating the Directory Service configuration
container
08/16 16:22:33 [INFO] Replicating CN=Configuration,DC=reskit,DC=com:
received 99 out of 1236 objects.
08/16 16:22:35 [INFO] Replicating CN=Configuration,DC=reskit,DC=com:
received 145 out of 1236 objects.
0
08/16 16:22:53 [INFO] Replicating CN=Configuration,DC=reskit,DC=com:
received 1186 out of 1236 objects.
```

Create the new domain.

```
08/16 16:22:54 [INFO] Creating Partition: DC=user,DC=reskit,DC=com; 12
objects remaining.
08/16 16:22:54 [INFO] Creating Partition: DC=user,DC=reskit,DC=com; 11
objects remaining.
08/16 16:22:54 [INFO] Creating Partition: DC=user,DC=reskit,DC=com; 10
objects remaining.
08/16 16:22:55 [INFO] Creating Partition: DC=user,DC=reskit,DC=com; 0
objects remaining.
```

Move the current users and groups from the registry to Active Directory.

```
08/16 16:22:57 [INFO] Creating new domain security principals
08/16 16:23:00 [INFO] The Directory Service install is completing
08/16 16:23:02 [INFO] NtdsInstall for user.reskit.com returned 0
08/16 16:23:02 [INFO] DsRolepInstallDs returned 0
```

Set the local LSA policy to host the domain.

```
08/16 16:23:02 [INFO] Setting AccountDomainInfo to:
08/16 16:23:02 [INFO]    Domain: USER0
08/16 16:23:02 [INFO]    Sid:  S-1-5-21-776561741-789336058-842925246
```

Configure the domain and domain controller services to autostart when the computer is restarted.

```
08/16 16:23:03 [INFO] Configuring service w32time
08/16 16:23:04 [INFO] Configuring service w32time to 16 returned 0
08/16 16:23:04 [INFO] Configuring service NETLOGON
08/16 16:23:05 [INFO] Configuring service NETLOGON to 16 returned 0

08/16 16:23:05 [INFO] DsRolepSetRegStringValue on
SYSTEM\CurrentControlSet\Control\Lsa\MSV1_0\Auth2 to RASSFM returned 0

08/16 16:23:05 [INFO] Configuring service RPCLOCATOR
08/16 16:23:06 [INFO] Configuring service RPCLOCATOR to 16 returned 0
08/16 16:23:06 [INFO] Configuring service IsmServ
08/16 16:23:06 [INFO] Configuring service IsmServ to 16 returned 0
08/16 16:23:06 [INFO] Configuring service kdc
08/16 16:23:07 [INFO] Configuring service kdc to 16 returned 0
08/16 16:23:07 [INFO] Configuring service TrkSvr
08/16 16:23:08 [INFO] Configuring service TrkSvr to 16 returned 0
08/16 16:23:08 [INFO] Configuring service NETLOGON
08/16 16:23:08 [INFO] Configuring service NETLOGON to 144 returned 0
```

Create a trust relationship to the parent domain.

```
08/16 16:23:08 [INFO] Setting the LSA policy information
08/16 16:23:08 [INFO] Setting the LSA policy information from policy
\\MARAK.reskit.com

08/16 16:23:08 [INFO] Creating a parent trust relationship on domain
reskit.com
08/16 16:23:08 [INFO] Creating trusted domain object on parent
08/16 16:23:08 [INFO]    DnsDomain: user.reskit.com
08/16 16:23:08 [INFO]    Flat name: USER0
08/16 16:23:08 [INFO]    Direction: 3
08/16 16:23:08 [INFO]    Type: 2
08/16 16:23:08 [INFO]    Attributes: 0x0

08/16 16:23:09 [INFO] Creating a trust relationship with domain
user.reskit.com
08/16 16:23:09 [INFO] Creating trusted domain object on child

08/16 16:23:09 [INFO]    DnsDomain: reskit.com
08/16 16:23:09 [INFO]    Flat name: RESKIT
08/16 16:23:09 [INFO]    Direction: 3
08/16 16:23:09 [INFO]    Type: 2
08/16 16:23:09 [INFO]    Attributes: 0x400000
08/16 16:23:14 [INFO] Setting the computer's Dns computer name root to
user.reskit.com
```

Set the registry and file ACLs to become a domain controller.
08/16 16:23:23 [INFO] Setting security on the domain controller and Directory Service files and registry keys

```
08/16 16:23:27 [INFO] Securing users\.default
08/16 16:23:27 [INFO] Securing users\.default\software\microsoft\netdde
08/16 16:23:27 [INFO] Securing
users\.default\software\microsoft\protected storage system provider
08/16 16:23:27 [INFO] Securing machine\software
08/16 16:23:28 [INFO] Securing machine\software\classes
08/16 16:23:49 [INFO] Securing machine\software\microsoft\command
processor
08/16 16:23:49 [INFO] Securing machine\software\microsoft\cryptography
08/16 16:23:49 [INFO] Securing machine\software\microsoft\driver signing
08/16 16:23:49 [INFO] Securing
machine\software\microsoft\enterprisecertificates
08/16 16:23:49 [INFO] Securing machine\software\microsoft\netdde
08/16 16:23:49 [INFO] Securing machine\software\microsoft\non-driver
signing
08/16 16:23:49 [INFO] Securing machine\software\microsoft\ntds
08/16 16:23:49 [INFO] Securing machine\software\microsoft\ole
08/16 16:23:49 [INFO] Securing machine\software\microsoft\protected
storage system provider
08/16 16:23:49 [INFO] Securing machine\software\microsoft\rpc
08/16 16:23:49 [INFO] Securing
machine\software\microsoft\systemcertificates
08/16 16:23:50 [INFO] Securing
machine\software\microsoft\windows\currentversion\explorer
08/16 16:23:50 [INFO] Securing
machine\software\microsoft\windows\currentversion\group policy
08/16 16:23:50 [INFO] Securing
machine\software\microsoft\windows\currentversion\installer
08/16 16:23:50 [INFO] Securing
machine\software\microsoft\windows\currentversion\policies
08/16 16:23:50 [INFO] Securing
machine\software\microsoft\windows\currentversion\run
08/16 16:23:50 [INFO] Securing
machine\software\microsoft\windows\currentversion\runonce
08/16 16:23:50 [INFO] Securing
machine\software\microsoft\windows\currentversion\runonceex
08/16 16:23:50 [INFO] Securing
machine\software\microsoft\windows\currentversion\uninstall
08/16 16:23:50 [INFO] Securing
machine\software\microsoft\Windows NT\currentversion
08/16 16:23:50 [INFO] Securing
machine\software\microsoft\Windows NT\currentversion\accessibility
08/16 16:23:50 [INFO] Securing
machine\software\microsoft\Windows NT\currentversion\aedebug
```

```
08/16 16:23:50 [INFO] Securing
machine\software\microsoft\Windows NT\currentversion\asrcommands
08/16 16:23:50 [INFO] Securing
machine\software\microsoft\Windows NT\currentversion\classes
08/16 16:23:50 [INFO] Securing
machine\software\microsoft\Windows NT\currentversion\drivers32
08/16 16:23:50 [INFO] Securing
machine\software\microsoft\Windows NT\currentversion\efs
08/16 16:23:50 [INFO] Securing
machine\software\microsoft\Windows NT\currentversion\font drivers
08/16 16:23:50 [INFO] Securing
machine\software\microsoft\Windows NT\currentversion\fontmapper
08/16 16:23:50 [INFO] Securing
machine\software\microsoft\Windows NT\currentversion\image file
execution options
08/16 16:23:50 [INFO] Securing
machine\software\microsoft\Windows NT\currentversion\inifilemapping
08/16 16:23:50 [INFO] Securing
machine\software\microsoft\Windows NT\currentversion\perflib
08/16 16:23:50 [INFO] Securing
machine\software\microsoft\Windows NT\currentversion\perflib\009
08/16 16:23:50 [INFO] Securing
machine\software\microsoft\Windows NT\currentversion\profilelist
08/16 16:23:50 [INFO] Securing
machine\software\microsoft\Windows NT\currentversion\secedit
08/16 16:23:50 [INFO] Securing
machine\software\microsoft\Windows NT\currentversion\svchost
08/16 16:23:50 [INFO] Securing
machine\software\microsoft\Windows NT\currentversion\time zones
08/16 16:23:50 [INFO] Securing
machine\software\microsoft\Windows NT\currentversion\windows
08/16 16:23:50 [INFO] Securing machine\software\policies
08/16 16:23:50 [INFO] Securing machine\system
0
08/16 16:24:31 [INFO] Securing c:\winnt\ntds
08/16 16:24:31 [INFO] Securing c:\winnt\profiles
08/16 16:24:31 [INFO] Securing c:\winnt\repair
08/16 16:24:31 [INFO] Securing c:\winnt\security
08/16 16:24:31 [INFO] Securing c:\winnt\system32
08/16 16:24:40 [INFO] Securing c:\winnt\system32\autoexec.nt
08/16 16:24:40 [INFO] Securing c:\winnt\system32\cmos.ram
08/16 16:24:40 [INFO] Securing c:\winnt\system32\config
08/16 16:24:41 [INFO] Securing c:\winnt\system32\config.nt
08/16 16:24:41 [INFO] Securing c:\winnt\system32\dhcp
08/16 16:24:41 [INFO] Securing c:\winnt\system32\dllcache
```

```
08/16 16:24:51 [INFO] Securing c:\winnt\system32\grouppolicy
08/16 16:24:51 [INFO] Securing c:\winnt\system32\hpmon.dll
08/16 16:24:51 [INFO] Securing c:\winnt\system32\hpmon.hlp
08/16 16:24:51 [INFO] Securing c:\winnt\system32\ias
08/16 16:24:51 [INFO] Securing c:\winnt\system32\midimap.cfg
08/16 16:24:51 [INFO] Securing c:\winnt\system32\ntmsdata
08/16 16:24:51 [INFO] Securing c:\winnt\system32\spool
08/16 16:24:51 [INFO] Securing c:\winnt\sysvol
08/16 16:24:51 [INFO] Securing c:\winnt\sysvol\domain\policies
08/16 16:24:52 [INFO] Securing c:\winnt\tasks
08/16 16:24:52 [INFO] Securing c:\winnt\temp
08/16 16:24:52 [INFO] Securing LanManServer

08/16 16:24:57 [INFO] SetProductType to 2 [LanmanNT] returned 0
08/16 16:24:57 [INFO] The attempted domain controller operation has
completed
```

Returns a success or failure when finished running the Active Directory Installation Wizard.

```
08/16 16:24:58 [INFO] DsRolepSetOperationDone returned 0
```

For more information about the Active Directory installation and removal process, see "Active Directory Data Storage" in this book. For more information about cross-reference objects, see "Name Resolution in Active Directory" in this book.

Database Issues

If the domain controller cannot shut down in an orderly fashion (which usually means a power failure), the database is left out-of-date, because the most recent pages in memory were not written to the disk. Transaction logs are used to recover the database. Any change made to the database is also appended to the current log file, and its disk image is always kept up-to-date. The database change process is as follows:

1. Lsass.exe writes the change to a database page in the memory buffer.
2. Lsass.exe writes the change to the log file.
3. Lsass.exe waits for the log file to be flushed to disk.
4. Lsass.exe confirms the transaction.

If Active Directory halts, preventing the database from being successfully flushed to disk, the database performs a recovery on the next startup. Essentially, the database reads through the log files in order and reapplies changes until the database is made consistent and up-to-date.

The default log file name is Edb.log. The ESE can create a new log file when the current one fills up (noncircular logging). Or, it can overwrite the oldest file when the log reaches a specified number of files (circular logging). Noncircular logging consumes disk space until the administrator manually deletes old log files, following a backup or restart. It saves all database changes and never automatically deletes log files.

Note The default setting for Windows 2000 is circular logging turned on.

That directory routinely contains the following files:

- Edbxxxxx.log
- Edb.chk
- Res1.log
- Res2.log

Each of the files that has a .log extension is going to be created at exactly the same size of 10 megabytes (MB). Edb.log is the "current" log file. If circular logging is turned off, when the Edb.log file is full of transactions, it is renamed to Edb00001.log. This naming convention continues to increment by using hexadecimal notation. Thus, if there is a question as to the condition of the log files, that can be determined by checking to see whether an unbroken series of log file names exist.

Res1.log and Res2.log are "placeholders"—designed to reserve (in this case) the last 20 MB of disk space on this drive or directory. This is designed to give the log files sufficient room for a graceful shutdown if all other disk space are consumed. Note that if circular logging is set to on, running out of space for log files is not
an issue.

The checkpoint file, Edb.chk, is created by the Jet Database. Edb.chk stores the database checkpoint, so that it can replay logs starting with the generation containing the checkpoint, if needed. The Edb.chk file is a pointer in the log sequence that maintains the status between memory and the database file on disk. In the event of a failure, it indicates the point in the log file from which the information store needs to start the recovery. The Edb.chk file is essential for efficient recovery because if it didn't exist, the information store must attempt recovery by starting from the beginning of the oldest log file it found on disk and has to check every page in every log file to determine whether it had already been written to the database. This process, of course, is very time consuming, especially if the only goal is to make the database consistent.

Every time the database is opened, a check is performed to see if the database is up to date with the related checkpoint. For example, did the database fail before updating the checkpoint? If the database is not up to date, the log files are replayed from the point that the checkpoint file indicates. Jet logging and recovery can still recover a database without a checkpoint, but the checkpoint allows faster recovery by directing recovery to begin closer to logged operations that must
be redone.

Edb.chk is updated automatically by Jet when Jet notices that it has a specific amount of changes in the log files that are not forwarded to the checkpoint. Also, it is updated at the end of a recovery process. Finally, it is updated when you successfully shut down the system, to close the database.

Note The Directory Services Restore Mode–only restriction applies only to the functions that work directly on the database.

For more information about Active Directory Database operations, see "Active Directory Data Storage" in this book.

Ensuring File Integrity

To ensure the integrity of the database files, you might need to perform preventive procedures such as integrity check, move, repair, recovery, and defragmentation.

Using the Integrity Command to Detect Low Level Database Corruption

By using the **integrity** command, you can detect low level (binary level) database corruption. The **integrity** command invokes the **esentutl** command-line tool, which reads every byte of the data file. Therefore, depending upon the size of your data file, the process might take a considerable amount of time.

The **integrity** command also makes sure that the correct headers exist in the database itself and that all of the tables are functioning and are consistent. In short, it checks the integrity of the directory service data files. This is used while in Directory Services Restore mode. If errors are encountered, they are recorded on the log files.

The length of time for the **integrity** command to complete its operation depends on the type of hardware you are using and the size of your directory database. (In testing environments, the speed of two gigabytes (GB) per hour was considered to be normal.) However, when you carry out the command, an online graph displays showing the percentage completed.

Important To run the Ntdsutil tool and the subsequent **integrity** command, you must be in Directory Services Restore mode.

Following is a sample run of an integrity check by using the Ntdsutil tool:

```
:\>ntdsutil
ntdsutil: files
file maintenance: Integrity
Opening database .
Executing Command: C:\WINNT\System32\esentutl.exe /g
"C:\WINNT\NTDS\ntds.dit" /!
10240 /8 /v /x /o
Initiating INTEGRITY mode...
        Database: C:\WINNT\NTDS\ntds.dit
  Temp. Database: INTEG.EDB
failed to get 515126 buffers
checking database header
checking database integrity
Scanning Status  ( % complete )
0    10   20   30   40   50   60   70   80   90  100
          |----|----|----|----|----|----|----|----|----|----|
                checking SystemRoot
                SystemRoot (OE)
                SystemRoot (AE)
        checking system table
                MSysObjectsShadow
                MSysObjects
.               Name
                RootObjects
                rebuilding and comparing indexes
        checking table "datatable" (6)
                checking data
.....................          checking long value tree (24)
...             checking index "PhantomIndex" (125)
.               checking index "INDEX_000901FD" (122)
                checking index "INDEX_000900DE" (121)
                checking index "INDEX_00090089" (120)
                checking index "INDEX_00090573" (119)
                checking index "INDEX_00090073" (118)
                checking index "INDEX_00090571" (117)
                checking index "INDEX_0009056C" (116)
                checking index "INDEX_00090553" (115)
                checking index "INDEX_0009013A" (114)
                checking index "INDEX_00090138" (113)
                checking index "INDEX_00090330" (112)
                checking index "INDEX_00090030" (111)
                checking index "INDEX_00090013" (110)
                checking index "INDEX_00000013" (109)
                checking index "INDEX_0000000B" (108)
                checking index "INDEX_00000007" (107)
                checking index "INDEX_00000003" (106)
.               checking index "INDEX_00150003" (105)
```

```
                    checking index "LCL_ABVIEW_index00000409" (104)
                    checking index "INDEX_00090363" (103)
                    checking index "INDEX_00090303" (102)
                    checking index "INDEX_00090290" (101)
                    checking index "INDEX_000901FF" (100)
                    checking index "INDEX_000900DD" (99)
                    checking index "INDEX_00090085" (98)
                    checking index "INDEX_00090057" (97)
                    checking index "INDEX_0009001C" (96)
                    checking index "INDEX_000201CC" (95)
.                   checking index "INDEX_000200D2" (94)
                    checking index "INDEX_0002000D" (93)
                    checking index "INDEX_0000002A" (92)
                    checking index "INDEX_00000004" (91)
                    checking index "NC_Acc_Type_Name" (90)
                    checking index "PDNT_index" (89)
..                  checking index "INDEX_00090001" (88)
.                   checking index "INDEX_000901F6" (85)
                    checking index "INDEX_000902EE" (84)
                    checking index "INDEX_000904E1" (83)
                    checking index "INDEX_000201D5" (80)
                    checking index "INDEX_000902BB" (77)
                    checking index "INDEX_000903B4" (76)
                    checking index "INDEX_000200A9" (75)
                    checking index "INDEX_0009039D" (74)
                    checking index "INDEX_0009039A" (73)
                    checking index "INDEX_00090098" (72)
                    checking index "INDEX_00090395" (71)
                    checking index "INDEX_0009028F" (69)
                    checking index "INDEX_00090582" (66)
                    checking index "INDEX_00020078" (65)
.                   checking index "INDEX_00020073" (62)
                    checking index "INDEX_00090171" (60)
                    checking index "INDEX_00090167" (58)
                    checking index "INDEX_00090062" (56)
                    checking index "INDEX_00090261" (55)
                    checking index "INDEX_0009014E" (52)
                    checking index "INDEX_0009014D" (51)
                    checking index "INDEX_0009014C" (50)
                    checking index "INDEX_00090147" (49)
                    checking index "INDEX_00090141" (48)
                    checking index "INDEX_00090140" (47)
                    checking index "INDEX_0009012E" (42)
                    checking index "INDEX_00020013" (39)
.                   checking index "INDEX_0009030E" (36)
                    checking index "INDEX_00090008" (32)
                    checking index "INDEX_00090202" (25)
```

```
                checking index "Ancestors_index" (13)
.               checking index "DRA_USN_CREATED_index" (12)
                checking index "DRA_USN_index" (11)
.               checking index "del_index" (10)
                checking index "INDEX_00090002" (9)
..              checking index "NC_Acc_Type_Sid" (8)
                checking index "INDEX_00090092" (7)
                rebuilding and comparing indexes
        checking table "hiddentable" (16)
                checking data
                rebuilding and comparing indexes
        checking table "link_table" (14)
                checking data
                checking index "backlink_index" (15)
                rebuilding and comparing indexes
        checking table "MSysDefrag1" (123)
                checking data
                checking index "TablesToDefrag" (124)
                rebuilding and comparing indexes
        checking table "sdproptable" (17)
                checking data
                checking index "clientid_index" (19)
                checking index "trim_index" (18)
                rebuilding and comparing indexes

integrity check completed.
Operation completed successfully in 13.640 seconds.
Spawned Process Exit code 0x0(0)

If integrity was successful, it is recommended
 you run semantic database analysis to insure
 semantic database consistency as well.
```

Determining the Location of Database Files and Log Files

To find out the location of the data files, log files, and working directory, you can use the **info** command, which is part of the **ntdsutil** command-line tool. This command does the following:

- Analyzes and reports the free space for all disks installed on the computer.
- Reads the registry keys that contact the location of the Active Directory files and reports their values.
- Reports the sizes of the data file, working directory, and log file.

Following is sample output from running the **info** command:

```
file maintenance: Info

Drive Information:

        C:\ NTFS (Fixed Drive  ) free(2.9 Gb) total(3.9 Gb)

DS Path Information:

        Database   : C:\WINNT\NTDS\ntds.dit - 12.1 Mb
        Backup dir : C:\WINNT\NTDS\dsadata.bak
        Working dir: C:\WINNT\NTDS
        Log dir    : C:\WINNT\NTDS - 40.0 Mb total
                         res2.log - 10.0 Mb
                         res1.log - 10.0 Mb
                         REPAIR.TXT - 0.0 Kb
                         edb00001.log - 10.0 Mb
                         edb.log - 10.0 Mb
```

Moving the Database

When you move the database from one location to another location on the disk, you can use the Ntdsutil command-line tool in Directory Services Restore mode. For example, you might need to move a log file or the Ntds.dit file to another drive if corruption occurs on the previously assigned drive or directory. Specifically, the **move db to %s** command moves the Ntds.dit data file to the new directory specified by the "%s" and updates the registry keys so that the directory service restarts by using the new location.

▶ **To move the Active Directory database**

1. Back up Active Directory. Windows 2000 Backup natively supports backing up Active Directory while online. This occurs automatically when you select the option to back up everything on the computer in the Backup Wizard, or independently by selecting to back up the "System State" in the wizard.
2. Restart the domain controller, select the appropriate installation from the startup menu, and then press **F8** to display the **Windows 2000 Advanced Options Menu**.
3. Select **Directory Services Restore Mode**, and then press ENTER. To start the boot process again, press ENTER.
4. Log on by using the Administrator account by using the password defined for the Local Administrator account in the offline SAM.
5. From the **Start** menu, point to **Programs** and **Accessories**, and then click **Command Prompt**.
6. At the command prompt, type **ntdsutil**, and then press ENTER.

7. Type **files**, and then press ENTER.
8. Type **info**, and then press ENTER. This displays current information about the path and size of the Active Directory database and its log files. Note the path.
9. Establish a location that has enough drive space for the move database to be stored.
10. Type the following, and then press ENTER:

    move DB to <*drive*>:\<*directory*>

    where <*drive*> and <*directory*> is the path to the location that you established in the previous step.

    Note You must specify a directory path. If the path contains any spaces, the entire path must be surrounded by quotation marks (for example, move DB to "c:\new folder").

    The database named Ntds.dit is moved to the location that you specified.
11. Type **quit**, and then press ENTER. To return to the command prompt, type **quit** again.

    Note It is highly recommended that you make a backup immediately or else the restore operation does not retain the new file location.

    ---

12. Restart the computer normally.

You can also move the log files from location to another. Specifically, the **Move logs to %s** command moves the directory service log files to the new directory specified by **%s** and updates the registry keys so that the directory service restarts by using the new location.

Offline Defragmentation

Active Directory automatically performs *online* defragmentation of the database at certain intervals (by default, every 12 hours) as part of the Garbage Collection process. Online defragmentation does not reduce the size of the database file (Ntds.dit), but instead optimizes data storage in the database and reclaims space in the directory for new objects. It prevents data storage problems. Performing *offline* defragmentation creates a new, compacted version of the database file. Depending on how fragmented the original database file was, the new file might be considerably smaller.

▶ **To perform offline defragmentation of the Active Directory database**

1. Back up Active Directory. Windows 2000 Backup natively supports backing up Active Directory while online. This occurs automatically when you select the option to back up everything on the computer in the Backup Wizard, or independently by selecting to back up the "System State" in the wizard.
2. Restart the domain controller, select the appropriate installation from the startup menu, and press F8 to display the **Windows 2000 Advanced Options Menu**.
3. Select **Directory Services Restore Mode**, and then press ENTER. To start the boot process again, press ENTER.
4. Log on by using the Administrator account with the password defined for the Local Administrator account in the offline SAM.
5. Click **Start**, point to **Programs** and then to **Accessories**, and then click **Command Prompt**.
6. At the command prompt, type **ntdsutil**, and then press ENTER.
7. Type **files**, and then press ENTER.
8. Type **info**, and then press ENTER. This displays current information about the path and size of the Active Directory database and its log files. Note the path.
9. Establish a location that has enough drive space for the compacted database to be stored.
10. Type the following, and then press ENTER:

    compact to *<drive>*:*<directory>*

    where *<drive>* and *<directory>* is the path to the location that you established in the previous step.

Note You must specify a directory path. If the path contains any spaces, the entire path must be surrounded by quotation marks (for example, compact to "c:\new folder").

A new database named Ntds.dit is created in the path that you specified.

11. Type **quit**, and then press ENTER. To return to the command prompt, type **quit** again.
12. Copy the new Ntds.dit file over the old Ntds.dit file in the current Active Directory database path that you noted in step 8.
13. Restart the computer normally.

Performing a Soft Recovery of the Log Files

In the event that the power source failed unexpectedly, you can perform a "soft" recovery of the log files. Because transaction data is written to the log files before it is written to the data files, you can re-run the log files to reproduce the effects the transactions would have had if they were made to the data file. The **Recover** command in the Ntdsutil command line tool invokes the Esentutl command-line tool to perform this "soft" recovery. All of the log files are scanned to ensure that all committed transactions are made to the data file.

Note Soft recovery is performed automatically when the DSA starts if the previous shutdown was not clean.

Following is sample output of running the Recover command:

```
File maintenance: Recover
Executing Command: C:\WINNT\System32\esentutl.exe /r /8 /o
/l"C:\WINNT\NTDS" /s"
C:\WINNT\NTDS" /!10240


Initiating RECOVERY mode...
       Log files: C:\WINNT\NTDS
    System files: C:\WINNT\NTDS

Performing soft recovery...

Operation completed successfully in 4.717 seconds.


Spawned Process Exit code 0x0(0)

If recovery was successful, it is recommended
 you run semantic database analysis to insure
 semantic database consistency as well.
```

Repairing the Database

The database might need to be repaired due to a power outage. To repair the database, use the Ntdsutil command-line utility. Specifically, the **repair** command invokes the Esentutl tool, which performs a low level (binary level) of repair to the data file. This means that it repairs the database information of which the ESENT is aware.

Caution Caution must be exercised when using the **Repair** command because you can experience the random loss of data. The exact type of data that can be lost is not known. This loss can occur when there is data necessary for the safe operation of Active Directory that is not identified in the ESE.

Ensuring Database Integrity

Because Active Directory is implemented on a transacted database system, the ESE historically called Jet, log files are used to support rollback semantics to ensure that transactions are committed to the database.

The Ntdsutil tool includes a semantics checker that can be invoked by selecting the Semantic database analysis option. The role of the semantic checker is to check the integrity of the contents of the Active Directory database.

The tool is run during Directory Service Restore mode. Errors are written into dsdit.dmp .xx log files. A progress indicator indicates the status of the check.

The following are examples of the functions that can be performed:

- Reference count check. Counts all of the references from the data table and the link table to ensure they match the listed counts for the record. (For more information about data and link tables, see "Active Directory Data Storage" in this book.) It also ensures that each object has a GUID, distinguished name and nonzero reference count. If it is a deleted object, it ensures that it has a deleted time and date, but does not have a GUID or a distinguished name.
- Deleted object check. Ensures that it has a deleted time and date, and a special relative distinguished name.
- Ancestor check. Checks to determine if the current Distinguished Name Tag (DNT) is equal to the ancestor list of the parent and the current DNT.

- Security descriptor check. Checks for a valid descriptor, ensuring that it has a control field, and that the discretionary access control list is not empty. If there are deleted objects without a discretionary control access list, a warning is printed.
- Replication check. Checks the UpToDate vector in the directory partition head to ensure that the correct number of cursors exist. It also checks to see that every object has property metadata vector. For the instance type of the object, it checks the metadata, the up-to-dateness vectors, the sub references, and partial attribute.

▶ To perform Semantic database analysis

1. Back up Active Directory. Windows 2000 Backup natively supports backing up Active Directory while online. This occurs automatically when you select the option to back up everything on the computer in the Backup wizard, or independently by selecting to back up the "System State" in the wizard.
2. Restart the domain controller, select the appropriate installation from the startup menu, and press **F8** to display the **Windows 2000 Advanced Option Menu**.
3. Select **Directory Services Restore Mode**, and then press ENTER. To start the boot process again, press ENTER.
4. Log on by using the Administrator account with the password defined for the Local Administrator account in the offline SAM.
5. From the **Start** menu, point to **Programs** and **Accessories**, and then click **Command Prompt**.
6. At the command prompt, type **ntdsutil** and then press ENTER.
7. Type **Semantic database analysis**, and then press ENTER.
8. Type **Verbose on**, and then press ENTER. This displays the Semantic Checker.
9. Type **go,** and then press ENTER. The Semantic Checker is started without repairing any errors it encounters.

Note To repair the errors encountered, select the **Go Fixup** option.

---

10. Type **quit**, and then press ENTER. To return to the command prompt, type **quit** again.

Following is a sample of running the Semantic database analysis option with verbose mode turned on:

```
ntdsutil: Semantic database analysis
semantic checker: Verbose on
Verbose mode enabled.
semantic checker:  Go
Opening database .
....Done.

Getting record count...2371 records
Writing summary into log file dsdit.dmp.0
Records scanned:       2300
Processing records..Done.
```

Schema Issues

The most common schema issues encountered are with upgrading the schema. The first place to look when you receive an error message while upgrading is the Schupgr.log file located in the system32 folder.

Some common problems reported with the Schema upgrade process are the following:

- Insufficient rights error: Usually Schema Upgrade LDIF files contain changes for both the schema and configuration directory partitions. By default, only Schema Admins can modify objects in the schema directory partition, and only enterprise administrators or root domain administrators can modify objects in the configuration directory partition.

  Note The user must be logged in as a member of Schema Admins and Enterprise Admins because Schupgr.exe runs within the security context of the current logged-on user.

  The user needs to be logged on as a member of both because **schupgr** runs with current logged in user credentials. Sometimes the user is logged in as a member of both, but still reports an insufficient rights error. This is usually caused by the unavailability of a global catalog when the user logged in. Schema/Enterprise admin group membership evaluation requires a global catalog. If a global catalog is not available, those might not be in the user's token. Make sure the Global Catalog is running, and then log off and log on again.

An example of insufficient rights would be the following:

```
Opened Connection to BARDOK2
SSPI Bind succeeded
Found Naming Context DC=bardok2,DC=com
Found Naming Context CN=Schema,CN=Configuration,DC=itreskit,DC=com
Found Naming Context CN=Configuration,DC=itreskit,DC=com
Current Schema Version is 11
Upgrading schema to version 12
Converting DNs in file C:\WINNT\System32\sch12.ldf
ERROR: Failed to read current FSMO role owner: 50 (Insufficient
Rights)
```

- Error in importing: ldifde.exe generates two files, ldif.log and ldif.err (in case of an error only). Check the files to see which entry is in error and what error code is returned. You can then try to import it manually through ldp or a separate ldif file that contains only this entry. Next capture a sniffer trace, and figure out from the error information what the problem is.
- Schupgr runs but doesn't update the schema version: The version number. is updated as the last entry in the ldif file. If anything else fails before, the version is not updated. If no errors are reported from Schupgr.log file, and the version number is still not updated, it is usually a problem with ldifde. Make sure ldifde is running correctly. The instances when this occurs is when ldifde access violates when loading.
- Schupgr error: Cannot obtain schema version to upgrade to:

  You are missing a file that winnt32 would have copied to your computer when running schupgr to upgrade to the current build. To resolve the problem, run winnt32 to upgrade; it blocks detecting the schema mismatch and copies down the two to three files that you need. Then run schupgr.
- "ERROR: Failed to transfer the schema FSMO role: 52 (Unavailable)" in the schupgr log. This means the current domain controller is not the schema FSMO role owner, and trying to transfer the role from the other domain controller (whoever is the FSMO role owner) to this domain controller failed. The error unavailable can come for various reasons; it can't reach the other domain controller or the other domain controller didn't respond and so on. You can retry after some time if there is any temporary network problems and so on.

  To check who is the current schema fsmo role owner, use either the Schema snap-in or the ntdsutil tool to view the current Schema FSMO role owner.

Note If the previous suggestions do not yield the Schema FSMO role owner use the LDP or ADSIEdit tool to look at the fsmo-role-owner attribute on the schema container (cn=schema,cn=configuration,...). The *fsmoRoleOwner* attribute contains the name of the server that is the schema-fsmo role owner.

To increase the DS diagnostics logging level (which logs schema failures to the event log, sometimes providing clues as to why a schema operation is rejected) increase the value of the **Internal Processing** entry in HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS\Diagnostics to Level 3.

FSMOs

There are a number of operations that have a single operations master. They include activities such as updating the schema, creating new domains in a forest, issuing relative identifiers (RIDs) for security principal objects, updating domains running Windows NT 4.0 and earlier, and referencing objects in other domains.

The transfer of operations master roles can be done through the graphical user interface. However, if you have a domain controller that is unavailable, that is holding a token, the Ntdsutil tool can be used to actually force the transfer, or what is commonly referred to as seizure. The tool also allows you to do metadata cleanup. Thus, if you delete a domain or delete a domain controller without cleaning up the data directly used in the Active Directory Installation Wizard, you can use Ntdsutil to clean up that metadata. The Ntdutil tool can also be used to do domain precreation. The benefit here is that you can precreate domains by using a specific administration account, and then have other users actually promote and create domains without having to have the same set of permissions that you used to actually precreate that domain.

Relative Identifier Master Operations Master Role Holder

The Relative ID (RID) operations master role holder must be available when a server needs to be supplied RIDs. The Dcpromoui.log file that follows shows the error that occurs when the RID master operations master role holder is not accessible. This particular error is captured by the Dcpromoui.log file during domain controller promotion, but is in fact unrelated. The error occurs when the domain controller is out of RIDs and the RID operations master role holder is unavailable.

Note If the server still has RIDs, the Relative ID operations master role holder is not required to be available for account creations.

To troubleshoot this error, determine which server is hosting the RID master operations master role holder and confirm that it has network connectivity by using the Netdiag tool. Also, review the directory service log in Event Viewer for RID Master–related errors.

```
dcpromoui t:0x398 00279                Enter DS::JoinDomain
dcpromoui t:0x398 00280                   Enter massageUserName
dcpromoui t:0x398 00281                   Exit  massageUserName
dcpromoui t:0x398 00282                   Calling NetJoinDomain
dcpromoui t:0x398 00283                   lpServer      : Reskit
dcpromoui t:0x398 00284                   lpDomain      : reskit-rdp.com
dcpromoui t:0x398 00285                   lpAccountOU   : (null)
dcpromoui t:0x398 00286                   lpAccount     : reskit-
rdp.com\administrator
dcpromoui t:0x398 00287                   fJoinOptions : 0x23
dcpromoui t:0x398 00288                   Error 0x2010 (!0 => error)
dcpromoui t:0x398 00289                Exit  DS::JoinDomain
dcpromoui t:0x398 00290                Exception caught
dcpromoui t:0x398 00291                catch completed
dcpromoui t:0x398 00292                handling exception
dcpromoui t:0x398 00293                Error Joining Domain
dcpromoui t:0x398 00294                The directory service was unable
to allocate a relative identifier.
dcpromoui t:0x398 00295                Enter State::SetOperationResults
result FAILURE message: The directory service was unable to allocate a
relative identifier.
dcpromoui t:0x398 00296                Exit  State::SetOperationResults
result FAILURE message: The directory service was unable to allocate a
relative identifier.
```

Operations Master and Duplicate Operations Master Role Holders

In Windows NT 4.0, if you received a duplicate primary domain controller in a domain on a network as a result of someone promoting a backup domain controller when the primary domain controller is offline, it resolves itself eventually when the downed primary domain controller came back on line because of the conflict in the NetBIOS name for the primary domain controller in the domain, which is unique.

In Windows 2000, the primary domain controller FSMO is not quite as important, but the main point is that you do not want it duplicated or you might end up with conflicts. If you had duplicate RID pool role owners, you might end up with duplicate SIDs.

The following checks are done to minimize the possibility of having duplicate RID pool role owners:

- The server is synchronized with others before seizing the RID master role.
- The global RID (available) pool state is replicated urgently to maximize chances that candidates for the new RID FSMO are up-to-date.
- If the RID master allocates a RID pool to a domain controller that overlaps with the RID pool of another domain controller, the domain controller whose pool overlaps with the new pool notices this when this information replicates to it, and then proceeds by invalidating its current pool and requesting a new RID pool. This prevents the domain controller from issuing further duplicates and quickly "moves" all domain controllers that have overlapping pools to acquire fresh pools that do not overlap.
- The operating system contains checks to detect and handle instances of duplicate RIDs.

One situation that has been identified as a possible cause of allocation of duplicate relative identifier pools is if the relative ID master role has been seized while the original relative ID master is still operational but has been temporarily disconnected from the network. In normal practice, after one replication cycle, the relative ID master role is assumed by one and only one domain controller, but it might be possible that before the role ownership is resolved, two different domain controllers might each request a new relative ID pool and they might be allocated the same RID pool.

The Ntdsutil tool contains an option, Security account management to detect and clean all instances of duplicate SIDs. Accounts with duplicate SIDs are deleted.

▶ **To detect and clean up all instances of duplicate SIDs**

1. Back up Active Directory.

   Windows 2000 Backup natively supports backing up Active Directory while you are online. This occurs automatically when you select the option to back up everything on the computer in the Backup Wizard, or independently when you select to back up the "System State" in the wizard.

2. Restart the domain controller, select the appropriate installation from the startup menu, and then press F8 to display the **Windows 2000 Advanced Options Menu**.

3. Select **Directory Services Restore Mode**, and then press ENTER. To start the boot process again, press ENTER.
4. Log on by using the Administrator account with the password that is defined for the Local Administrator account in the offline SAM.
5. Click **Start**, point to **Programs** and then to **Accessories**, and then click **Command Prompt**.
6. At the command prompt, type **ntdsutil** and then press ENTER.
7. Type **Security account management**, and then press ENTER.
8. Type **Connect to server** <*server name*>, and then press ENTER.
9. At the **Security Account Maintenance** prompt, type **Cleanup Duplicate SID**, and then press ENTER.

   A duplicate SID cleanup operation is carried out and all results are logged in the Dupsid.log file that is located in the directory from which you ran the Ntdsutil tool.
10. Type **quit**, and then press ENTER. To return to the command prompt, type **quit** again.

In general, if any operations master role holders get duplicated, this resolves itself eventually by replication. The newer role owner has the change in its directory database with a higher USN, and thus overwrites the previous role owner when new role owner replicates to the directory database. The only damage that might occur is that it writes to the older role owner before replication solves the problem of duplicate role owners.

The best practice is to never let duplicates happen. However, if duplicate RIDs occur, you can fix it by putting the older role owner in the same site as the new role owner and immediately forcing replication to occur.

For more information about FSMOs and troubleshooting FSMOs, see "Managing Flexible Single-Master Operations" in this book.

Replication Issues

Active Directory knows only two kinds of replication: synchronous RPC replication, and asynchronous replication by using the Inter-site Messaging (ISM) service. When the directory uses the ISM service, it sends and receives messages in an asynchronous manner.

To summarize, the following types of transports are recognized:

- Synchronous RPC: trusted, transport-provided Kerberos v5 security.
- Asynchronous completion: untrusted transport, directory provided certificate security.

Note One important requirement of the ISM transport module: that it work between disconnected networks. This is not an implementation requirement of the ISM transport module, but a feature requirement. The kinds of environments in which you deploy do not have a directly connected trusted path. Instead, the message must pass through gateways; part of the message travelling over an unsecured link. Whatever transport works in this environment must tolerate long latencies, must be routable through gateways, and must be able to be stored and forwarded in case one or more gateways are unavailable.

▶ To troubleshoot mail-based replication

1. Check the event log for relevant messages. Possible errors can be problems with the KCC in constructing the topology, problems from the SMTP service (SMTPSVC) in delivering the mail, problems from the ISM service in reading the messages, or problems from the NTDS in decoding and applying the mail.
2. Verify that the KCC setup is on SMTP-based connections between the servers in the sites you want.

   This indicates that the site links are what you expect.

3. Verify that the replication links are established by using the correct transport. Do this by looking for connections that have the SMTP transport associated with them. You might also use **repadmin /showreps** and to look for the "via SMTP" designation.

   Note that the KCC does not create a connection by using SMTP until the following criteria are met:

   - The servers are in sites connected by SMTP site links.
   - The site link path between the sites is cheaper by using SMTP than IP.
   - You are not attempting to replicate writable replicas of the same domain over SMTP. (This is not supported.) Thus, Global Catalog replication is supported as well as replication of the configuration and schema directory partitions (which are not domains).
   - Each server is configured to receive mail. The IIS must have been installed on both servers.

     Note that communication between sites is by definition between bridgehead servers. The KCC chooses the bridgehead for each site unless they are set explicitly. Verify that if you are using explicit bridgeheads, that they hold the domain you are trying to replicate.

4. Before you move a server into a site connected only by mail-based replication, you want to verify that the Domain Controller Certificate is present. A certificate authority must be installed in your enterprise on one of the domain controllers. It must be an "Enterprise Certificate Authority." After some time (up to 8 hours), all the domain controllers in the domain are "auto enrolled" with a Domain Controller Certificate. You can verify whether a particular computer holds the domain controller Certificate by using the Certificates snap-in (look under Personal), or by using the command **repadmin /showcert**.
5. For mail-based replication, you need to decide if mail routing is necessary. If the two servers have direct IP connectivity and can send mail directly to each other, no further configuration is required. However, if the two domain controllers must go through mail gateways to deliver mail to each other, you must configure the domain controller to use the mail gateway. Typically this is done by setting a "Smart Host" in the Default SMTP Virtual Server, under IIS in the computer configuration Snap-in.

6. Determine whether mail-based replication is succeeding or not by checking the display from *repadmin/showreps*. This shows the current error code and the last success time. If the current error code is "request is pending" and the last success has been more than an hour, you can suspect that your mail is being delayed or not delivered. Note that it is normal for delivery of mail to not happen immediately, because the communication is store and forward, not a direct connection.

   The first thing you want to check is whether the SMTPSVC is picking up and delivering the mail. You can check the Inetpub\mailroot\Queue directory on the destination, any gateway computers, and the source. The Queue is the queue of outgoing mail. If the Queue directory contains a large number of files, this usually means the SMTPSVC is backed up or unable to process all the mail. A workaround is to run the following:

   - net stop iisadmin
   - net start stmpsvc
   - net start ftpsvc
   - net start wwwsvc

   This causes all pending mail to be sent.

   - You can verify mail based directory synchronization as follows. Consider computers A and B. Using the Windows 2000 Shell Explorer on computers A and B, bring up a view of the <ntds database directory>\Drop directory. Suppose A pulls changes from B. On A, force a synchronization of the connection to B using the Active Directory Sites and Services snap-in. System A requests changes from B using a request mail message. You should see one or more temporary mail files appear in the Explorer Drop window on B. Then it should disappear as it is processed. Soon you should see a temporary mail file appear in the Explorer Drop window on A. This is the response mail message containing the data. It should also disappear as it is processed.
   - The next thing that you can check is whether the Intersite Messaging Service (ISM service) is reading the mail. If you check the directory <ntds logfile directory>\Drop on both the destination and the source. By default, the log file directory is %windir%\ntds, unless it was moved by using Ntdsutil. The Drop directory contains *incoming* mail. If this directory contains a large number of mail messages, it indicates that either the service is stopped, or that there is too much mail coming in for the service to handle. If the service is not running, it should be restarted. If mail is coming in too fast, the replication period for mail-based replication should be increased. As a last resort, it is possible to stop the ISM service and delete all the messages in the Drop directory, and then restart the service. This should give the service a chance to get out from under the huge backlog. Eventually, the messages are retransmitted by the senders.

7. Another thing to check is whether there are delivery problems. Please verify that each leg of the mail route can contact its next hop directly by IP. In addition to IP connectivity, it is recommended that you verify that each server can resolve the name of the server it is trying to reach. The mail address that a domain controller uses to contact another domain controller is its "guid-based name," which looks like <guid>._msdcs.<forest-root-dns-name>. It is recommended that you verify that you can ping this name and receive a successful response.
8. When mail cannot be delivered, it is returned as a "Delivery Status Notification" by the SMTPSVC. These DSNs are logged to the event log when logged is set high enough. In order to see DSNs, you must set the Active Directory Diagnostic level of Intersite Messaging Service to level 1 and restart the SMTP service. For more information about Active Directory diagnostic levels and how to set them, see "Active Directory Diagnostic Levels" earlier in this chapter.

Replication Event Viewer Entries

The types of entries that might appear in Directory Service log in Event Viewer that pertain to replication include errors such as the following:

- KCC
- Long-running Inbound Replication
- Conflict with Certificate Services
- RPC Unavailable
- Unknown User Name/Bad Password
- Automatic Topology Generator Was Unable To Complete the Topology

Knowledge Consistency Check Replication Errors

Error: ID1311, from Event Source "NTDS KCC") in the Directory Service log

You might see the following entry (with ID 1311 from event source "NTDS KCC") in the Directory Service log.

```
The Directory Service consistency checker has determined that either (a)
there is not enough physical connectivity published via the Active
Directory Sites and Services Manager to create a spanning tree
connecting all the sites containing the partition DC=mycorp,DC=com, or
(b) replication cannot be performed with one or more critical servers in
order for changes to propagate across all sites (most often due to the
servers being unreachable).
For (a), please use the Active Directory Sites and Services Manager to
do one of the following:
1. Publish sufficient site connectivity information such that the system
can infer a route by which this Partition can reach this site.  This
option is preferred.
2. Add an ntdsConnection object to a Domain Controller that contains the
partition DC=mycorp,DC=com in this site from a Domain Controller that
contains the same partition in another site.
For (b), please see previous events logged by the NTDS KCC source that
identify the servers that could not be contacted.
```

This behavior can occur if the KCC has determined that a site has been orphaned from the replication topology.

One computer in a specific site owns the role of creating inbound replication connection objects between bridgehead servers from other sites. This domain controller is known as the Inter-Site Topology Generator. While analyzing the site link and site link bridge structure to determine the most cost-effective route to synchronize a naming context between two points, it might determine that a site does not have membership in any site link and therefore has no means to create a replication object to a bridgehead server in that site. The first site in Active Directory (named "Default-First-Site-Name"), is created automatically for the administrator. This site is a member of the default site link ("DEFAULTIPSITELINK"), which is also created automatically for the administrator, and is used for RPC communication over TCP/IP. If the administrator creates two additional sites ("site1" and "site2" for example), the administrator must define a Site Link that each site is going to be a member of before they can be written to Active Directory.

However, the administrator can open the properties of a site link and modify which sites reside in the site link. If the administrator were to remove a site from all site links, the KCC displays the error message listed earlier to indicate that a correction needs to be made to the configuration.

Note When the KCC is displaying this message, it is in a mode where it does not remove any connections. Normally, the KCC cleans up old connections from previous configurations or redundant connections. Thus, you might find extra unexpected connections during this time. The solution is to correct the topology problem so that the spanning tree can be formed.

This error might also occur when replication has failed from a particular bridgehead server in a site and no other bridgehead servers are available. For more information about bridgehead servers, see "Active Directory Replication" in this book.

Examples of Replication Event Viewer Messages

Examples of errors that involve long-running inbound replication and Certificate Services conflicts and other replication Event Viewer messages:

Note The error messages are not displayed by default , and the diagnostic level must be increased first. The frequency of these messages appearing in Event Viewer are rare. However, they are useful for troubleshooting and optimizing performance.

```
Event 1580
Severity=Informational
A long running inbound replication has finished.  The elapsed time was
%1 minutes.
The operation was %2, and the options were %3. The status of the
operation was:
The operation specific arguments are:
Parameter 1:
Parameter 2:
Parameter 3:
Parameter 4:
%n
```

This data is for information and may be useful in tuning the replication performance of the system. Since only one inbound replication may occur at a time, long running replications delay other replications from coming in in a timely manner. This system has been delayed from receiving other directory updates because this replication went on as long as it did. A long running replication may indicate a large number of updates, or a number of complex updates (DN-valued attributes) occurring at the source server. Performing these updates during non-critical times may prevent replication delays during important times.

A long running replication is normal in the case of adding a new replica to a system, either because of installation, Global Catalog promotion, or connection creation by the KCC. A long running replication may also occur for a system that has been down, or a partition that has been out of touch for an extended period.

The record data is the status code.

.
logging_level: 1

Event 1579
Severity=Warning
Due to contention with the Certificate Services for resources, replication was stalled for %1 minutes, %2 seconds. It took an unusually long time to prepare an asynchronous replication message for transmission. This condition should be transient. If this issue persists, please contact Microsoft Product Support Services for assistance.

.
logging_level: 0

Event 1574
Severity=Warning
Due to contention with the Security Descriptor Propagator for resources, inbound replication was stalled for %1 minutes, %2 seconds. This condition should be transient. If this issue persists, please contact Microsoft Product Support Services for assistance.

.
logging_level: 0

```
Event 1575
Severity=Informational
One or more new attributes has been added to the partial attribute set
for partition %1. A full synchronization will be performed from source
%2 on the next periodic synchronization.
.
logging_level: 1

Event 1560
Severity=Informational
A new replica for partition %1 has been added to this server.  This
server will now perform a full synchronization from source %2 with
options %3.
.
logging_level: 1

Event 1561
Severity=Informational
The user has requested a full synchronization of partition %1 from
source %2 with options %3.
.
logging_level: 1

Event 1562
Severity=Informational
The full synchronization of partition %1 from source %2 with options
%3 in being continued.
.
logging_level: 1
```

RPC Server Is Unavailable

The "RPC server is unavailable" errors typically indicate that the computer is down.

Note If the name can not be resolved, Active Directory generates an error that explicitly says "The name can not be resolved."

For more information on RPC server unavailable error messages, see "Name Resolution" earlier in this chapter.

Unknown User Name/Bad Password

Regarding the "Unknown user name/bad password" issue, run the **repadmin** command on the domain controller.

For example, run the following: **repadmin /showmeta CN=<*domain controller name*>,OU=Domain Controllers,DC=<*domain name*>,DC=<*domain name*>.**

If the version numbers on the *unicodePwd* attribute of either object is sufficiently different between the two domain controllers (that is greater than or equal to two), it might mean that the passwords are not synchronized, that replication hasn't occurred, and therefore that the domain controllers can no longer authenticate to each other. If that is the case, reset the computer account passwords.

Note The Dcdiag tool contains a computer object metadata comparison test that automates the steps described in the preceding section. For more information about the Dcdiag tool, see "Domain Controller Issues" earlier in this chapter.

Replication Failing with Access Denied

Other security errors might also fall under this diagnosis, such as "Wrong Target Name" or "Cannot locate domain controller." It is important to understand that a circular dependency exists between replication and distributed security using Kerberos v5. Replication is authenticated using Kerberos v5, and Kerberos security principals are replicated using replication.

Usually, if two domain controllers cannot replicate with each other because of a security problem, it is because one or both of these domain controllers haven't replicated the information needed to identify each other.

When Active Directory is installed on a computer, that computer becomes a domain controller (refer to this as the target), it joins the enterprise by communicating with an existing domain controller (refer to this as the source). Until the target restarts and configures itself, the source is the only domain controller in the enterprise with knowledge of the target.

Note The source becomes a single point of failure for a brief period until it replicates its knowledge off to other domain controllers. This is one reason that the removal of Active Directory from a domain controller requires successful replication outbound before attempting to run the Active Directory Installation Wizard. Under unusual circumstances, if a computer completely fails after being a source, knowledge of a new domain controller can be lost to the rest of the enterprise.

Thus, if a third domain controller attempts to replicate with the new system, replication can fail with a security error because knowledge of the new system has not replicated throughout the enterprise. Replication of knowledge of the new system replicates in the Configuration directory partition, but knowledge of the Kerberos v5 principal for the new system replicates in the Domain directory partition. Thus there can be a temporary disconnect between when a third computer knows of a new computer, and when that computer can authenticate with them.

Therefore, when replication receives the error "Access Denied," it should be verified whether knowledge of newly promoted domain controllers has replicated throughout the enterprise. In the event that the enterprise is having replication problems, it might be preventing the computer account object of a newly promoted domain controller from being replicated.

Automatic Topology Generator Was Unable to Complete the Topology

You also might see a message in the Directory Service log in Event Viewer that says "Automatic topology generator was unable to complete the topology for *<distinguished name of the site>*." This message indicates that there is an exception in the KCC.

To log more information, increase the value of the **Internal Processing** registry entry to 3 and wait 15 minutes:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS\Diagnostics**Internal Processing**

Alternatively, you can also run **repadmin /kcc**, and reset the value of the registry entry to 0.

Note This method must also be used whenever replication returns the status of "Internal error." The Internal Processing level should be set on both of the computers that are replicating data.

Monitoring the Replication Links

You can use the Replication Administration (Repadmin) command-line tool to monitor the current links for a specific domain controller, including the domain controllers that are replicating to and from the current domain controller. By viewing these links in Repadmin, you can see the replication topology as it exists for the current server. By seeing the replication topology, you can check replication consistency between replication partners, monitor replication status, and display replication metadata.

More importantly, you can also manipulate replication topology by forcing specific replication events and triggering KCC recalculation. However, you must force replication only when you know that a domain controller is offline, or when network connections are not working. During normal operation, the KCC automatically manages the replication topology for each directory partition on domain controllers.

For example, to track which domain controller received a particular replicated change, you can enter the following:

```
repadmin/showmeta "CN=JSmith,OU=PR,OU=Marketing,DC=Reskit,DC=com" <name
of domain controller>
```

where *<name of domain controller>* is the host name of the target domain controller for which you are tracking replicated changes for "JSmith" in the "PR" OU in the "Marketing" OU, in the "Reskit.com" domain. The output resulting from this command shows the update sequence number (USN), the originating DSA, USN, date and time, version number, and the replicated attribute.

Domain Mode Changes

To determine whether directory updates are being replicated to all domain controllers, you need to use the Repadmin and Ldp tools. One primary example of operations that can affect replication integrity are domain mode changes.

For example, the domain mode change is propagated through normal replication. There is an attribute called *ntMixedMode* for objects of class *domainDNS* (for example, the domain). Nonzero indicates the mode is mixed, zero indicates native mode. You can determine if the domain mode change has propagated by checking this attribute on all the domain controllers in the domain.

You can use the Repadmin tool to view the replication topology as seen from the perspective of each domain controller and the ISM matrix information. In addition, the Repadmin tool can be used to manually create the replication topology (although in normal practice, this is not necessary), to force replication events between domain controllers and to view both the replication metadata and up-to-date vectors.

Note During the normal course of operations, there is no requirement for manual creation of the replication topology. Incorrect use of this tool might adversely impact the replication topology. The major use of this tool is to monitor replication so that problems such as offline servers or unavailable LAN/WAN connections, can be identified.

Repadmin

Repadmin.exe is a command-line tool that lets you view and change replication status on domain controllers when you need to diagnose and troubleshoot replication between Windows 2000–based domain controllers. You can use Repadmin to view the current replication topology, manually create the replication topology, and force replication events.

Note During normal operation, the KCC performs automatic replication topology generation, and manual management of the replication topology is not required.

For information about using Repadmin, see the /Support Tools Help on the Windows 2000 operating system CD.

Viewing the Connections for a Server

The Repadmin tool can be used to show the current links for a specific domain controller, including the domain controllers that are replicating to and from this domain controller. By viewing these links in Repadmin, you can see the replication topology as it currently exists for that server. The links might be unreachable, which prevents any new links from that server from being added.

When you use Repadmin to view the links for a domain controller, you are viewing the replication partners that the KCC is currently using for that server. If you can see a connection object in the Sites container, you might not see that connection represented in Repadmin; for example, to view the current replication partners for a particular server.

▶ **To view current replication partners for a server**

- At the command prompt, type the following:

  repadmin /showreps server ***<nc> <guid-of source>***

Forcing Replication Between Replication Partners

There are four methods that can be used to initiate replication between direct replication partners. Three methods use administrative tools. The fourth method involves writing a Visual Basic script. For instructions about how to use the script, see the Microsoft Knowledge Base link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. Follow the links for article Q232072.

For each of the following methods that are described, the "source" server describes the domain controller that replicates changes to a replication partner. The "destination" domain controller receives the changes.

Initiating Replication by Using Active Directory Sites and Services

In the Active Directory Sites and Services MMC console, right-click a connection object, and then click **Replicate Now**. (For information about how to initiate replication by using Active Directory Sites and Services, see Windows 2000 Server Help.)

Initiating Replication by Using Repadmin

Repadmin is a command-line tool that is included in the Support directory on the Windows 2000 operating system CD. You can use Repadmin to first determine the directory replication partners of the destination server and then issue a command to synchronize the source server with the destination by using the object GUID of the source server.

▶ **To use Repadmin to force replication between two servers**

1. At a command prompt, type the following:

   repadmin /showreps <*destination_server_name*>

   Note Refer to the Repadmin example that follows as you walk through these steps.

2. Under the Inbound Neighbors section of the output, find the directory partition that needs synchronization and locate the source server with which the destination is to be synchronized. Note the *objectGuid* value of the source server.
3. Initiate replication by entering the following command:

   ```
   repadmin /sync <directory_partition_DN>
   <destination_server_name><source_server_objectGuid>
   ```

For example, to initiate replication on DC1 of the domain directory partition support.reskit.com so that changes are replicated from DC2, use the following command:

```
repadmin /sync dc=support,dc=microsoft,dc=com DC1 d2e3badd-e07a-11d2-
b573-0000f87a546b
```

If the command is successful, Repadmin.exe displays the following message:

```
ReplicaSync() from source: d2e3badd-e07a-11d2-b573-0000f87a546b, to
dest: DC1 is successful.
```

Optionally, you can use the following switches at the command prompt:

```
 - /force: Overrides the normal replication schedule.
 - /async: Starts the replication event. (Repadmin.exe does not wait for
the replication event to finish.)
```

The following is an example of running **repadmin**:

```
C:\WINNT\idw>repadmin /showreps
repadmin /showreps
Washington\NTGROUP1
DSA Options : (none)
objectGuid  : 3a34efb9-f828-11d2-a68d-00c04fb9d14e
invocationID: 39216b7e-f828-11d2-8128-00105a68cf71

==== INBOUND NEIGHBORS ======================================

DC=dsysreskit,DC=reskit,DC=microsoft,DC=com
    Bldg\NTGROUP2 via RPC
        objectGuid: cc6d76a3-a71a-11d2-bbd0-00105a24d6db
        Last attempt @ 1999-05-10 22:47.33 failed, result 1722:
            The RPC server is unavailable.
        Last success @ 1999-05-10 22:02.32.
        6 consecutive failure(s).

CN=Schema,CN=Configuration,DC=reskit,DC=microsoft,DC=com
    Washington\WORKSTATION1 via RPC
        objectGuid: ed8a3ba0-d439-11d2-99e7-08002ba3ed3b
        Last attempt @ 1999-05-10 22:47.32 was successful.
    Washington\RESKIT-DC-07 via RPC
        objectGuid: 6a7ff635-baeb-11d2-8fda-0008c709d19e
        Last attempt @ 1999-05-10 22:47.33 was successful.
    Bldg\NTGROUP2 via RPC
        objectGuid: cc6d76a3-a71a-11d2-bbd0-00105a24d6db
        Last attempt @ 1999-05-10 22:47.33 failed, result 1722:
            The RPC server is unavailable.
        Last success @ 1999-05-10 22:02.32.
        6 consecutive failure(s).
```

```
CN=Configuration,DC=reskit,DC=microsoft,DC=com
    Washington\WORKSTATION1 via RPC
        objectGuid: ed8a3ba0-d439-11d2-99e7-08002ba3ed3b
        Last attempt @ 1999-05-10 22:47.32 was successful.
    Bldg\NTGROUP2 via RPC
        objectGuid: cc6d76a3-a71a-11d2-bbd0-00105a24d6db
        Last attempt @ 1999-05-10 22:47.33 failed, result 1722:
            The RPC server is unavailable.
        Last success @ 1999-05-10 22:02.32.
        6 consecutive failure(s).
    Washington\RESKIT-DC-07 via RPC
        objectGuid: 6a7ff635-baeb-11d2-8fda-0008c709d19e
        Last attempt @ 1999-05-10 22:48.26 was successful.

==== OUTBOUND NEIGHBORS FOR CHANGE NOTIFICATIONS ============

DC=dsysreskit,DC=reskit,DC=microsoft,DC=com
    Washington\DSYSRESKIT0 via RPC
        objectGuid: abbf2810-f51b-11d2-84a0-00105a68cf71
    Washington\RESKIT-DC-07 via RPC
        objectGuid: 6a7ff635-baeb-11d2-8fda-0008c709d19e

CN=Schema,CN=Configuration,DC=reskit,DC=microsoft,DC=com
    Washington\DSYSRESKIT0 via RPC
        objectGuid: abbf2810-f51b-11d2-84a0-00105a68cf71
    Washington\RESKIT-DC-07 via RPC
        objectGuid: 6a7ff635-baeb-11d2-8fda-0008c709d19e
    Washington\WORKSTATION1 via RPC
        objectGuid: ed8a3ba0-d439-11d2-99e7-08002ba3ed3b

CN=Configuration,DC=reskit,DC=microsoft,DC=com
    Washington\DSYSRESKIT0 via RPC
        objectGuid: abbf2810-f51b-11d2-84a0-00105a68cf71
    Washington\RESKIT-DC-07 via RPC
        objectGuid: 6a7ff635-baeb-11d2-8fda-0008c709d19e
    Washington\WORKSTATION1 via RPC
        objectGuid: ed8a3ba0-d439-11d2-99e7-08002ba3ed3b
```

Viewing Replication Status and Performance

Active Directory Replication Monitor (Replmon.exe) is a graphical tool that you can use to view low-level status and performance of replication between Active Directory domain controllers. Replication Monitor can be used to monitor replication features, as follows:

- Automatically detects all directory partitions on the selected server, including Global Catalog servers. It displays whether or not the monitored server is a Global Catalog server, automatically discovers the directory partitions that the monitored server hosts, graphically displays this breakdown and shows the replication partners that are used for inbound replication for each directory partition. ReplMon distinguishes between direct replication partners, transitive replication partners, bridgehead servers, and servers removed from the network in the user interface. When there is a failure from a specific replication partner, this is reflected by a change in the icon used for the partner.
- Displays servers with which the computer is replicating both directly and transitively.

  Allows administrators to display the properties for the monitored server including: the server name, the DNS host name of the computer, the location of the computer account in Active Directory, preferred bridgehead status, any special flags for the server (for example, if it is the PDC Emulator for its domain or not), which computers it believes to hold the FSMO roles, and the replication connections (Replmon differentiates between administrators and automatically generated connection objects).
- For direct replication partners, displays each USN value, number of failed attempts, reasons, and flags used for each partner. For direct replication partners, a series of property pages displays the following for each partner: the name of the domain controller, its GUID, the directory partition that it replicates to the monitored server, the transport used (RPC or SMTP and distinguishes between intra- and inter-site when RPC is used), the time of the last successful and attempted replication events, Update Sequence Number (USN) values, and any special properties of the connection between the two servers.
- Offers a server wizard so that administrators can select from a list without knowing the server name. The administrator also can create an .ini file that predefines the names of the servers to monitor, which is then loaded by ReplMon to populate the user interface.
- Automatically detects the domain of the client and displays the appropriate option.

- Displays a graphical view of the intra-site topology and by using the context menu for a specific domain controller in the view, allows the administrator to quickly display the properties of the server and any inter-site connections that exist for that server.
- Allows administrators to display how remote domain controllers are configured (for example, whether they have the PDC emulator role).
- In Automatic Update mode, polls the server at an administrator-defined interval to get current statistics and replication state. This feature generates a history of changes for each server that is monitored and its replication partners and allows the administrator to see topology changes as they occur for each monitored server.
- Allows administrators to synchronize between only two domain controllers. Allows administrators to trigger replication on a server with a specific replication partner, with all other domain controllers in the site, or all other domain controllers within a site and between sites. (Replication partners are established by the data that they hold. Not all domain controllers are involved in this process; it depends on the domain controller being synchronized.)
- Allows administrators to trigger the KCC to re-calculate the replication topology.
- Logs all of the tool's activities.
- Keeps a log file that shows the history of the replication state of the server; the log file is viewable in the Details mode of the tool. Records the history of replication status per-directory partition per-replication partner, allowing a granular history to be generated for what occurred between two domain controllers. This history can be viewed by the person who is running Replmon.
- Monitors the count of failed replication attempts. If the failure meets or exceeds an administrator-defined value, it can write to the event log and send mail. Monitors the count of failed replication attempts for each replication partner. If the failure count meets or exceeds an administrator-defined value, Replmon can write to the event log and send an e-mail notification to the administrator.
- Allows administrators to show which objects have not yet replicated from a particular server computer.

  Displays the metadata of an Active Directory object's attributes, which include the attribute name, version number, the time the attribute was last changed, which domain controller last changed the attribute, and the USNs on the monitored server and on the computer where the change was originally written.

- Displays information and instructions about using each feature when you move your mouse over it.

  Displays a graphical view of the intra-site topology and, by using the context menu for a specific domain controller in the view, allows the administrator to quickly display the properties of the server and any inter-site connections that exist for that server. Supports a mode that records the attributes that were replicated to the monitored server from its replication partners.
- Generates reports that can aid in technical support calls. Allows the administrator to generate a status report for the monitored server, which includes a listing of the directory partitions for the server, the status of each replication partners (direct and transitive) for each of the directory partitions, the status of any Group Policy objects, the domain controllers that hold the FSMO roles, a snapshot of the performance counters on the computer, and the registry configuration of the server (including parameters for the KCC, Active Directory, Jet, and LDAP). Additionally, the administrator can also choose to record (in the same report), the enterprise configuration, which includes each site, site link, site link bridge, subnet, and domain controller (regardless of domain) and the properties of each type of object that was mentioned. For example, for the domain controller properties, this records the GUID that makes up the DNS record that is used in replication, the location of the computer account in Active Directory, the Inter-Site mail address (if it exists), the host name of the computer, and any special flags for the server (whether or not it is a Global Catalog server). This can be extremely helpful when troubleshooting an Active Directory replication problem.
- Supports supplying alternate credentials to allow ReplMon to simultaneously monitor replication status of domain controllers from multiple forests.

Replmon Requirements

Active Directory Replication Monitor must be installed on a computer that is running Microsoft® Windows® 2000 Professional or Windows 2000 Server. The computer can be a domain controller, member server, member workstation, or stand-alone computer. In addition, Replmon can be used to monitor domain controllers from different forests simultaneously.

Using Ldp.exe to Find the DSA Object GUID

The server GUID is a reference point that is used in Active Directory and DNS to locate a domain controller primarily for the purposes of replication. This GUID is automatically generated for each domain controller, is unique when created, and is not duplicated.

Note There are various GUIDs that are used for different purposes. The GUIDs that are used by the Repadmin tool are called "DSA objectGuid" because they are GUIDs of the Ntds settings object. This is displayed in the first four lines of **repadmin /showreps** output.

▶ **To identify the DSA object GUID for a particular domain controller that might be useful troubleshooting replication issues**

1. Using Ldp.exe, search the configuration directory partition with the following criteria:

   ```
   Base DN: CN=Sites,CN=Configuration,DC=RootDomainName,DC=Com
   Filter : (cn=NTDS Settings)
   Scope: Subtree
   Attributes: objectGUID
   ```

   Replace RootDomainName with the name of the first domain that was installed in the enterprise (the "forest root domain").

2. In the search results, locate the entry that represents the appropriate server for which tyou are determining the GUID. The *objectGUID* attribute must also be present and look like the following:

   ```
   ***Searching...
      ldap_search_s(ld, "cn=sites,cn=configuration,dc=reskit,dc=com", 2,
   "(cn=NTDS
      Settings)", attrList, 0, &msg)
      Result <0>: (null)
      Matched DNs:
      Getting 1 entries:
      >> Dn: CN=NTDS
      Settings,CN='server-name',CN=Servers,CN='site-
   name',CN=Sites,CN=Configuration,DC=reskit,DC=com
   1> objectGUID: e99e82d5-deed-11d2-b15c-00c04f5cb503;
   ```

   The DSA objectGUID is identified by the value that is associated with the *objectGUID* attribute.

For more information about Active Directory command-line tools, see *Windows 2000 Resource Kit* Tools Help, which is included on the *Windows 2000 Resource Kit* companion CD.

Disaster Recovery

When you have experienced a domain controller failure either due to environmental hazards or equipment malfunction, you need to first repair the domain controller itself and then recover the data.

Because Active Directory is not only a transactional database, but a directory service on top of a transactional database, it is able to recover lost information because:

- The database uses log files to recover lost information.
- The directory service uses replication to recover data from other servers in the domain.

There are a variety of tools that can be used to repair the domain controller itself and recover Active Directory.

For more information about Windows 2000 disaster protection, including backups, restores, and repairs, see the *Server Operations Guide*. Also see "System Recovery and Troubleshooting" in the product documentation, and Windows 2000 Server Help.

Repairing a Domain Controller

To repair a failed Windows 2000 Server domain controller there are several options available to you. You might need to use one or all of the following methods to repair a failed domain controller:

- *Ntbackup tool Emergency Repair Disk (ERD) wizard.* You would log on by using an account that has Administrator or Backup Operator privileges. You can use the wizard to prepare a set of disaster recovery disks so that you can restart the domain controller.
- *Reinstall the Windows 2000 operating system and run the Active Directory Installation Wizard (Dcpromo.exe).* In the case of a major hardware malfunction that requires that the computer be completely rebuilt, reinstall the operating system. This ensures that the number and size of disk volumes is the same or larger than the previous computer. Reapply your Network Connections and DNS settings as originally configured.

- *Netdom tool.* In the case you needed to remove a domain, you would first run the Active Directory Installation Wizard to remove Active Directory from all domain controllers in the domain that are being removed. Then use the netdom tool to remove the domain itself (including cross reference and trusted domain objects). For example, at the command prompt, type **netdom trust /remove /force**
- *Ntdsutil tool Cleanup command.* To cleanup metadata left behind by decommissioned or failed domain controllers, use the **cleanup** command. It removes the defunct domain controller's identification and information from the directory. You might have to run the Dcpromo tool in addition and rename the new domain controller with the same name as the old domain controller. Replication brings the domain controller up to date with regard to its replication partners.

For more information about installing and removing Active Directory with the Active Directory Installation Wizard (Dcpromo tool), see "Active Directory Data Storage" in this book. For more information about the Ntdsutil tool, see "Active Directory Diagnostic Tool (Ntdsutil.exe)" in this book.

Repairing a Windows NT 4.0–based Backup Domain Controller

Recovering a lost backup domain controller account becomes important when you are running Windows NT 4.0 in a mixed mode environment. It's important to know how to recover if the computer account for a Windows NT 4.0–based backup domain controller becomes corrupt or is accidentally deleted.

Note If the computer account for a backup domain controller in a mixed-mode domain gets deleted, you can use the **dsacls** command.

▶ **To repair a backup domain account**

1. On the orphaned backup domain controller, log on locally by using an account with administrator privileges.
2. Start Server Manager.

   From the **Start** menu, click **Run**, and then type:

   svrmgr

   Server Manager for Windows NT 4.0 or for Windows NT 3.*x* is displayed.
3. Re-create the account for the backup domain controller. (This actually happens on the primary domain controller.)
4. Use the **force sync** command to reset the password properly.

Recovering from a Deleted Windows 2000 Computer and Domain Account

When you restart the computer, you might receive the following error message:

```
event id 26 application pop-up

Application popup: lsass.exe - System Error : Security Accounts Manager
initialization failed because of the following error: No mapping between
account names and security IDs was done.  Error Status: 0xc0000073.
Please click OK to shutdown this system and reboot into Directory
Services Restore Mode, check the event log for more detailed
information.
```

The problem is that if you delete a computer account, you need to wait for the delete to replicate to all domain controllers in the domain, before doing anything further with the computer, such as joining or running the Active Directory Installation Wizard. Otherwise, the join process and Active Directory Installation Wizard re-uses the existing account and then the delete replicates in causing start or logon failures.

The solution to the computer account problem differs for clients, servers, and domain controllers:

- For a client or server, it is easy to recover from this situation. Rejoin the computer.
- For a domain controller, there is System State data to be considered, such as RID, Service Principal Names (SPNs), and FRS subscription. This System State data is affected and the only means of recovery is to reinstall it as a domain controller or authoritatively restore the domain controller's computer account.

The following is a typical scenario:

- An administrator deletes a computer account for a computer.
- An administrator rejoins the server to the domain.
- The join code attempts to find a domain controller *with* the account because the domain controller wants to avoid creating duplicate accounts.
- It is guaranteed to find a domain controller that hasn't yet replicated the deletion, if the join was attempted shortly after deleting the account (for example, within a replication latency).
- It joins to that domain controller and sets the password, and so on and then reports a success.
- The delete replicates throughout the domain and then removes the account.

Recovering Active Directory

After the domain controller is repaired, it needs to be restored with Active Directory.

- *Replication.* If there is more than one domain controller in a domain, Active Directory is restored through normal replication with the replication partners.
- *Ntbackup Backup and Restore Wizards.* Use the Ntbackup tool to restore the System State from a backup copy, which recovers Active Directory, FRS (including SYSVOL), and Certificate Services (if installed). This option is appropriate if there are no other domain controllers in the domain with which the domain controller can replicate.

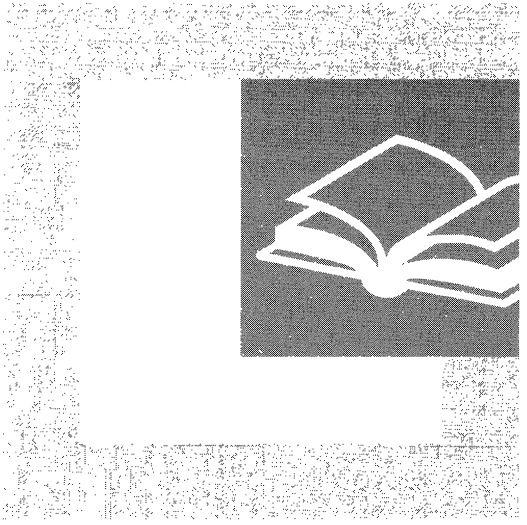
In cases where Active Directory needs to be recovered from hardware failure or replacement where data on other domain controllers is known to be stable, you need to perform only a nonauthoritative restore from the most recent backup. After the nonauthoritative restore, Active Directory replication automatically begins propagating any changes from other domain controllers that occurred after the time of the backup.

Additional Resources

- For more in-depth technical information about Active Directory, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.
- For more information about specific examples of problems mentioned in this chapter as well as current diagnostics and troubleshooting recommendations, see the Microsoft Product Support Services link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

PART 2

Distributed Security

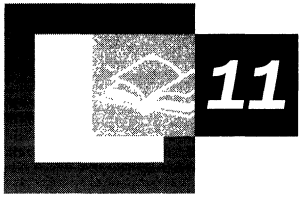


Security is an essential feature for managers and administrators of all networks, large or small. Part 2 examines features of Microsoft® Windows® 2000 that you can use to secure access to the network, access to resources, and the privacy and integrity of data and communications.

In This Part

CHAPTER 11

Authentication



The default protocol for network authentication in Microsoft® Windows® 2000 is the Kerberos v5 authentication protocol. An emerging authentication standard, the Kerberos protocol provides a foundation for interoperability. It also enhances the security of enterprise-wide network authentication. Key components of the protocol's implementation in Windows 2000 include the integration of initial authentication with the Winlogon single sign-on architecture, the use of Active Directory™ (the directory service included in Windows 2000) as the domain's security account database, and the implementation of the Kerberos client as a Windows 2000 security provider through the Security Support Provider Interface (SSPI).

In This Chapter

Related Information in the Resource Kit

- For more information about how network clients find a domain controller, see "Name Resolution in Active Directory" in this book.
- For more information about authorization, see "Access Control" in this book.

Basic Concepts of Authentication

Authentication is a process for verifying the identity of something or someone. When you authenticate an object, the goal is to verify that you have the genuine article. When you authenticate a person, the goal is to verify that you are not dealing with an imposter.

Both kinds of authentication occur when you cross an international border. A guard asks for credentials; you present a passport. The guard authenticates the passport by verifying that it was issued by a security authority the local government trusts—trusts, at least, to issue passports. The guard authenticates you by verifying that your face matches the face of the person pictured on the passport. If the passport proves to be valid and you prove to be its owner, you are allowed to proceed. Otherwise, you are denied access to the country/region you want to enter.

The kind of authentication that takes place at an international border is based on trust. The local government does not know you, but it trusts that your government does. When your government issued your passport, it did not know you either. It trusted the agency that issued your birth certificate. The agency that issued your birth certificate in turn trusted the physician who signed the certificate. The physician witnessed your birth and stamped the certificate with your footprint, direct proof of your identity. Trust transferred in this way, through trusted intermediaries, is *transitive*. Transitive trust between security authorities is the foundation for network security in Windows 2000.

Interactive Logon

Logging on at the keyboard of a computer running Windows 2000 is like crossing an international border. A guard asks for identification; you present credentials issued by a trusted authority. In this case, the guard is Winlogon, a security service running in a process it shares with the Local Security Authority (LSA). Winlogon displays a dialog box that asks you to identify your account and the security authority who issued it. It also requires you to substantiate your claim that you are the account holder, which on standard Windows 2000 systems you do by typing a password. On specially equipped systems, your credentials might be taken from a smart card that you insert into a card reader. Whatever proof of identity you supply, Winlogon collects it, packages it in a data structure, and passes everything to the LSA for verification. If the LSA can verify that your account is valid and that you are the account holder, Winlogon sets up an interactive session on the computer. Otherwise, you are denied access to the computer.

How the LSA verifies your identity depends on where your account was issued. If your account was issued by the LSA itself, the LSA can validate your information by checking its own account database. If you are using an account issued by the security authority for the local domain or by the security authority for a trusted domain, the LSA must contact the issuing authority and ask it to verify that the account is valid and that you are the account holder.

Remote Logon

No matter where your credentials are authenticated, the process establishes your identity with just one security authority—the LSA on the computer where you are logging on. The fact that you are able to use a domain account means only that the LSA on your workstation trusts the security authority for the domain. Other Windows 2000 computers in the domain share the same trust, yet before you can browse those computers you must log on to each of them as well. It does not matter that your "passport" was stamped at the border of the last computer where you logged on. You must show it again whenever you want access to another network client that is running Windows 2000.

When you want access to a system across the network that is running Windows 2000 from the computer where you have logged on, you are not required to provide information in a dialog box as you are required to do when you log on at a keyboard. Instead, the LSA on your workstation establishes your identity with the LSA on the remote computer by using the credentials that were cached during your initial interactive logon to the network.

Security Principals

In Windows 2000, any entity that can initiate action is a *security principal*. Thus, security principals can be either human users or inanimate entities such as computers or services ("daemons," if you are from the UNIX world). Security principals establish a context for their actions by presenting credentials from a security authority that is trusted by the LSA on the computer where the principal intends to act.

For example, Windows 2000 computers participate in a network domain by communicating with a domain controller, and they do this even when no human user is logged on. To initiate communications, the computer must have an account in the domain and must present credentials to prove that it is the account holder. Before accepting communications from the computer, the LSA on the domain controller authenticates the computer's identity just as it would for a human security principal.

Although most Windows 2000 applications run in the security context of the user who starts them, this is not true of services. Windows 2000 services are started by the service controller, often automatically when the computer starts. They continue to run long after the last human user has logged off. Services have to log on to domain accounts to gain access to domain resources, just as human users and Windows 2000 computers do. Before starting a service, the service controller logs on to the account designated for the service and presents the service's credentials for authentication by the LSA.

For example, when a Windows 2000 computer joins a domain, the Net Logon service on the computer connects to a domain controller and opens a secure channel to it. To obtain an authenticated connection, Net Logon must have credentials that are trusted by the remote computer's LSA. It uses credentials for the local computer's domain account, just as all other services running as Local System do. Any service not running as Local System must have its own domain account.

Authentication Protocols

Windows 2000 supports several protocols for verifying the identities of users who claim to have accounts on the system. These include protocols for authenticating dial-up connections and protocols for authenticating external users who are trying to connect to the network over the Internet. However, there are only two choices for network authentication within and between Windows 2000 domains:

Kerberos v5 Protocol The Kerberos v5 authentication protocol is the default for authentication of users who are logging on to domain accounts from computers that are running Windows 2000.

NTLM Protocol The Windows NTLM protocol was the default for authentication in Microsoft® Windows NT® version 4.0. It is retained in Windows 2000 for compatibility with clients and servers that are running Windows NT version 4.0 and earlier. It is also used to authenticate logons to stand-alone computers that are running Windows 2000.

The Kerberos protocol is the protocol of choice in Windows 2000, when there is a choice. Computers with Microsoft® Windows 3.11, Microsoft® Windows® 95, Microsoft® Windows® 98, or Windows NT 4.0 must use the NTLM protocol for network authentication in Windows 2000 domains. Computers with Windows 2000 use NTLM when they are authenticating to servers that are running Windows NT 4.0 and when they are requesting access to resources in Windows NT 4.0 domains.

How Kerberos Authentication Works

The Kerberos authentication protocol provides a mechanism for mutual authentication between a client and a server before a network connection is opened between them. The protocol assumes that initial transactions between clients and servers take place on an open network—an environment where most clients and many servers are not physically secure and packets traveling along the network can be monitored and modified at will. In other words, the protocol is designed for an environment that is much like today's Internet, where an attacker can easily pose as either a client or a server and can readily eavesdrop on or tamper with communications between legitimate clients and servers.

Basic Concepts for the Kerberos Protocol

The Kerberos protocol relies heavily on an authentication technique that makes use of shared secrets. The basic concept is quite simple: If a secret is known by only two people, either person can verify the identity of the other by confirming that the other person knows the secret.

For example, let's suppose that Alice often sends messages to Bob and that Bob needs to be sure that a message from Alice really has come from Alice before he acts on its information. They decide to solve their problem by selecting a password, and they agree not to share this secret with anyone else. If Alice's messages can somehow demonstrate that the sender knows the password, Bob knows that the sender is Alice.

The only question for Alice and Bob to resolve is *how* Alice can show that she knows the password. She might simply include it somewhere in her messages, perhaps in a signature block at the end—*Alice, Our$ecret*. This would be simple and efficient and might even work if Alice and Bob can be sure that no one else is reading their mail. Unfortunately, that is not the case. Their messages pass over a network used by people like Carol, who has a network analyzer and a hobby of scanning traffic in hope that one day she might spot a password. So it is out of the question for Alice to prove that she knows the secret simply by saying it. To keep the password secret, she must show that she knows it without revealing it.

The Kerberos protocol solves this problem with *secret key cryptography*. Rather than sharing a password, communication partners share a cryptographic key. They use knowledge of this key to verify one another's identity. For this method of authentication to work, the shared key must be symmetric—a single key must be capable of both encryption and decryption. One party proves knowledge of the key by encrypting a piece of information, the other by decrypting it.

Authenticators

A simple protocol that uses secret key authentication begins when someone is outside a communications door and wants to go in. To gain entry, this person presents an *authenticator* in the form of a piece of information encrypted in the secret key. The information in the authenticator must be different each time the protocol is executed, otherwise an old authenticator could be reused by anyone who happens to overhear the communication. Upon receiving an authenticator, the person guarding the door decrypts it and knows from what is inside it whether the decryption was successful. If it was successful, the doorkeeper knows that the person presenting the authenticator has the correct key. Only two people have the key; the doorkeeper is one of them, so the person who presented the authenticator must be the other one.

If the person outside the door wants mutual authentication, the same protocol can be executed in reverse, with a slight difference. The doorkeeper can extract part of the information from the original authenticator, encrypt it in a new authenticator, and then give the new authenticator to the person waiting outside the door. The person outside the door can then decrypt the doorkeeper's authenticator and compare the result with the original. If there is a match, the person outside the door knows that the doorkeeper was able to decrypt the original, so he must have the correct key.

It might help to walk through an example. Suppose Alice and Bob decide that before transferring any information between their computers, each will use knowledge of a shared secret key to verify the identity of the party at the other end of the connection. In situations where Alice is the wary guest and Bob is the suspicious host, they agree to follow this protocol:

1. Alice sends Bob a message containing her name in plaintext and an authenticator encrypted in the secret key she shares with Bob. In this protocol, the authenticator is a data structure with two fields. One field contains information about Alice. For simplicity, let's say this is her name. The second field contains the current time on Alice's workstation.
2. Bob receives the message, sees that it is from someone claiming to be Alice, and uses the key he shares with Alice to decrypt the authenticator. He extracts the field that contains the time on Alice's workstation and evaluates the time.

   Bob's task is easier if his clock is reasonably synchronized with Alice's clock, so let's suppose both Alice and Bob use a network time service to keep their clock times fairly close. Let's say the time skew is never more than five minutes. This way, Bob can compare the time from the authenticator with the current time on his clock. If the difference is greater than five minutes, he can automatically reject the authenticator.

If the time is within the allowable skew, it's probable that the authenticator came from Alice, but Bob still does not have proof that it actually came from her. Another person might have been watching network traffic and might now be replaying an earlier attempt by Alice to establish a connection with Bob. However, if Bob has recorded the times of the authenticators that were received from Alice during the past five minutes, he can defeat attempts to replay earlier messages by rejecting any message with a time that is the same as or earlier than the time of the last authenticator. If this authenticator yields a time later than the time of the last authenticator from Alice, then this message must be from Alice.

3. Bob uses the key he shares with Alice to encrypt the time shown on Alice's message and sends the result back to her.

   Note that Bob does not send back all of the information taken from Alice's authenticator, just the time. If he sent back everything, Alice would have no way of knowing whether someone posing as Bob had simply copied the authenticator from her original message and sent it back to her unchanged. He sends just a piece of the information in order to demonstrate that he was able to decrypt the authenticator and manipulate the information inside. He chooses the time because that is the one piece of information that is sure to be unique in Alice's message to him.

4. Alice receives Bob's reply, decrypts it, and compares the result with the time in her original authenticator. If the times match, she can be confident that her authenticator reached someone who knows the secret key needed to decrypt it and extract the time. She shares that key only with Bob, so it must be Bob who received her message and replied.

This process is illustrated in Figure 11.1.

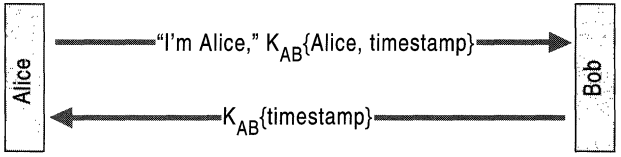


Figure 11.1 A Simple Protocol for Mutual Authentication

Key Distribution

One problem with the simple protocol described in the preceding section is that it does not explain how or where Alice and Bob get a secret key to use in their communications with each other. If they are people, Alice and Bob can meet, perhaps in an alley, and agree on a secret key. But if Alice is a client program that is running on a workstation and Bob is a service that is running on a computer somewhere across the network, that method does not work. There is the further problem that the client, Alice, might want to talk to many services and will need keys for each of them. Likewise, the service, Bob, might talk to many clients and will need keys for each of them. If each client needs a key for every service and each service needs a key for every client, key distribution can quickly become a difficult problem to solve. The need to store and protect so many keys on so many computers presents an enormous security risk.

The name Kerberos suggests how the protocol resolves the problem of key distribution. Kerberos (also known as Cerberus) was a figure in classical Greek mythology, a three-headed dog who kept living intruders from entering the underworld. Like the mythical guard dog, the protocol has three heads, which in this case are a client, a server, and a trusted third party that mediates between the client and server. The trusted intermediary in the protocol is known as the Key Distribution Center (KDC).

The KDC is a service that runs on a physically secure server. It maintains a database with account information for all security principals in its *realm*—the protocol's equivalent of a Windows 2000 domain. Along with other information about each security principal, the KDC stores a cryptographic key known only to the security principal and the KDC. This key is used in exchanges between the security principal and the KDC and is known as a *long-term key*. In most implementations of the protocol, the long-term key is derived from a user's logon password.

When a client wants to talk to a server, the client sends a request to the KDC, and the KDC distributes a unique *session key* for the two parties to use when they authenticate each other, as illustrated in Figure 11.2. The server's copy of the session key is encrypted in the server's long-term key. The client's copy of the session key is encrypted in the client's long-term key.

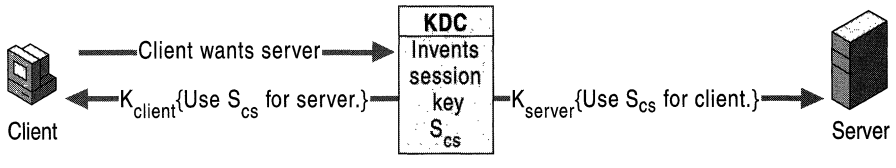


Figure 11.2 Key Distribution (in Theory)

In theory, the KDC can fulfill its role as a trusted intermediary by sending the session key directly to each of the security principals involved, as illustrated in Figure 11.2. But, in practice, that procedure would be extremely difficult to implement. For one thing, it would mean that the server would have to retain its copy of the session key in memory while it waited for the client to call. Moreover, the server would need to remember a key not just for this client but for every client who might ask for service. Key management would consume considerable resources on the server and would thus limit its scalability. In addition, given the vagaries of network traffic, a client's request for service might reach the server before the KDC's message arrived there with the session key. The server would have to suspend its reply to the client while it waited to hear from the KDC. This would require the server to save state, imposing still another burden on the server's resources. What actually happens in the Kerberos protocol is considerably more efficient.

Session Tickets

The KDC responds to the client's request to talk to a server by sending both copies of the session key to the client, as shown in Figure 11.3. The client's copy of the session key is encrypted with the key that the KDC shares with the client. The server's copy of the session key is embedded, along with authorization data for the client, in a data structure called a *session ticket*. The entire structure is then encrypted with the key that the KDC shares with the server. The session ticket—with the server's copy of the session key safely inside—becomes the client's responsibility to manage until it contacts the server.

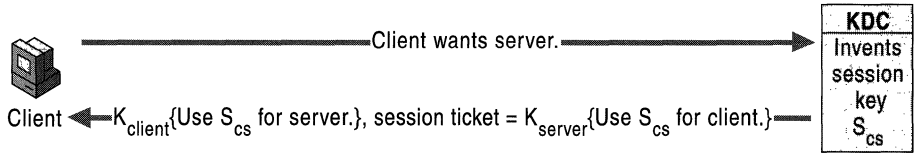


Figure 11.3 Key Distribution (in Practice)

Note that the KDC is simply providing a ticket-granting service. It does not keep track of its messages to make sure they reach the intended address. No harm is done if the KDC's messages fall into the wrong hands. Only someone who knows the client's secret key can decrypt the client's copy of the session key. Only someone who knows the server's secret key can read what is inside the ticket.

When the client receives the KDC's reply, it extracts the ticket and the client's copy of the session key, putting both aside in a secure cache, which is located in volatile memory, not on disk. When the client wants admission to the server, it sends the server a message that consists of the session ticket, which is still encrypted with the server's secret key, and an authenticator, which is encrypted with the session key, as illustrated in Figure 11.4. The session ticket and authenticator together are the client's credentials to the server.

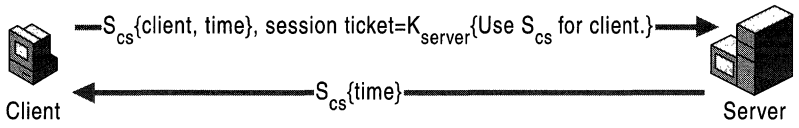


Figure 11.4 Mutual Authentication (Client/Server)

When the server receives credentials from a client, it decrypts the session ticket with its secret key, extracts the session key, and uses the session key to decrypt the client's authenticator. If everything checks out, the server knows that the client's credentials were issued by a trusted authority, the KDC. If the client has asked for mutual authentication, the server responds by using the session key to encrypt the timestamp from the client's authenticator. The server then returns the encrypted timestamp to the client, just as Bob returned the encrypted timestamp to Alice in the communication illustrated in Figure 11.1.

One benefit of using session tickets is that the server does not have to store the session key that it uses with this client. It is the client's responsibility to hold a session ticket for the server in its credentials cache and present the ticket each time it wants access to the server. Whenever the server receives a session ticket from a client, it can use its secret key to decrypt the ticket and extract the session key. When the server no longer needs the session key, it can discard it.

Another benefit of using session tickets is that the client does not have to go back to the KDC each time it wants access to a particular server. Session tickets can be reused. As a precaution against the possibility that someone might steal a copy of a ticket, session tickets have an expiration time that is specified by the KDC in the ticket's data structure. How long a session ticket is valid depends on the Kerberos policy for the domain. Tickets usually are good for no longer than eight hours, about the length of a normal logon session. When the user logs off, the credentials cache is flushed and all session tickets—as well as all session keys—are destroyed.

Ticket-Granting Tickets

A user's long-term key is derived from a password. When Alice logs on, for example, the Kerberos client on her workstation accepts her password and then converts it to a cryptographic key by passing the text of the password through a one-way hashing function.

The KDC gets its copy of Alice's long-term key from her record in its account database. When it receives a request from the Kerberos client on Alice's workstation, the KDC searches its database for Alice, pulls up her account record, and takes her long-term key from a field in the record.

This process of computing one copy of the key from a password and fetching another copy of the key from a database actually takes place only once, when a user initially logs on to the network. Immediately after accepting the user's password and deriving the user's long-term key, the Kerberos client on the user's workstation asks the KDC for a session ticket and session key that it can use in subsequent transactions with the KDC during this logon session.

The KDC responds to the client's request by returning a session ticket for itself. This special session ticket is called a *ticket-granting ticket (TGT)*. Like an ordinary session ticket, a TGT contains a copy of the session key that the service (in this case the KDC) uses in communicating with the client. The message that returns the TGT to the client also includes a copy of the session key that the client can use in communicating with the KDC. The TGT is encrypted in the KDC's long-term key. The client's copy of the session key is encrypted in the user's long-term key.

When the client receives the KDC's reply to its initial request, it uses its cached copy of the user's long-term key to decrypt its copy of the session key. It can then discard the long-term key derived from the user's password, for it is no longer needed. In all subsequent exchanges with the KDC, the client uses the session key. Like any other session key, this key is temporary, valid only until the TGT expires or the user logs off. For that reason, it is called a *logon session key*.

From the client's point of view, a TGT is just another ticket. Before it attempts to connect to a service, the client first checks its credentials cache for a session ticket to that service. If it does not have one, it checks the cache again for a TGT. If it finds a TGT, the client fetches the corresponding logon session key from the cache, uses this key to prepare an authenticator, and sends both the authenticator and the TGT to the KDC, along with a request for a session ticket for the service. In other words, gaining admission to the KDC is no different from gaining admission to any other service in the domain—it requires a session key, an authenticator, and a ticket (in this case, a TGT).

From the KDC's point of view, TGTs allow it to shave a few nanoseconds off the turnaround time for ticket requests. The KDC looks up the user's long-term key only once, when it grants an initial TGT. For all other exchanges with this client, the KDC can decrypt the TGT with its own long-term key, extract the logon session key, and use that to validate the client's authenticator.

Authentication Across Domain Boundaries

The functions of the KDC are divided into two distinct services: an authentication service, whose job is to issue TGTs, and a ticket-granting service, whose job is to issue session tickets. This division of labor allows the Kerberos protocol to operate across domain boundaries. A client can get a TGT from the authentication service of one domain and use it to get session tickets from the ticket-granting service of another domain.

The easiest way to grasp how cross-domain authentication works is to consider the simplest case—a network that has only two domains. For our example, let's say one domain is East and the other domain is West. If the administrators for these domains are members of the same organization (or if for some other reason they are willing to treat each other's domain users as their own), they can enable authentication across domain boundaries by sharing interdomain keys. (In Windows 2000 this happens automatically when two domains establish a trust relationship.) After this is accomplished, the ticket-granting service in each domain is registered as a security principal with the other domain's KDC. This means the ticket-granting service in each domain can treat the ticket-granting service in the other domain as just another service, something for which properly authenticated clients can request and receive session tickets.

When a user with an account in East wants access to a server with an account in West, the Kerberos client on the user's workstation sends a request for a session ticket to the ticket-granting service in the user's account domain, East. The ticket-granting service in East sees that the desired server is not a security principal in its own domain, so it replies by sending the client a *referral ticket.* This is simply a TGT that is encrypted with the interdomain key shared by the KDC in East and the KDC in West. The client uses the referral ticket to prepare a second request for a session ticket, and this time sends the request to the ticket-granting service in the server's account domain, West. The ticket-granting service in West uses its copy of the interdomain key to decrypt the referral ticket. If decryption is successful, it sends the client a session ticket to the desired server in its domain.

The referral process is more complicated on networks with more than two domains. In theory, the KDC in each domain could establish a direct link to the KDC in every other domain on the network, in each case sharing a different inter-domain key. In practice, the number and complexity of these relationships can easily become unmanageable, especially on a large network. The Kerberos protocol solves the problem by making direct links unnecessary. A client in one domain can get a ticket to a server in another domain by traveling a referral path through one or more intermediate domains.

For example, consider a network with three domains, East, West, and CorpHQ. The KDC in East does not share an inter-domain key with the KDC in West, but both East and West do share inter-domain keys with CorpHQ. In this case, when a user with an account in East wants access to a server with an account in West, the referral path begins at the KDC for the user's account domain, East, passes through an intermediate domain, CorpHQ, and ends at the KDC for the server's account domain, West. The client must send its request for a session ticket three times, to three different KDCs.

1. The client asks the KDC for East to give it a ticket to the server in West.

   The KDC for East sends the client a referral ticket to the KDC for CorpHQ. This ticket is encrypted in the interdomain key East shares with CorpHQ.

2. The client asks the KDC for CorpHQ to give it a ticket to the server in West.

   The KDC for CorpHQ sends the client a referral ticket to the KDC for West. This ticket is encrypted in the interdomain key CorpHQ shares with West.

3. The client asks the KDC for West to give it a ticket to the server in West.

   The KDC in West replies by sending a ticket for the server in West.

Subprotocols

The Kerberos protocol is comprised of three subprotocols. The subprotocol in which the KDC gives the client a logon session key and a TGT is known as the Authentication Service (AS) Exchange. The subprotocol in which the KDC distributes a service session key and a session ticket for the service is known as the Ticket-Granting Service (TGS) Exchange. The subprotocol in which the client presents the session ticket for admission to a service is called the Client/Server (CS) Exchange.

To see how the three subprotocols work together, let's look at how Alice, a user at a workstation, gets access to Bob, a service on the network.

AS Exchange

Alice begins by logging on to the network. She types her logon name and her password. The Kerberos client running on Alice's workstation converts her password to an encryption key and saves the result in its credentials cache.

The client then sends the KDC's authentication service a Kerberos Authentication Service Request (KRB_AS_REQ). The first part of this message identifies the user, Alice, and the service for which she is requesting credentials, the ticket-granting service, as illustrated in Figure 11.5. The second part of the message contains *preauthentication data* that proves Alice knows the password. This data is usually a timestamp encrypted with Alice's long-term key, although the protocol permits other forms of preauthentication data.

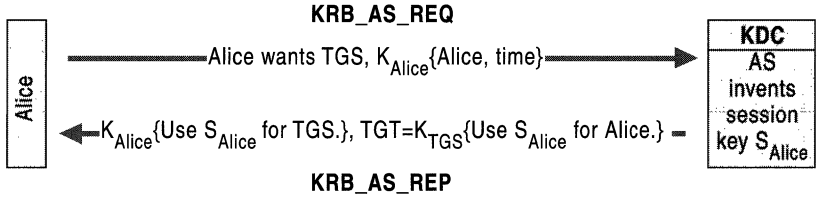


Figure 11.5 AS Exchange

When the KDC receives KRB_AS_REQ, it looks up the user, Alice, in its database, gets her long-term key, decrypts the preauthentication data, and evaluates the timestamp inside. If the timestamp passes the test, the KDC can be assured that the preauthentication data was encrypted with Alice's long-term key and thus that the client is genuine.

After it has verified Alice's identity, the KDC creates credentials that the Kerberos client on her workstation can present to the ticket-granting service. First, the KDC invents a logon session key and encrypts a copy of it with Alice's long-term key. Second, it embeds another copy of the logon session key in a TGT, along with other information about Alice, such as her authorization data. The KDC encrypts the TGT with its own long-term key. Finally, it sends these credentials back to the client in a Kerberos Authentication Service Reply (KRB_AS_REP).

When the client receives the reply, it uses the key derived from Alice's password to decrypt her logon session key and stores the key in its credentials cache. Then the client extracts the TGT from the message and stores that in its credentials cache as well.

TGS Exchange

The Kerberos client on Alice's workstation requests credentials for the service, Bob, by sending a Kerberos Ticket-Granting Service Request (KRB_TGS_REQ), as illustrated in Figure 11.6. This message includes the user's name, an authenticator encrypted with the user's logon session key, the TGT obtained in the AS Exchange, and the name of the service for which the user wants a ticket.

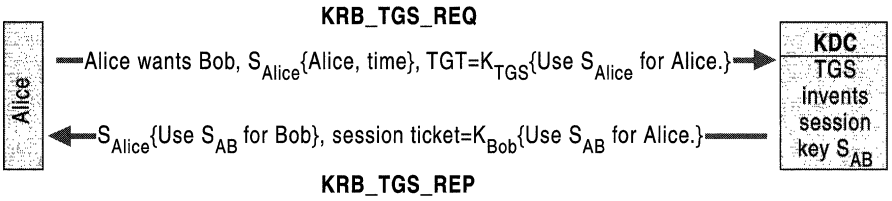


Figure 11.6 TGS Exchange

When the KDC receives KRB_TGS_REQ, it decrypts the TGT with its secret key and extracts Alice's logon session key. The KDC then uses the logon session key to decrypt the authenticator and evaluates it. If the authenticator passes the test, the KDC extracts Alice's authorization data from the TGT and invents a session key for the client, Alice, to share with the service, Bob. The KDC encrypts one copy of this session key with Alice's logon session key. It embeds another copy of the session key in a ticket, along with Alice's authorization data, and encrypts the ticket with Bob's long-term key. The KDC then sends these credentials back to the client in a Kerberos Ticket-Granting Service Reply (KRB_TGS_REP).

When the client receives the reply, it uses Alice's logon session key to decrypt the session key to use with the service, and stores the key in its credentials cache. Then it extracts the ticket to the service and stores that in its cache.

CS Exchange

The Kerberos client on Alice's workstation requests service from Bob by sending Bob a Kerberos Application Request (KRB_AP_REQ), as illustrated in Figure 11.7. This message contains an authenticator encrypted with the session key for the service, the ticket obtained in the TGS Exchange, and a flag indicating whether the client wants mutual authentication. (The Kerberos protocol can provide mutual authentication but does not require it. In Windows 2000, clients always ask for mutual authentication. Clients using other implementations of the protocol might not.)

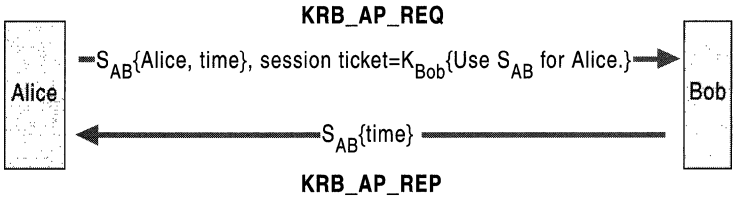


Figure 11.7 CS Exchange

The service, Bob, receives KRB_AP_REQ, decrypts the ticket, and extracts Alice's authorization data and the session key. Bob uses the session key to decrypt Alice's authenticator and then evaluates the timestamp inside. If the authenticator passes the test, Bob looks for a mutual authentication flag in the client's request. If the flag is set, he uses the session key to encrypt the time from Alice's authenticator and returns the result in a Kerberos Application Reply (KRB_AP_REP).

When the client on Alice's workstation receives KRB_AP_REP, it decrypts Bob's authenticator with the session key it shares with Bob and compares the time returned by the service with the time in the client's original authenticator. If the times match, the client knows that the service is genuine, and the connection proceeds. During the connection, the session key can be used to encrypt application data, or the client and server can share another key for this purpose.

Tickets

What is in a ticket, how ticket expiration times are calculated, and how much of a ticket's content is known by the client are important things to know when you want to configure Kerberos policy; therefore, they deserve a closer look.

What Is in a Ticket

Kerberos messages and tickets have a precise data structure and format. For more information about ticket fields and flags, see the Request for Comments (RFC) link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. Follow the links to RFC 1510.

How the KDC Limits a Ticket's Lifetime

Kerberos tickets have a start time and an expiration time. At any time after the start time but before the expiration time, a client holding a session ticket for a particular service can present the ticket and gain access to the service, no matter how many times the client has used the ticket previously. To reduce the risk of a ticket or its corresponding session key being compromised, administrators can set a maximum lifetime for tickets. This value is one element of Kerberos policy an administrator can set for the domain.

When a client asks the KDC for a session ticket to a service, it can request a specific start time. If this time is missing from a request or it is a time already past, the KDC sets the ticket's *starttime* field to the current time.

Whether or not clients specify a start time, their requests must include an expiration time. The KDC determines the value of a ticket's *endtime* field by adding the maximum ticket life fixed by Kerberos policy to the value of the ticket's *starttime* field. It then compares the result with the requested expiration time. Whichever of these times is earliest becomes the ticket's *endtime*.

Renewable Tickets

Encryption keys wear out with frequent use. The more often an encryption key is used, the more examples of the key an attacker has available to study and the greater the temptation for this person to expend the effort necessary to break the code. The keys used most often in the Kerberos protocol are the session keys. Clients use the same key each time they connect to a server, for as long as the session ticket remains valid. When the ticket expires, the client gets a new ticket with a fresh session key, but until then, the key is not changed.

One way to limit the exposure of session keys to potential attackers is to force the session keys to change often. You can do this by setting Kerberos policy so that the maximum session ticket life is relatively short. Another strategy is to permit tickets to be renewed, which allows session keys to be refreshed periodically without having to issue a completely new ticket. If Kerberos policy permits renewable tickets, the KDC sets a RENEWABLE flag in every session ticket it issues and sets two expiration times in the ticket. One expiration time limits the life of the current instance of the ticket. The second expiration time sets a limit on the cumulative lifetime of all instances of the ticket.

The expiration time for the current instance of a session ticket is held in the *endtime* field. Just as for nonrenewable session tickets, *endtime* is the value of the *starttime* field plus the maximum ticket life specified by the Kerberos policy. A client holding a renewable ticket must present it to the KDC for renewal before the *endtime* is reached and present a fresh authenticator as well. When the KDC receives a session ticket for renewal, it checks a second expiration time held in the *renew-till* field. This time is set when the ticket is first issued; the value is the session ticket's *starttime* plus the maximum cumulative ticket life specified by the Kerberos policy. When the KDC renews the ticket, it checks to see that the *renew-till* time has not yet arrived. If it has not arrived, the KDC issues a new instance of the session ticket with a later *endtime* and a new session key.

This means that administrators can set Kerberos policy can so that session tickets must be renewed at relatively short intervals—every day, perhaps. When tickets are renewed, a new session key is issued, which minimizes the value of a compromised key. At the same time, administrators can also set the cumulative session ticket life for a relatively long period—one month, one year, or whatever. At the end of that time, the session ticket expires and is no longer valid for renewal.

What Happens When Tickets Expire

When a session ticket expires, ongoing operations are not interrupted. Session tickets are used only to authenticate new connections with servers. After a connection has been set up, it no longer matters whether the session ticket is still valid. However, when a TGT expires, the Kerberos client might have to interrupt the activities of the user to ask for a password.

The KDC does not notify clients when their session tickets or TGTs are about to expire. In fact, it does not keep track of transactions with clients beyond the short-term records required to prevent replay attacks. If a client presents an outdated TGT to the KDC's ticket-granting service, the service responds with an error message. Network servers also return error messages when they receive expired tickets. All responsibility for renewing or replacing tickets rests with the client.

What Clients Know About Tickets

Clients have to know some of the information that is inside session tickets and TGTs in order to manage their credentials cache. When the KDC returns a ticket and session key as the result of an AS Exchange or TGS Exchange, it packages the client's copy of the session key in a data structure that includes the information in the ticket's *flags, authtime, starttime, endtime,* and *renew-till* fields. The entire structure is encrypted in the client's key and returned in a KRB_AS_REP or KRB_TGS_REP reply message.

Delegation of Authentication

Multi-tier client/server applications present a special situation for the Kerberos protocol. In this kind of application, a client might connect to a server that must connect to a second server on the back end. For this to happen, the first server must have a session ticket to the second server. Ideally, this session ticket should limit the first server's access on the second server to what the client is authorized to do, rather than what the server is authorized to do.

The Kerberos protocol deals with this situation through a mechanism known as *delegation of authentication.* Essentially, the client delegates authentication to a server by telling the KDC that the server is authorized to represent the client.

Delegation can be done in two ways. One way is for the client to get a session ticket for the back-end server and give it to the front-end server. Tickets obtained in this way—by a client for a proxy—are called *proxy tickets.* The difficulty with using proxy tickets is that the client must know the name of the back-end server. This difficulty is overcome by the second method of delegation, which allows the client to give the front-end server a TGT that the front-end server can use to request session tickets for the back-end server as needed. Session tickets obtained in this way—with credentials forwarded by a client—are called *forwarded tickets.* Whether the KDC allows clients to obtain proxy tickets or forwardable TGTs is determined by the Kerberos policy set by an administrator for the domain.

Proxy Tickets

When the KDC issues a TGT to a client, it checks the Kerberos policy for the domain to see if proxy tickets are allowed. If they are allowed, the KDC sets the PROXIABLE flag in the TGT that the KDC issues to the client.

The client obtains a proxy ticket by presenting a TGT to the ticket-granting service and asking for a session ticket to the back-end server. The client's request includes a flag that signals that it wants a proxy ticket; the request also includes the name of the server that will represent the client. When the KDC receives the client's request, it creates a session ticket for the back-end server, sets the PROXY flag in the session ticket, and sends the ticket back to the client. The client then sends the session ticket to the front-end server, which uses it for access to the back-end server.

Forwarded Tickets

If a client wants to delegate the task of obtaining session tickets for a back-end server to a front-end server, it must ask the KDC for a forwardable TGT. It does this through an AS Exchange by indicating to the KDC the name of the server that will act on the client's behalf. If Kerberos policy permits forwarding, the KDC creates a TGT for the front-end server to use in the client's name, sets the FORWARDABLE flag in the TGT, and sends the ticket back to the client. The client then forwards the TGT to the front-end server.

When the front-end server requests a session ticket to the back-end server, it presents the client's TGT to the KDC. When the KDC issues a session ticket for the back-end server, it sees the FORWARDABLE flag in the TGT, sets the FORWARDED flag in the session ticket for the requested server, and returns the session ticket to the front-end server. The option of setting the FORWARDED flag in a session ticket provides an added level of security because it permits administrators to configure individual servers to reject session tickets that have been forwarded.

Kerberos Components in Windows 2000

Windows 2000 implements the KDC as a domain service. It uses Active Directory as its account database and gets additional information about security principals from the Global Catalog.

As in other implementations of the Kerberos protocol, the KDC is a single process that provides two services.

Authentication Service The authentication service issues TGTs that are good for admission to the ticket-granting service in its domain. Before network clients can get tickets for services, they must obtain a TGT from the authentication service in the user's account domain.

Ticket-Granting Service The ticket-granting service issues tickets that are good for admission to other services in its own domain or for admission to the ticket-granting service of a trusted domain. When clients want access to a service, they must contact the ticket-granting service in the service's account domain, present a TGT, and ask for a session ticket. If the client does not have a TGT for admission to the ticket-granting service in the other domain, it must obtain one through the referral process that begins at the ticket-granting service in the user's account domain and ends at the ticket-granting service in the service's account domain.

The KDC is located on every domain controller, as is the Active Directory service. Both services are started automatically by the domain controller's Local Security Authority (LSA) and run in the process space of the LSA. Neither service can be stopped. Windows 2000 ensures availability of these services by allowing each domain to have several domain controllers, all peers. Any domain controller can accept authentication requests and ticket-granting requests addressed to the domain's KDC.

The security principal name used by the KDC in all Windows 2000 domains is *krbtgt*, as specified by RFC 1510. An account for this security principal is created automatically when a new Windows 2000 domain is created. The account cannot be deleted, nor can the account name be changed. A password is assigned to the KDC's account automatically; this password, like the passwords assigned to domain trust accounts, is changed on a regular schedule. The password for the KDC's account is used to derive a secret key for encrypting and decrypting the TGTs that the KDC issues. The password for a domain trust account is used to derive a Kerberos inter-realm key for encrypting and decrypting referral tickets.

All instances of the KDC in a domain use the domain account for the security principal krbtgt. Clients address messages to a domain's KDC by including both the service's principal name (krbtgt) and the name of the domain. Both items of information are also used in tickets to identify the issuing authority.

Account Database

The account database that provides information about security principals to the KDC is the domain's Active Directory. Each security principal is represented by an account object in Active Directory. The cryptographic key used in communications with a user, a computer, or a service is stored as an attribute of that security principal's account object.

Only domain controllers are Active Directory servers. Each domain controller keeps a writable copy of the directory so that accounts can be created, passwords reset, and group memberships modified at any domain controller. Changes made to one replica of the directory are automatically propagated to all other replicas. Windows 2000 does not, however, implement the Kerberos replication protocol. Instead, it replicates the information store for Active Directory by using a proprietary multimaster replication protocol over a secure channel between replication partners.

Physical storage of account data is managed by the directory system agent (DSA), a protected process that is integrated with the LSA on the domain controller. Clients of the directory service are never given direct access to the data store. Any client that wants access to directory information must use one of the supported Active Directory Service Interfaces (ADSI) to connect to the DSA and then search for, read, and write directory objects and their attributes.

Requests for access to an object or attribute in the directory are subject to validation by Windows 2000 access control mechanisms. Like file and folder objects in the NTFS file system, objects in Active Directory are protected by access control lists (ACLs) that specify who can have access to the object and in what way. Unlike the ACLs on files and folders, however, ACLs on Active Directory objects can allow or deny access not only to the entire object but also to individual attributes of the object. Thus, attributes for sensitive account information can be protected by permissions that are more restrictive than permissions granted for other attributes of an account object.

The most sensitive information about an account is, of course, its password. Although an account object's password attribute stores an encryption key that is derived from the password, not the password itself, this key is just as useful as the password to a would-be intruder. Therefore, access to an account object's password attribute is granted only to the account holder, never to anyone else, not even to administrators. Only processes with Trusted Computer Base privilege—processes running in the security context of the LSA—are allowed to read or change password information.

To hinder an offline attack by someone with access to a domain controller's backup tape, an account object's password attribute is further protected by using a *system key* to encrypt its value. This key can be stored on removable media so that it can be safeguarded separately, or it can be stored on the domain controller and protected by a dispersal mechanism. Administrators can choose where the system key is stored and which of several algorithms is used to encrypt password attributes. For more information about using system key protection, see "Encrypting File System" in this book.

Kerberos Policy

Kerberos policy is defined at the domain level and is implemented by the domain's KDC. Kerberos policy is stored in Active Directory as a subset of the attributes of domain security policy. By default, policy options can be set only by members of the Domain Admins group. The policy includes these options:

Enforce user logon restrictions When this option is enabled, the KDC validates every request for a session ticket by examining user rights policy on the destination computer to verify that the user has the right to either **Log on locally** or **Access this computer from network**. Verification is optional because the extra step takes time and might slow network access to services. The default is enabled.

Maximum lifetime for service ticket. A *service ticket* is a session ticket. Settings are in minutes. The setting must be greater than ten minutes and less than the setting for **Maximum user ticket lifetime**. The default is ten hours.

Maximum lifetime for user ticket A *user ticket* is a TGT. Settings are in hours. The default setting is ten hours.

Maximum lifetime for user ticket renewal Settings are in days. The default setting is seven days.

Maximum tolerance for computer clock synchronization Settings are in minutes. The default is five minutes.

Delegation of Authentication

In Windows NT, a service could impersonate its clients only to gain access to resources on the computer where the service process was running. In Windows 2000, a service can impersonate its clients not only when it gains access to resources on the service's computer but also when it gains access to resources on other computers. This is because the Kerberos authentication protocol supports delegation of authentication.

Delegation works only under the following conditions:

- The computers that are hosting the client process, the service process, and processes for any back-end services must all be running Windows 2000 in a Windows 2000 domain.
- The client's user account must be enabled for delegation.
- The service's account must be enabled for delegation.

To configure a user account for delegation, right-click the User object in Active Directory Users and Computers. Then click **Properties** and then the **Account** tab. In the **Account** options list, look for the option **Account is sensitive and cannot be delegated**; make sure this option is not checked.

How you configure the service account depends on whether the service runs under a computer's Local System account or under its own domain user account. If the service is configured to run under the Local System account, the computer where the service runs must be trusted for delegation. To configure a computer account as trusted for delegation, right-click the Computer object in Active Directory Users and Computers, click **Properties** and then the **General** tab. Select the **Trust computer for delegation** check box.

Caution When you trust a computer for delegation, you enable delegation for all services that run under the Local System account on the computer. If an unwary administrator installs an untrusted service on the computer and configures it to run as Local System, it too is going to be able to gain access to network resources while impersonating other users. A better practice is to configure services that use delegation to run under their own domain user accounts managed by domain administrators.

If the service is configured to run under its own domain user account, the user account of the service must be enabled to act as a delegate. To configure the user account of a service, right-click the User object, click **Properties**, and then click the **Account** tab. In the **Account** options list, check the option **Account is trusted for delegation**.

Preauthentication

By default the KDC requires all accounts to use preauthentication. This makes offline password-guessing attacks very difficult. However, preauthentication can be disabled for individual accounts when this is necessary for compatibility with other implementations of the protocol. To disable preauthentication, right-click the User object in Active Directory Users and Computers. Click **Properties**, and then click the **Account** tab. In the **Account** options list, check **Do not require Kerberos preauthentication**.

Kerberos Security Support Provider

The Kerberos authentication protocol is implemented as a *security support provider (SSP)*—a dynamic-link library (DLL) that is supplied with the operating system. Windows 2000 also includes an SSP for NTLM authentication. By default, both the Kerberos protocol and the NTLM protocol are loaded by the LSA on a computer that is running Windows 2000 when the system starts. In Windows 2000 domains, either of these SSPs can be used to authenticate network logons and client/server connections. Which SSP is used depends on the capabilities of the computer on the other side of the connection. The Kerberos SSP is always the first choice.

System services and transport-level applications have access to SSPs through the Microsoft Security Support Provider Interface (SSPI). The SSPI is a Microsoft® Win32® interface with methods for enumerating the SSPs available on a computer, selecting one, and then using it to obtain an authenticated connection. The methods provided in the SSPI are generic, black-box routines that developers can use without knowing the details of a particular protocol. For example, when a client/server connection is authenticated, the application on the client's side of the connection sends credentials to the server using the SSPI method InitializeSecurityContext. If the Kerberos SSP has been selected, InitializeSecurityContext generates a KRB_AP_REQ message from the client. The application on the server's side of the connection responds with the SSPI method AcceptSecurityContext, which generates a KRB_AP_REP message from the server. After the connection has been authenticated, the LSA on the server uses information from the client's session ticket to build an access token. The server then invokes the SSPI method ImpersonateSecurityContext to attach the access token to an impersonation thread for the service.

After the LSA establishes the security context for an interactive user, another instance of the Kerberos SSP might be loaded by a process running in the user's security context to support message signing and sealing.

All distributed services in Windows 2000 domains use the SSPI for authentication. Thus, all domain services support the Kerberos protocol. Services that use the Kerberos protocol for authentication include:

- Print spooler services.
- Common Internet File System/Server Message Block (CIFS/SMB) remote file access.
- Lightweight Directory Access Protocol (LDAP) queries to Active Directory.
- Distributed file system management and referrals.
- Internet Protocol Security (IPSec) host-to-host security authority authentication.
- Reservation requests for network Quality of Service.
- Intranet authentication to Internet Information Services (IIS).
- Remote server or workstation management using authenticated remote procedure call (RPC).
- Certificate requests to the Certificate Services for domain users and computers.

Credentials Cache

Windows 2000 stores tickets and keys obtained from the KDC in a credentials cache, an area of volatile memory protected by the LSA. Only processes running in the LSA's security context have access to the cache. Its memory is never paged to disk. All objects stored there are destroyed when a security principal logs off or the system is shut down.

The credentials cache is managed by the Kerberos SSP, which runs in the LSA's security context. Whenever tickets and keys must be obtained or renewed, the LSA calls the Kerberos SSP to accomplish the task.

The credentials cache is also used to store a copy of an interactive user's password-derived key. If the user's TGT expires during a logon session, the Kerberos SSP uses its copy of the password-derived key to obtain a new TGT without interrupting the user's logon session. The password-derived key is not stored permanently on the computer, and the local copy in the credentials cache is destroyed when the credentials cache is flushed.

Password-derived keys for services and computers are handled differently. They are stored in a secure area of the computer's registry, just as they were in of Windows NT. Password-derived keys for user accounts on the local system, which are used only for access to computers in stand-alone mode, are also stored in the registry. These keys are never used for network access.

Finding the KDC

When the Kerberos SSP wants to send an initial authentication request to the KDC in a user's account domain, it must locate a domain controller for that domain. It finds the domain controller by using the domain controller locator. For more information about the locator, see "Active Directory Logical Structure" in this book.

The locator can only find KDCs that are in Active Directory–based domains. When computers that are running Windows 2000 participate in other Kerberos realms, the Domain Name System (DNS) names for KDC servers must be stored in the client computer's registry. The Kerberos SSP looks in the registry for the DNS domain name of the user's Kerberos realm and then resolves this name to an Internet Protocol (IP) address by querying a DNS server.

IP Transport

According to RFC 1510, when a client contacts the KDC, the client should send a User Datagram Protocol (UDP) datagram to port 88 at the KDC's IP address. The KDC should respond with a reply datagram to the sender's IP address.

UDP is a connectionless transport protocol, which makes it a logical choice when an exchange of messages must precede a connection. UDP is also well suited to applications that send one message and expect one response, such as the exchange between a client and the KDC. However, UDP works best when each datagram is transmitted as a single unit—that is, within a single frame. The Maximum Transmission Unit (MTU) for an Ethernet frame is 1500 octets. If the physical network is Ethernet, Kerberos messages that are sent as UDP datagrams can carry up to 1500 octets of data.

Windows 2000 authorization data can easily total more than 1500 octets. Because this data is needed only by computers that are running Windows 2000, it is omitted from session tickets to computers with other operating systems. As a result, messages to other systems are well within the limits of UDP transport, and that is how they are transmitted. Messages that carry session tickets for computers that are running Windows 2000 are transmitted by using the Transmission Control Protocol (TCP), which has a much greater capacity than UDP. The use of TCP transport in Windows 2000 is consistent with recently proposed revisions to RFC 1510.

Authorization Data

The Kerberos protocol is a protocol for authentication, not authorization. It verifies that security principals are who they say they are but does not determine the objects to which they can have access or what type of access they can have. The Kerberos protocol provides a field for authorization data in session tickets, but it does not specify the form of the data or how servers should use it. These decisions are left to whatever access control mechanism is available on the system.

Name-based vs. Identity-based Authorization

On some operating systems, applications are required to implement their own mechanisms for determining the level of a user's authorization. Applications often do this by maintaining private lists that contain the names of users who are authorized access. Database applications, for example, often maintain private authorization tables to control the fields in a record that a particular user can view or change. This kind of access control mechanism can be integrated with Kerberos authentication simply by ensuring that the authorization data field of a session ticket carries some form of the security principal's name.

Unfortunately, private authorization mechanisms are just that—private. A database application is powerless to prevent an unauthorized user from running another application and using it to edit the data file. On Windows 2000, access to resources is controlled by the operating system itself. Applications can implement their own private authorization mechanisms, but whether they do or not, the Windows 2000 operating system's access control mechanism protects all files and other objects that can be shared by two or more processes.

The header of every securable object includes a security descriptor with an ACL. The ACL is maintained by the object's owner, who decides which security principals can have access to the object and how they can gain access to it. The operating system enforces the owner's decisions by performing an access check whenever an application requests a handle to a protected object. Before the operating system returns the handle, it examines the object's ACL to see whether the security principal that is using the application is authorized access. If the security principal is not authorized access, the application is denied access.

Another important difference from other access control mechanisms is that security principals are not identified by name, either by the operating system or in ACLs. Instead, each security principal is assigned a unique security identifier (SID), an alphanumeric value with a structure similar to a telephone number. Like the country/region code used in international calling, the first part of a SID identifies the domain where the SID was issued. Like the number for a specific telephone within a country or region, the second part of a SID identifies an account within the issuing domain. The value for a domain is unique within an enterprise, and the value for an account is unique within a domain. Unlike telephone numbers, however, SIDs are never reused. There is no possibility that a user might be assigned a SID that once belonged to another user.

A third important difference from name-based access control is that authorization is determined by not only the user's identity but also the user's membership in one or more security groups. In fact, the preferred method of controlling resources is to grant access to groups rather than to individuals, adjusting the level of a group's authorization according to the needs of its members. This method of controlling access makes it easier to keep ACLs up-to-date on networks that have thousands of users and millions of objects. Group membership can be managed centrally by administrators, who can change a particular user's level of authorization for many resources simply by adding or removing the user from a group.

Windows 2000 makes resource security still easier to manage by allowing groups to be nested. A group created in one domain can be added to the membership of a group created in another domain or to the membership of a universal group used throughout a tree of trusted domains. As a result, when employees change jobs, their level of authorization can be changed throughout the enterprise without touching the ACLs on specific objects.

Like individual security principals, Windows 2000 security groups also have SIDs. A user's level of authorization is determined, then, by a list of SIDs—the SID for the user and the SID for each security group to which the user belongs.

For more information how the operating system determines a user's level of authorization, see "Access Control" in this book.

How the KDC Prepares Authorization Data

When the Kerberos protocol is used for authentication, the list of SIDs that identify a security principal and the groups to which this principal belongs is transported to the local computer in the authorization data field of a session ticket. Authorization data is gathered in two separate steps—when the KDC in a Windows 2000 domain prepares a TGT and when the KDC prepares a session ticket for a server in the domain.

When a user requests a TGT, the KDC in the user's account domain queries the domain's Active Directory. The user's account record includes an attribute for the user's SID as well an attribute with SIDs for any domain security groups to which the user belongs. The list of SIDs returned by the KDC's query is placed in the TGT's authorization data field. In a multiple-domain environment, the KDC also queries the Global Catalog for universal groups that include the user or one of the user's domain security groups. If it finds any, the SIDs for these groups are added to the list in the TGT's authorization data field.

When the user requests a session ticket for a server, the KDC in the server's domain copies the contents of the TGT's authorization data field to the session ticket's authorization data field. If the server's domain is different from the user's account domain, the KDC queries Active Directory to find out whether any security groups in the local domain include the user or one of the user's security groups. If there are any such groups, their SIDs are added to the list in the session ticket's authorization data field.

How Services Use Authorization Data

In Windows 2000, services act in their own security contexts only when they are trying to gain access to resources on their own behalf. For the most part, this happens only when they are doing their own housekeeping—reading or writing configuration data stored in registry keys, binding to communications ports, and completing other tasks that are not related to work for a particular client. When a service is doing something for a client, it impersonates the client and acts in the client's security context. This means that in addition to verifying the identity of a client, Windows 2000 services also must take on some of their client's characteristics—specifically, the client's level of authorization on the system.

When a service sets up housekeeping on a computer that is running Windows 2000, it calls the SSPI method AcquireCredentialsHandle to gain access to its own credentials (the secret key for the account under which the service runs). The service then binds to a communications port, where it listens for messages from prospective clients.

When a client requests a connection and presents a session ticket, the service asks the Kerberos SSP to verify the client's credentials by calling the SSPI method AcceptSecurityContext and passing as an argument the client's session ticket along with a handle to the service's secret key. The SSP verifies the ticket's authenticity, opens it, and passes the contents of the authorization data field to the LSA. If the data includes a list of SIDs, the LSA uses them to build an access token that represents the user on the local system. In addition, the LSA queries its own database to determine whether the user or one of the user's security groups is a member of a security group created on the local system. If any such groups are found, the LSA adds these SIDs to the access token. The LSA then confirms for the calling service that the client's identity has been authenticated and encloses a reference to the client's access token.

Upon receiving confirmation, the service completes its connection with the client and attaches the client's access token to an impersonation thread by calling ImpersonateSecurityContext. When the impersonation thread requires access to an object, it presents the client's token. The operating system performs an access check by comparing the SIDs in the client's token to the SIDs in the object's ACL. If the operating system finds a match, it checks to see that the entry in the ACL grants the level of access requested by the thread. If the authorized level of access matches the level of access requested, the thread is allowed access to the object. Otherwise, access is denied.

Why Authorization Data Is Signed

Session tickets are encrypted with the secret key for the account under which the service starts. When a service acquires a handle to its own credentials on the system, it gains access to that secret key. The difficulty is that an unscrupulous user with a legitimate network account but limited authorization on the local computer could install a rogue service on the computer. This user could request a session ticket for the service, and the service could decrypt the ticket, modify the authorization data by adding the SID for a privileged group, encrypt the altered ticket, and present it to the LSA in a call to AcceptSecurityContext. The result would be to elevate the user's level of authorization on the computer where the service is running.

To prevent tampering, authorization data is signed by the KDC before it is stored in a session ticket. Any attempt to alter the data invalidates the signature. The LSA on a Windows 2000 computer always checks the signature on authorization data in session tickets that untrusted services present in calls to AcceptSecurityContext. As far as the LSA is concerned, an untrusted service is any service that is not running under the Local System account. This account is used by services installed with the operating system—by the native Server service, for example. Other services can be configured to use the Local System account, but this must be done by a member of the Administrators group. The assumption is that the administrator who installs the service can vouch for its security.

If a session ticket is presented by an application that is not running as Local System, the LSA asks the KDC in its domain to verify the signature on the ticket's authorization data. The question is asked and answered by an RPC over Net Logon's secure channel to the domain controller. Requests for verification do not need to travel beyond the local domain because session tickets are always issued —and, therefore, authorization data is always signed—by the KDC in the destination computer's domain.

Logging on Interactively

Users have a natural tendency to think that logging on to an account in a Windows 2000 domain gives them access to the network. That is, of course, not true. When the Kerberos protocol is used for network authentication, what you actually get when you first log on is access to the domain's authentication service. Specifically, you get a TGT that you can present when requesting session tickets for other services in the domain.

When you log on to a domain account from a computer running Windows 2000, you always need at least one session ticket—a ticket for the computer where you are logging on. The reason for this is quite simple. You cannot use a computer that is running Windows 2000 without using at least some system services. When you use a system service, you become a client of the service, and to become a client you must first be authenticated by the service. For that to happen, you'll need a session ticket. On computers running Windows 2000, system services run under the Local System account on the computer, and when the computer is joined to a domain, these services participate in the domain by using the computer's domain account. Domain users gain admission to services running as Local System by presenting a session ticket for the computer where the services are running. Domain users who log on interactively are no exception. They, too, must have a session ticket for the computer before they are allowed access to the computer's services.

Logon Process

When a user with an account in a Windows 2000 domain logs on at the keyboard of a computer that is running Windows 2000, the user's logon request is processed in three stages:

1. The user asks for admission to the ticket-granting service for the domain.

   This is accomplished through an AS Exchange between the Kerberos SSP on the computer and the KDC in the user's account domain. The result is a TGT that the user can present in future transactions with this KDC.

2. The user asks for a ticket for the computer.

   This is accomplished through a TGS Exchange between the Kerberos SSP on the computer and the KDC for the computer's account domain. The result is a session ticket that the user can present when he or she requests access to system services on the computer.

3. The user asks for admission to Local System services on the computer.

   This is accomplished when the Kerberos SSP on the computer presents a session ticket to the LSA on the computer.

If the computer's account domain is different from the user's account domain, an extra step is involved. Before the Kerberos SSP can request a session ticket for the computer, it must ask the KDC in the user's domain for a TGT that is good for admission to the KDC in the computer's domain. The SSP can then present the TGT to the KDC in the computer's domain and get a session ticket for the computer.

Exactly how the logon process works depends on how you configure the computer. With standard configurations of Windows 2000, interactive users log on with a password. In another optional configuration of Windows 2000, users log on with a smart card. Although the basic process is the same for both configurations, there are some differences.

Logging on with a Password

Suppose Alice has a network account in the domain named West. The computer she usually uses, Workstation, also has an account in West. When Alice logs on to the network, she begins by pressing the key combination CTRL+ALT+DEL, which is the Secure Attention Sequence (SAS) on computers with a standard Windows 2000 configuration.

In response to the SAS, Winlogon switches to the logon desktop and dispatches to a DLL called the Graphical Identification and Authentication (GINA), a component loaded in Winlogon's process. GINA is responsible for collecting the logon data from the user, packaging it in a data structure, and sending everything to the LSA for verification. Third parties can develop replacement GINAs, but in this case Winlogon has loaded the standard component (MSGINA.dll) supplied with the Windows 2000 operating system. MSGINA displays the standard logon dialog box.

Alice types her user name and password. She selects **West** from the list of domain names. When she clicks **OK** to dismiss the dialog box, MSGINA returns her logon information to Winlogon. Winlogon then sends the information to the LSA for validation by calling LsaLogonUser.

Upon receiving a data structure with Alice's logon data, the LSA immediately converts Alice's plaintext password to a secret key by passing it through a one-way hashing function. It saves the result in the credentials cache, where the hashed password can be retrieved when it is needed for encryption and decryption.

To validate Alice's logon information and set up her logon session on the computer, the LSA must obtain the following:

- A TGT that is good for admission to the ticket-granting service.
- A session ticket that is good for admission to the computer.

The LSA gets these tickets by working through the Kerberos SSP, which exchanges messages directly with the domain's KDC, as illustrated in Figure 11.8.

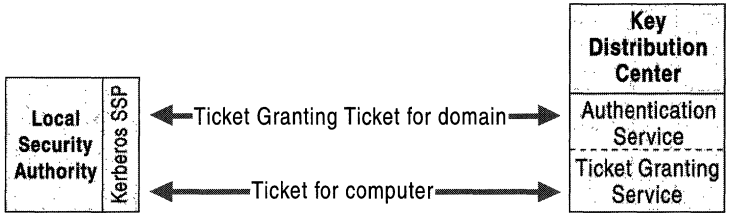


Figure 11.8 Kerberos Exchanges for Logging on Interactively

The messages follow this sequence:

1. The LSA sends a KRB_AS_REQ message to the KDC's authentication service in the domain West.

   The message includes:

   - The user's principal name (Alice).
   - The name of the account domain (West).
   - Preauthentication data encrypted with the secret key derived from Alice's password.

2. The KDC's authentication service replies with a KRB_AS_REP message.

   The message includes:

   - A session key for Alice to share with the KDC, encrypted with the secret key derived from Alice's password.
   - A TGT for the KDC in the domain West, encrypted with the KDC's secret key. The TGT includes a session key for the KDC to share with Alice and authorization data for Alice.

     The authorization data includes the SID for Alice's account, SIDs for security groups in the domain West that include Alice, and SIDs for universal groups in the enterprise that include either Alice or one of her domain groups.

3. The LSA sends a KRB_TGS_REQ message to the KDC's ticket-granting service in the domain West.

   The message includes:

   - The name of the destination computer (Workstation).
   - The name of the destination computer's domain (West).
   - Alice's TGT.
   - An authenticator encrypted with the session key Alice shares with the KDC.

4. The KDC replies with a KRB_TGS_REP message.

   The message includes:

   - A session key for Alice to share with Workstation, encrypted with the session key Alice shares with the KDC.
   - Alice's session ticket to Workstation, encrypted with the secret key Workstation shares with the KDC.

     The session ticket includes a session key for Workstation to share with Alice and authorization data copied from Alice's TGT.

Upon receipt of Alice's session ticket, the LSA decrypts it with the computer's secret key and extracts her authorization data. It then queries the local Security Accounts Manager (SAM) database to determine whether Alice is a member of any security groups local to the computer and whether she has been given any additional user rights on the local computer. It adds any SIDs returned by this query to the list taken from the ticket's authorization data. The entire list is then used to build an access token. A handle to the access token is then returned to Winlogon, along with the identifier for Alice's logon session and confirmation that her logon information is valid.

Winlogon creates a window station and several desktop objects for Alice, attaches her access token, and starts the shell process she will use to interact with the computer. Alice's access token is subsequently inherited by any application process that she starts during her logon session.

Logging on with a Smart Card

In standard logons, a user initially proves his or her identity to the KDC by demonstrating knowledge of a secret known only to the user and to the KDC. This shared secret is the cryptographic key derived from the user's password. The same key is used for both encryption and decryption. For this reason, shared secret keys are said to be symmetric. The cryptographic key derived from the user's password is used only during the AS Exchange for the following:

- When the client encrypts preauthentication data.
- When the KDC decrypts preauthentication data.
- When the KDC encrypts the logon session key.
- When the client decrypts the logon session key.

To support logging on with a smart card, Windows 2000 implements a public key extension to the Kerberos protocol's initial AS Exchange. In contrast to shared secret key cryptography, public key cryptography is *asymmetric*; that is, two different keys are needed—one to encrypt, another to decrypt. Together, the keys needed to perform both operations make up a private/public key pair.

When a smart card is used in place of a password, a private/public key pair stored on the user's smart card is substituted for the shared secret key derived from the user's password. The private key is known only to the owner of the pair and is never shared. The public key can be made available to anyone with whom the owner wishes to exchange confidential information.

In the public key extension to the Kerberos protocol, the initial AS Exchange is modified so that the KDC encrypts the user's logon session key with the public half of the user's key pair. The client decrypts the logon session key with the private half of the pair.

The logon process begins when the user inserts a smart card into a card reader attached to the computer. When a computer that is running Windows 2000 is configured for smart card logon, the insertion of the card signals the SAS, just as the key combination CTRL+ALT+DEL signals the SAS on computers configured for password logon. In response, Winlogon dispatches to MSGINA, which displays a logon dialog box. But in this case, the user types just one item of information, a personal identification number (PIN).

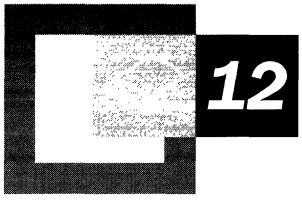
MSGINA sends the user's logon information to the LSA by calling LsaLogonUser, just as it does with a password logon. The LSA uses the PIN for access to the smart card, which contains the user's private key along with an X509 v3 certificate that contains the public half of the key pair. All cryptographic operations that use these keys take place on the smart card.

The Kerberos SSP on the client computer sends the user's public key certificate to the KDC as preauthentication data in its initial authentication request, the KRB_AS_REQ message. The KDC validates the certificate, extracts the public key, and then uses the public key to encrypt a logon session key. It returns the encrypted logon session key and a TGT in its KRB_AS_REP reply message to the client. If the client is in possession of the private half of the key pair, it can use the private key to decrypt the logon session key. Both the client and the KDC then use this logon session key in all further communications with one another. The remainder of the authentication process is the same as for standard logon.

For information about the types of smart cards and smart card readers supported by Windows 2000, see the Microsoft Windows Hardware Compatibility List link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

CHAPTER 12

Access Control



The Microsoft® Windows® 2000 operating system protects files, applications, and other resources from unauthorized use. Although you might already know how to use tools to assign privileges or set permissions, understanding what privileges and permissions really are, why they are necessary, and how they function can help you manage shared resources effectively. Understanding these processes can also help you avoid unnecessary risks and troubleshoot any problems you might encounter.

In This Chapter

Related Information in the Resource Kit

- For information about how to diagnose, troubleshoot and recover Active Directory, see "Active Directory Diagnostics, Troubleshooting, and Recovery" in this book.
- For information about authentication, see "Authentication" in this book.
- For information about user rights, see the appendix "User Rights" in this book.
- For information about well-known security identifiers, see the appendix "Well-Known Security Identifiers" in this book.

Access Control Model

The security systems in Windows 2000 are based on technologies originally developed for Windows NT. Both operating systems control access to resources in fundamentally the same way. If you are familiar with Windows NT 4.0 or earlier versions of Windows NT, you know that its access control model has the following characteristics:

User-based authorization In Windows NT, every application that you start runs in your security context, not in its own security context. The application can do only what you are authorized to do. For example, when you use Microsoft Word, it can access only documents that you are authorized to access. When another user uses Word, it can access only documents that the other user is authorized to access. In Windows 2000, this aspect of the access control model is significantly enhanced. Applications can be designed to run in restricted security contexts, giving them fewer privileges and more limited access than their users.

Discretionary access to securable objects In Windows NT, the user who owns an object can control who has permission to use it and in what way. An object's owner can give permission for different kinds of access to particular users or groups of users. For example, the owner of a file object can give read and write permission to all members of a group while denying write access to a particular member of the group. In Windows 2000, owners can allow or deny other users access to individual properties of certain types of objects as well as to the entire object.

Inheritance of permissions In Windows NT, you can control permissions for new objects created in a container object by setting inheritable permissions on the container. For example, the permissions that you set on an NTFS folder are inherited by new subfolders and files created within the folder. In Windows 2000, the permissions that you set on a container are inherited by existing objects in the container as well as newly created objects.

Administrative privileges In Windows NT, you can control which users or groups have the right to perform various administrative functions or to take any action that affects systemwide resources. For example, an administrator can give one group of users the right to log on interactively, and give a more select group of users the right to load and unload device drivers. In Windows 2000, you can centrally manage administrative privileges on all computers joined to a domain by using Group Policy.

Auditing of system events In Windows NT, you can use the auditing feature to detect attempts to circumvent protections on resources or to create an audit trail of administrative actions on the system. For example, you can set security policy so that failed logon attempts are recorded in the security event log. If another administrator changes the auditing policy so that failed logon attempts are no longer audited, the log shows this event too. In Windows 2000, you can use Group Policy to centrally who is allowed to manage security logs on computers joined to a domain.

Key Terms

Access control, like all technologies, has a specialized vocabulary. It is recommended that you understand how the following key terms are defined in the context of the access control model for Windows 2000.

Security principal. A user, group, computer, or service. Security principals have accounts. Local accounts are managed by the Security Accounts Manager (SAM) on the computer. Domain accounts are managed by Active Directory.

Security identifier (SID). A value that uniquely 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.

Security context. Information that describes a particular security principal's identity and capabilities on the computer. In Windows 2000, all activities take place in a security context. The security subsystem uses the security context to determine what a process and its threads of execution can do to objects on the computer, and who will be held accountable for what they have done.

Access token. A data structure containing the SID for a security principal, SIDs for the groups that the security principal belongs to, and a list of the security principal's privileges (also known as user rights) on the local computer. An access token is created for every security principal who logs on locally, at the computer's keyboard, or remotely, through a network connection. The access token provides a security context for the security principal's actions on the computer.

Thread. An executable entity within a process. For the purpose of this discussion, think of a thread as a sequence of program code scheduled for execution on the processor. A process can have several threads, all executing at the same time. The operating system coordinates execution by assigning each thread a scheduling priority.

Subject. A thread executing within the security context of an authenticated security principal. Before a subject is allowed to carry out an operation on a protected object, the security subsystem first performs an access check by comparing information in the subject's access token with information in the object's security descriptor to determine whether the subject is authorized to do what it wants to do.

Impersonation. The ability of a thread to act in a security context different from that of the process that owns the thread. Impersonation is designed to address the security requirements of client/server applications by allowing a service to act in its own security context when acting on its own behalf and in a client's security context when acting on the client's behalf.

Object. Any resource that can be manipulated by a program or process. Objects include resources you can see through the user interface, such as files, folders, printers, registry keys, Active Directory objects, and the Windows Desktop. They also include resources that you cannot see, such as sessions, processes, threads, and access tokens. An object can function as a logical repository for other objects. If an object can hold other objects, it is called a container; if it does not have this capability, it is called a noncontainer. Container objects can hold both noncontainer objects and also other container objects. For example, a Folder object in the file system can contain File objects (noncontainers) and also other Folder objects (containers). In a hierarchy of objects, the relationship between a container and its content is expressed by referring to the container as the *parent* and to an object in the container as the child. The parent-child metaphor is particularly useful in describing the concept of object inheritance, which allows a child object to inherit certain characteristics, such as security restrictions, from its parent object.

Securable object. Any object that can be shared. In Windows 2000, all securable objects have security descriptors that contain information about how access to the object is controlled.

Security descriptor. A data structure containing the security information associated with a securable object. A security descriptor identifies an object's owner by SID. If permissions are configured for the object, its security descriptor contains a discretionary access control list (DACL) with SIDs for the users and groups who are 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.

Owner. The only security principal who has an inherent right to allow or deny permission to access an object. The first owner of an object is usually the security principal associated with the thread that created the object. An object's owner can transfer ownership by giving another security principal permission to take ownership. By default, the built-in Administrators group on a computer is assigned a user right that allows this group to take ownership of all objects on the computer.

Permission. Authority to perform an operation or a set of operations on an object. Permissions are granted or denied by an object's owner. Because access to an object is 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. Users can (and usually do) have different user rights on different computers.

Access right. A permission from a subject's point of view. When a human user allows or denies permission through the Access Control Settings dialog box, the result is recorded as an access control entry (ACE) in the object's DACL. In the user interface, a permission is represented by a word or phrase. In an ACE, the same permission is represented by a set of bit flags in an access mask. Each bit flag corresponds to an access right—that is, to a particular operation that can be performed on the object.

How Access Control Works

The details of how access control works are quite complex, but the big picture is fairly simple: Subjects act on objects. In the sentence, "Alice opens the file," Alice is the subject, or the agent of an action; opens is the action; and the file is the object. The grammar is similar in Windows 2000.

However, there are some important differences. When we say "Alice opens the file," we know that it isn't really Alice who opens the file; it's done by a program. To be more precise, the program runs as a process with threads of execution. It is actually one of those threads that opens the file. Threads are the only real agents of action on a computer. In the grammar of access control, the subject is always a thread.

In order for a thread to gain access to an object, it must identify itself to the operating system's security subsystem. A thread does not have a security identity, so it must borrow one from a security principal, such as Alice. When Alice logs on, her security identity is encapsulated in an access token that is associated with her logon session. When Alice starts an application, it runs as a process within her logon session. The application process and each of its threads of execution receive copies of Alice's access token. When one of the application's threads needs to open a file, the thread identifies itself as Alice's agent by presenting her access token. Thus responsibility for anything that the thread does to the file on Alice's behalf is charged to Alice.

Of course threads do not open files in the same way that users do. Threads are simply pieces of program code executing on the computer, and they interact with objects through one of several application programming interfaces (APIs) provided by the operating system. If a thread's job is to open a file, its code might contain the following instruction:

```
hfile=CreateFile(pszFile,GENERIC_WRITE,0,NULL,OPEN_EXISTING,0,NULL);
```

The second argument in the call to CreateFile specifies a desired set of access rights, GENERIC_WRITE, which indicates to the operating system that the thread wants to open the file and modify it. Other APIs work in a similar fashion. The caller must signal its intentions for an object by specifying a desired level of access.

Before allowing the thread of execution to proceed, the operating system performs an access check to determine whether the security principal associated with the thread is authorized the level of access that the thread has requested. 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 users or groups.

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 12.1 illustrates this process.

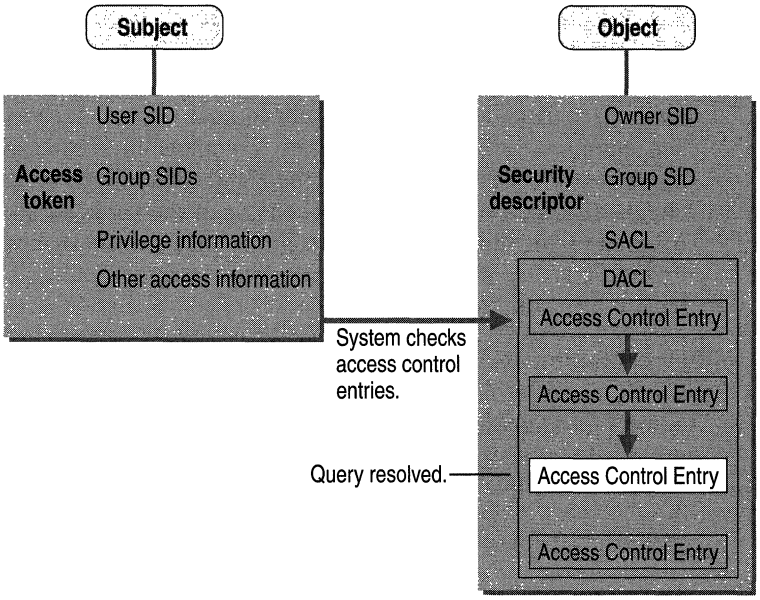


Figure 12.1 Validating a Request for Access

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 access-checking reaches the ACE that allows access to the user's group before it reaches the ACE that denies access to the user, the user is allowed to access the object. This is clearly not a desirable outcome.

In general, ACEs are listed in what is called *canonical* order, which places deny ACEs before allow ACEs. When the canonical order is used, an access check processes all ACEs that deny access before any ACE that allows access.

However, there are cases where the canonical order does not serve the purpose. For example, consider the DACL shown in Table 12.1.

Table 12.1 Non-Canonical Order

Type	Name	Permission
Allow	Administrators	Full Control
Deny	Network	Read
Allow	Users	Read

This DACL allows administrators access to the object when they log on locally and over the network but restricts non-administrative users by allowing them access only when they are logged on locally. This DACL is non-canonical, but it does serve a useful purpose, and so can other non-canonical DACLs. Thus non-canonical DACLs are not prohibited, just discouraged because their effects can be hard to understand. You cannot create a non-canonical DACL through the user interface. However, you can do it programmatically.

The canonical order just described is simplified somewhat for the purpose of this overview. It does not, for example, account for the ordering of ACEs inherited from a parent object. For the precise canonical order, see "Order of ACEs in a DACL" later in this chapter.

Rights

A right is authorization to perform an operation. In Windows 2000, only one right is inherent—the right to allow or deny access to resources that you own. All other rights must be granted, which means that they also can be withdrawn. From an administrator's point of view, there are two types of rights: permissions and user rights.

Permissions

A *permission* is authorization to perform an operation on a specific object, such as a file. Permissions are granted by owners. If you own an object, you can grant any user or security group permission to do whatever you are authorized to do with it. This includes granting permission to take ownership.

Tip Although you can give permissions to individual users, it is more efficient to give them to a security group. 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 Alice allows the Marketing group, and only the Marketing group, permission to read her file, users who are not members of the Marketing group are implicitly denied access. The operating system will not allow users who are not members of the Marketing group to read the file.

Permissions can also be explicitly denied. For example, Alice might not want Bob to be able to read her file, even though he is a member of the Marketing group. She can exclude Bob by explicitly denying him permission to read the file. In fact, this is exactly how explicit denials are best used—to exclude a subset (such as Bob) from a larger group (such as Marketing) 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 display as **Permission Entries** in the Access Control Settings dialog box. Figure 12.2 shows the ACEs in the DACL for Alice's file.

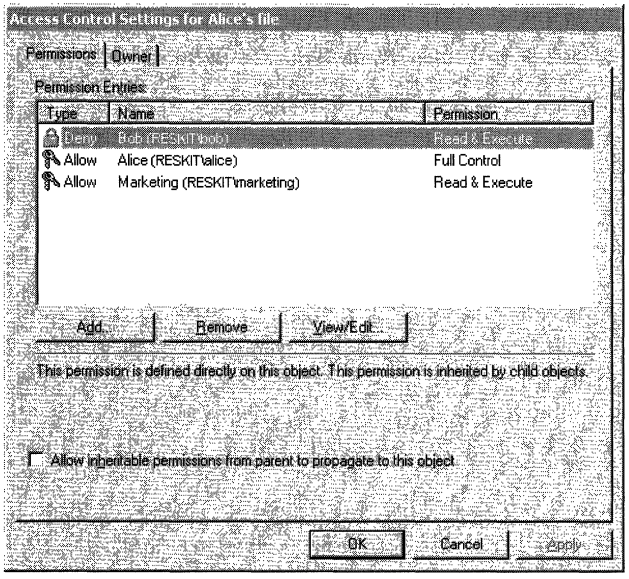


Figure 12.2 Permissions for Alice's File

Note You can set permissions at the file level only if the files are stored on an NTFS volume. On a FAT volume, you can set permissions for shares but not for files and folders within a share. Moreover, share permissions on a FAT volume restrict network access only, not access by users working directly on the computer. There is no good reason to use FAT on a shared volume. It is recommended that you use an NTFS volume.

Setting Permissions on Active Directory Objects

You can set permissions for an object in Active Directory just as you do for an object on an NTFS volume. Right-click the object, then click **Properties**. When the Properties dialog box displays, click the **Security** tab.

If you are using Active Directory Users and Computers, the Properties dialog box might not have a Security tab. Viewing the security settings for an object is considered an advanced feature of Active Directory Users and Computers. If the Properties dialog box for the object you have selected does not have a Security tab, click **Cancel**. In Active Directory Users and Computers, click the **View** menu, then click **Advanced Features**.

The Security tab shows only a high-level view of an object's access control settings. For complete information, including information on how permissions set on a container object are inherited by child objects, click **Advanced**. In the Access Control Settings dialog box, click the **Permissions** tab. If the object is a container, the **Permission Entries** list indicates which permissions apply only to the object itself (that is, only to the container), which apply only to child objects (that is, only to objects in the container), and which apply only to certain types of child objects (only to User objects, for example).

This aspect of the user interface reflects an important difference between how permissions are propagated from container objects in NTFS and how they are propagated from container objects in Active Directory. If you own a folder object in an NTFS volume, you can control who has access to the folder and all that it contains by setting a permission on the folder that applies not only to the folder but to all objects in the folder. For example, you can add a permission to the Folder object that allows, say, the Marketing group read access. If you apply the permission to **This folder, subfolders and files**, it is propagated to all objects contained within the folder. The Marketing group then has read access to the folder and to all its subfolders and files. Propagation of permissions works a bit differently for Active Directory objects. If you own a container object in Active Directory, you have the option to allow access to certain types of objects in the container without allowing access to other types of child objects. For example, if you own an Organizational Unit (OU) object, you can add a permission that allows the Marketing group read access and then apply the permission to a particular type of child object contained by the OU. If you apply it to Contact objects, for example, the permission is propagated only to Contact objects but not to User objects or to any other type of child object. In other words, permissions propagated from container objects in Active Directory can be *object-specific*, whereas permissions propagated from container objects in an NTFS volume cannot.

Viewing and Setting Per-Property Permissions

You can manage permissions for objects in Active Directory at two levels:

- **At the object level.** Permissions allowed or denied at the object level apply to the entire object. For example, you can set an object-level permission on an OU object that allows a particular group, such as Account Operators, to create child objects in the OU.

- **At the property level**. Permissions allowed or denied at the property level apply only to specific properties. For example, you can set a property-level permission on a User object that allows a particular user, such as Principal Self (that is, the user represented by the object), to change the Home Address property of the User object.

Per-property permissions cannot be managed directly from the Access Control Settings dialog box, but that is a good place to start. To open the Access Control Settings dialog box, select an object in one of the tools used to manage Active Directory, such as Active Directory Users and Computers. Right-click the object, then click **Properties**. When the Properties dialog box displays, click the **Security** tab, then click **Advanced**.

To add new per-property permissions, click **Add**. When the Permission Entry dialog box displays, click the **Properties** tab. The list of available per-property permissions is displayed under **Permissions**.

To view or edit existing per-property permissions, open the Access Control Settings dialog box, click an item in **Permission Entries**, and then click **View/Edit**. When the Permission Entry dialog box displays, click the **Properties** tab. Per-property permissions are listed under **Permissions**. The per-property permissions allowed and denied to the Authenticated Users group for the user Alice are shown in Figure 12.3.

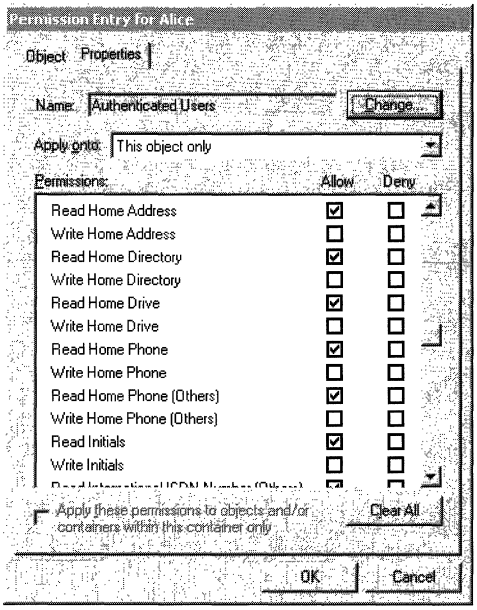


Figure 12.3 Per-Property Permissions for a User Object

The **Properties** tab does not list every property of an object, only those that are commonly used for controlling access. The properties of objects in Active Directory and the object types themselves are defined in the schema. The number of object types and properties (or *attributes*) is quite large. The user interface for access control filters out object types and properties in order 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
```

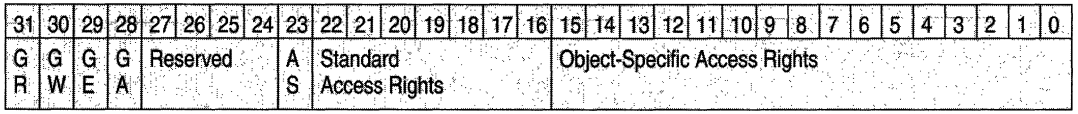
Object types are in brackets. If an "@" sign is on the next line below the bracketed object type, and it is set to 7, then the object type is filtered. To stop filtering for that object type, change the setting to 0. The names of filtered attributes are listed below the object type. If an attribute name is set to 7, it is filtered. To stop filtering, change the setting to 0.

When you change Dssec.dat, your changes are not reflected on the Properties tab until you close Active Directory Users and Computers—or whatever other tool you are using—and restart it. Filter data is read when the tool initializes.

For more information on object types and their attributes, see "Active Directory Schema" in this book.

Access Masks

In an ACE, permissions are represented by one or more bits in a 32-bit value called an access mask. When a thread requests access to an object, it specifies the type of access that it desires by using an access mask as well. During an access check, the operating system compares the desired access mask supplied by the thread with the access mask in each ACE of the object's DACL, looking for bits that match. Figure 12.4 illustrates the layout of an access mask.



Key

GR	Generic Read
GW	Generic Write
GE	Generic Execute
GA	Generic All
AS	Right to access SACL

Figure 12.4 Layout of an Access Mask

Each bit corresponds to an access right—a particular operation or set of operations that can be performed on the object. Turning a bit on in a desired access mask signals that the thread requests the right to perform the corresponding operation. Turning a bit on in the access mask for an ACE signals that the corresponding operation is either allowed or denied, depending on the type of ACE.

Although many of the access rights in an access mask correspond to permissions that can be set through the user interface, not all of them do. Some can be set only by a process, such as the process that creates an object, and cannot be manipulated through the user interface. Windows 2000 uses an access mask format that includes the following types of access rights:

Generic access rights These rights apply to all objects, but they mean different things for different object types. Each object type maps each generic access right to a different set of standard and object-specific access rights. Table 12.2 lists generic access rights

Table 12.2 Generic Access Rights

Constant in Win32 API	Meaning
GENERIC_ALL	Read, write, and execute access
GENERIC_EXECUTE	Execute access
GENERIC_READ	Read access
GENERIC_WRITE	Write access

Standard access rights These rights are more specific than generic access rights but still apply to operations available for most object types. Table 12.3 lists standard access rights.

Table 12.3 Standard Access Rights

Constant in Win32 API	Meaning
DELETE	The right to delete the object.
READ_CONTROL	The right to read the information in the object's security descriptor, not including the information in the SACL.
SYNCHRONIZE	The right to use the object for synchronization. Some object types do not support this access right.
WRITE_DAC	The right to modify the DACL in the object's security descriptor.
WRITE_OWNER	The right to change the owner in the object's security descriptor.

SACL access right This right controls the ability to read or change an object's SACL, which controls auditing for the object. The operating system allows access to the SACL only if the subject's access token includes the **Manage auditing and security log** (SeSecurityPrivilege) privilege.

Object-specific access rights Each type of securable object defines its own set of access rights. For a list of the specific access rights that apply to each type of object, see the Microsoft Platform Software Development Kit (SDK) link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Extended Rights

In Active Directory, you can control which users have the right to perform a particular operation on an object or one of its properties just as you do in the NTFS file system—by setting permissions. However, there are certain operations that have semantics that are not tied to specific properties. These operations might also require access control. For example, you can grant users a Send As right that an email program can use to determine whether a particular user can send mail on another user's behalf. Access controls on custom actions or operations are called *extended rights*.

Extended rights are not defined by an access mask. Instead, each extended right is identified by a globally unique identifier (GUID). This GUID corresponds to a controlAccessRight object that is stored in the Extended-Rights container within a forest's Configuration container. An ACE that grants an extended right specifies a GUID corresponding to a particular controlAccessRight object.

The list of extended rights is not fixed. Developers can create new extended rights for custom operations by adding controlAccessRights objects to the Extended-Rights container for a forest. For information about creating extended rights, see the Microsoft Platform Software Development Kit (SDK) link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

You can view the Configuration container and see the extended rights defined for your forest by using ADSI Edit. To see the complete set of extended rights defined for your forest, open ADSI Edit, connect to the Configuration container, and then click the Extended-Rights container. Figure 12.5 shows the Extended-Rights container for one forest.

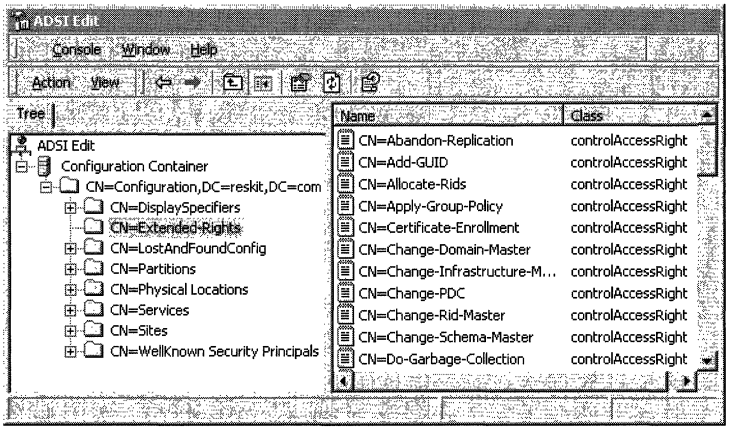


Figure 12.5 Extended-Rights Container

For more information about using ADSI Edit to view the Configuration container, see "Active Directory Data Storage" in this book.

Although the operating system treats extended rights differently from other access rights, the difference is not apparent in the user interface. They display along with ordinary permissions on the Security tab of an object's Properties dialog box. You allow or deny an extended right just as you allow or deny a permission—by checking the appropriate box.

User Rights

A *user right* is authorization to perform an operation that affects an entire computer rather than a specific object on the computer. User rights are divided into two categories: *logon rights* and *privileges*. Logon rights control how human users and other security principals are authorized to access a computer—at the keyboard, through a network connection, as a service, or as a batch job. Privileges control which users are authorized to manipulate system resources—by setting the computer's internal clock, for example, by loading and unloading device drivers, by backing up or restoring files and folders, or by doing anything else that affects the system as a whole. For a complete list of user rights and a description of their default settings, see the appendix "User Rights" in this book.

Unlike permissions, which are granted by an object's owner, user rights are assigned as part of the security policy for the computer. To view the user rights assignment for a computer, log on using an account that has administrative authority, open the Administrative Tools folder in Control Panel, and then start Local Security Policy. Figure 12.6 shows the user rights assignment in the security policy for a computer joined to a domain.

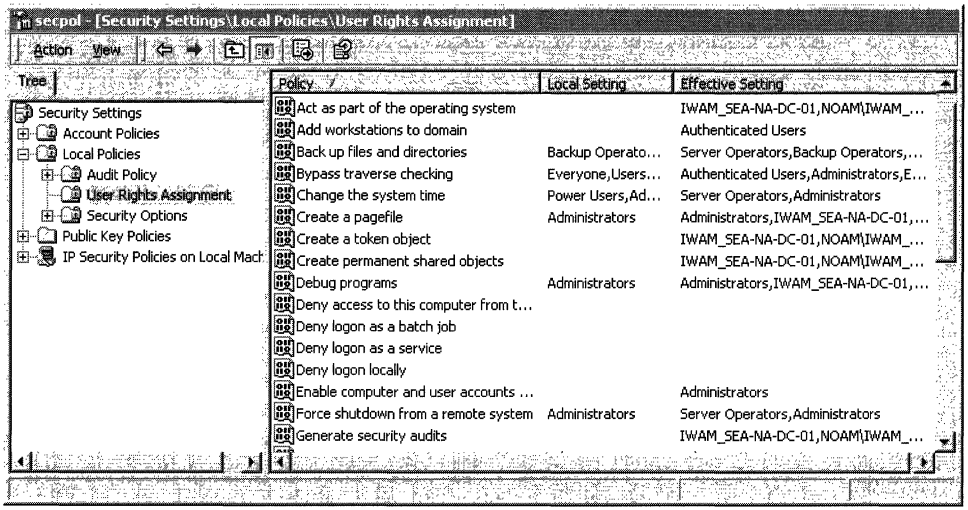


Figure 12.6 User Rights Assignment

For more information about using Group Policy to configure local security policy, see "Group Policy" in this book.

Conflicts Between Privileges and Permissions

For the most part, conflicts between privileges and permissions occur only in situations where the rights required to administer a system overlap the rights of resource ownership. When rights conflict, a privilege overrides a permission.

For example, one common administrative task is backing up files and folders. In order to do its job, backup software must be able to traverse all folders in an NTFS volume, list the contents of each folder, read the attributes of every file, and read data in any file that has its archive attribute set. It would not be practical to arrange this access by coordinating with the owner of every file and folder, so the required rights are included in the **Back up files and directories** (SeBackupPrivilege) privilege, which is assigned by default to two built-in groups: Administrators and Backup Operators. Any user who has this privilege can access all files and folders on the computer for the purpose of backing up the system. The privilege does not give a user the right to access files and folders for any other purpose. A backup operator cannot, for example, use a word processor to open a file if the owner has not granted the backup operator permission to do so.

The ability to take ownership of files and other objects is another case where an administrator's need to maintain the system takes priority over an owner's right to control access. Normally, you can take ownership of an object only if its current owner gives you permission to do so. Owners of NTFS objects can allow another user to take ownership by granting the other user Take Ownership permission. Owners of Active Directory objects can do the same thing by granting another user Modify Owner permission. If the current owner gives you permission, and you do take ownership, you can do whatever you want with the object. You can even deny the previous owner access to it. For this reason, owners are understandably reluctant to give Take Ownership or Modify Owner permission to anyone. However, the people who own objects do sometimes change jobs or leave the company altogether, and they do not always take care to give another user permission to take ownership of resources they leave behind. This is exactly the type of situation for which the **Take ownership of files or other objects** (SeTakeOwnershipPrivilege) privilege is intended. A user who has this privilege can take ownership of an object without the current owner's permission. By default, the privilege is assigned only to the built-in Administrators group. Used correctly, it allows an administrator to take ownership of an abandoned resource and then transfer ownership by granting another user Take Ownership or Modify Owner permission.

Security Identifiers

A security identifier (SID) is a unique value of variable length used to identify a security principal or security group. Windows 2000 uses SIDs in the following access control components:

Access tokens One SID in an access token identifies the user represented by the token. Additional SIDs identify the security groups to which the user 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 user or group for whom access is allowed, denied, or audited.

The SID that identifies a particular account or group is generated by the system at the time the account or group is created. The SID for a local account or group is generated by the Local Security Authority (LSA) on the computer and stored with other account information in a secure area of the registry. The SID for a domain account or group is generated by the domain security authority and stored as an attribute of the User or Group object in Active Directory.

SIDs are unique within the scope of the account or group they identify. The SID for every local account and group is unique on the computer where it was created. No two accounts or groups on the computer ever share the same SID. Likewise, the SID for every domain account and group is unique within an enterprise. The SID for an account or group created in a domain never matches the SID for any other account or group created in the same domain. The SID for an account or group created in one domain of an enterprise never matches the SID for an account or group created in another domain of the same enterprise.

SIDs are also unique for all time. Security authorities never issue the same SID twice, and they never reuse SIDs for deleted accounts. For example, suppose Alice has an account in a Windows 2000 domain, leaves her job, and then later returns to a different job at the same company. When Alice leaves her job, an administrator deletes her account and with it the SID identifying the user Alice. When Alice returns to a new job in the same company, an administrator creates a new account, and Windows 2000 generates a new SID for the user Alice. The new SID does not match the old one, so none of the access that was given to Alice's old account is transferred to Alice's new account. Her two accounts represent two completely different security principals.

Security Identifier Structure

A security identifier (SID) is a data structure in binary format that contains a variable number of values. The first values in the structure contain information about the SID structure. The remaining values are arranged in a hierarchy, like a telephone number, and identify the SID-issuing authority (Windows 2000, for example), the SID-issuing domain, and a particular security principal or group. Figure 12.7 illustrates the structure of a SID.

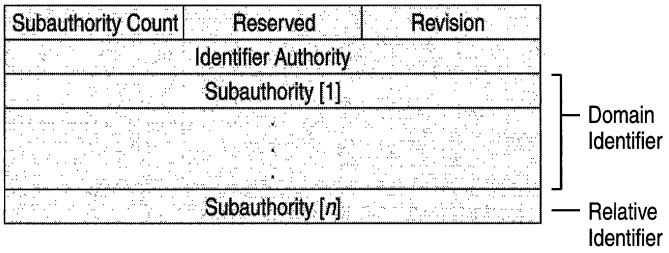


Figure 12.7 SID Structure

The individual parts of a SID are as follows:

Revision This value indicates the version of the SID structure used in a particular SID. The structure used in all SIDs created by Windows NT and Windows 2000 is revision level 1.

Identifier authority This value identifies the highest level of authority that can issue SIDs for this particular type of security principal. For example, the identifier authority value in the SID for the group Everyone is 1 (World Authority). The identifier authority value in the SID for a specific Windows NT and Windows 2000 account or group is 5 (NT Authority).

Subauthorities The most important information in a SID is contained in a series of one or more subauthority values. All values up to but not including the last value in the series collectively identify a domain in an enterprise. This part of the series is the *domain identifier*. The last value in the series identifies a particular account or group relative to a domain. This value is the *relative identifier* (RID).

The components of a SID are easier to visualize when SIDs are converted from binary to string format using standardized notation:

S-R-X-Y^1-Y^2...-Y^{n-1}-Y^n

In this notation, the components of a SID are as follows:

- **S** indicates that the string is a SID.
- **R** is the revision level.
- **X** is the identifier authority value.
- **Y** is a series of subauthority values, where *n* is the number of values. This number corresponds to the Subauthority Count shown in Figure 12.7.

The SID's most important information is contained in the series of subauthority values. The first part of the series (-Y^1-Y^2...-Y^{n-1}) is the domain identifier. This element of the SID becomes significant in an enterprise with several domains, for the domain identifier is what differentiates SIDs issued by one domain from SIDs issued by all other domains in the enterprise. No two domains in an enterprise share the same domain identifier. The last item in the series of subauthority values (-Y^n) is the relative identifier. It is what distinguishes one account or group from all other accounts and groups in the domain. No two accounts or groups in any domain share the same relative identifier.

For example, the SID for the built-in Administrators group is represented in standardized SID notation as the following string:

S-1-5-32-544

This SID has:

- A revision level, 1
- An identifier authority value, 5 (NT Authority)
- A domain identifier, 32 (Builtin)
- A relative identifier, 544 (Administrators)

SIDs for built-in accounts and groups always have the same domain identifier value, 32. This value identifies the domain Builtin, which exists on every computer running Windows NT or Windows 2000. It is never necessary to distinguish one computer's built-in accounts and groups from another computer's built-in accounts and groups because they are local in scope—local either to a single computer or, in the case of domain controllers for a network domain, local to several computers acting as one. However, built-in accounts and groups do need to be distinguished from one another within the scope of the Builtin domain, therefore the SID for each account and group has a unique relative identifier. A relative identifier value of 544 is unique to the built-in Administrators group. No other account or group in the Builtin domain has a SID with a final value of 544.

For another example, consider the SID for the global group Domain Admins. Every domain in an enterprise has a Domain Admins group, and the SID for each group is different. The following is the SID for Reskit\Domain Admins:

```
S-1-5-21-1004336348-1177238915-682003330-512
```

The SID for Reskit\Domain Admins has:

- A revision level, 1
- An identifier authority, 5 (NT Authority)
- A domain identifier, 21-1004336348-1177238915-682003330 (Reskit)
- A relative identifier, 512 (Domain Admins)

The SID for Reskit\Domain Admins is distinguished from the SIDs for other Domain Admins groups in the same enterprise by its domain identifier, 21-1004336348-1177238915-682003330. No other domain in the enterprise uses this value as its domain identifier. The SID for Reskit\Domain Admins is distinguished from the SIDs for other accounts and groups created in the Reskit domain by its relative identifier, 512. No other account or group in the domain has a SID with a final value of 512.

Relative Identifier Allocation

It is fairly easy for the system to generate a unique relative identifier for each account and group created on a stand-alone computer, where accounts and groups are stored in an account database managed by a local Security Accounts Manager (SAM). The SAM on a stand-alone computer can simply keep track of relative identifier values it has used before, making sure that it never uses them again.

Generating unique relative identifiers is a more complex process in a network domain. Windows 2000 network domains can have several domain controllers, each of them a host for Active Directory, where account information is stored. This means that in a network domain there are as many copies of the account database as there are domain controllers. What is more, every copy of the account database is a master copy. New accounts and groups can be created on any domain controller. Changes made to Active Directory on one domain controller are replicated to all other domain controllers in the domain. The process of replicating changes in one master copy of the account database to all other master copies is called a *multimaster operation*.

The process of generating unique relative identifiers is a *single-master operation.* One domain controller is assigned the role of *relative identifier (RID) master*, and it allocates a sequence of relative identifiers to each domain controller in the domain. When a new domain account or group is created in one domain controller's replica of Active Directory, it is assigned a SID, and the relative identifier for the new SID is taken from the domain controller's allocation of relative identifiers. When its supply of relative identifiers begins to run low, the domain controller asks the RID master for another block.

Each domain controller makes sure that when it has used one value in a block of relative identifiers, it never uses that value again. The RID master makes sure that when it has allocated a block of relative identifiers, it never allocates those values again. The result of this teamwork is that every account and group created in the domain has a unique relative identifier.

Several other tasks performed by domain controllers are single-master operations. For example, one domain controller in an enterprise is assigned responsibility for ensuring that each domain has a unique name and a unique domain identifier. The domain controller assigned that role is called the *domain naming master.* For more information about single-master operations, see "Managing Flexible Single Master Operations" in this book.

SID vs. GUID

When a new domain user or group account is created, Active Directory stores the account's SID in the Object-SID (objectSID) property of a User or Group object. It also assigns the new object a globally unique identifier (GUID), which is a 128-bit value that is unique not only in the enterprise but also across the world. GUIDs are assigned to every object created by Active Directory, not just User and Group objects. Each object's GUID is stored in its Object-GUID (objectGUID) property.

Active Directory uses GUIDs internally to identify objects. For example, the GUID is one of an object's properties that is published in the global catalog. Searching the global catalog for a User object's GUID will yield results if the user has an account somewhere in the enterprise. In fact, searching for any object by Object-GUID might be the most reliable way of finding the object you want to find. The values of other object properties can change, but the Object-GUID never changes. When an object is assigned a GUID, it keeps that value for life.

However, SIDs can sometimes change. The SID for a Group object won't change. Groups stay in the domain where they were created. But people move and when they do, their accounts can move with them. If Alice moves from North America to Europe, but stays in the same company, her account can be transferred with her. An administrator for the enterprise can simply move her User object from, say, Reskit\Noam to Reskit\Euro. If he does, the User object for Alice's account needs a new SID. The domain identifier portion of a SID issued in Noam is unique to Noam, so the SID for Alice's account in Euro has a different domain identifier. The relative identifier portion of a SID is unique relative to the domain, so if the domain changes, the relative identifier also changes.

Thus when a User object moves from one domain to another, a new SID must be generated for the user account and stored in the Object-SID property. Before the new value is written to the property, the previous value is copied to another property of a User object, SID-History (sIDHistory). This property can hold multiple values. Each time a User object moves to another domain, a new SID is generated and stored in the Object-SID property and another value is added to the list of old SIDs in SID-History. When a user logs on and is successfully authenticated, the domain authentication service queries Active Directory for the all of the SIDs associated with the user—the user's current SID, the user's old SIDs, and the SIDs for the user's groups. All of these SIDs are returned to the authentication client and are included in the user's access token. When the user tries to gain access to a resource, any one of the SIDs in the access token, including one of the SIDs in SID-History, could allow or deny the user access.

The reason for maintaining a SID history is obvious. If you allow or deny users access to a resource by virtue of their jobs, you should allow or deny access to a group, not an individual. This way, when users change jobs or move to other departments, you can easily adjust their access by removing them from certain groups and adding them to others. However, if you allow or deny an individual user access to resources, you probably want that user's access to remain the same no matter how many times the user's account domain changes. The SID-History property makes this possible. When a user changes domains, there is no need to change the access control list (ACL) on any resource. If an ACL has the user's old SID but not the new one, the old SID is still in the user's access token, listed among the SIDs for the user's groups, and the user will be granted or denied access based on the old SID.

The reason for using SIDs at all, and not GUIDs, is for backward compatibility. Windows NT uses SIDs to identify users and groups in ACLs on resources. You shouldn't have to change ACLs on all your resources when you upgrade, simply because someone came up with a better scheme. So, in Windows 2000, ACLs continue to identify users and groups by SID, not GIUD—even ACLs on resources in Active Directory. A user gains access to, say, a Group Policy object, based on one of the user's SIDs, not on the GUID for the User object.

Well-Known SIDs

The values of certain SIDs are constant across all systems. These are called well-known SIDs because they identify generic users or generic groups. For example, well-known SIDs identify the following users and groups:

Everyone (S-1-1-0) The generic group Everyone automatically includes everyone who uses the computer, even anonymous guests. The identifier authority value for this SID is 1 (World Authority). It has only one subauthority value, 0 (Null RID).

Creator Owner (S-1-3-0) The generic user Creator Owner is a placeholder in an inheritable ACE. When the ACE is inherited, the system replaces the SID for Creator Owner with the SID for the object's current owner. The identifier authority value for this SID is 3 (Creator Authority). It has only one subauthority value, 0 (Null RID).

Principal Self (S-1-5-10) The generic user Principal Self is a placeholder in an ACE on a User, Group, or Computer object in Active Directory. When you grant permission to Principal Self, you grant it to the security principal represented by the object. During an access check, the operating system replaces the SID for Principal Self with the SID for the security principal represented by the object. The identifier authority for this SID is 5 (NT Authority). It has only one subauthority value, 10 (Self RID).

There are many other well-known SIDs. To see a list of them, see the appendix "Well-Known Security Identifiers" in this book.

Access Tokens

An access token is a protected object that contains information about the identity and privileges associated with a user account. When a user logs on interactively or tries to make a network connection to a computer running Windows 2000, the logon process authenticates the user's logon credentials. If authentication is successful, the logon process returns a SID for the user and a list of SIDs for the user's security groups. The Local Security Authority (LSA) on the computer uses this information to create an access token, which includes the SIDs returned by the logon process as well as a list of privileges assigned by local security policy to the user and to the user's security groups. A copy of the access token is attached to every process and thread that executes on the user's behalf. Whenever a thread interacts with a securable object or tries to perform a system task that requires privileges, the operating system checks the access token associated with the thread to determine its level of authorization.

Contents of an Access Token

An access token contains a complete description of the security context for a process or thread, including the following information:

User The SID for the user's account. If the user logs on to an account on the local computer, the user's SID is taken from the account database maintained by the local SAM. If the user logs on to a domain account, the SID is taken from the Object-SID property of the User object in Active Directory.

Groups A list of SIDs for security groups that include the user. The list also includes SIDs from the SID-History property of the User object representing the user's account in Active Directory.

Privileges A list of privileges held on the local computer by the user and by the user's security groups.

Owner The SID for the user or security group who, by default, becomes the owner of any object that the user either creates or takes ownership of.

Primary Group The SID for the user's primary security group. This information is used only by the POSIX subsystem and is ignored by the rest of Windows 2000.

Default Discretionary Access Control List (DACL) A built-in set of permissions that the operating system applies to objects created by the user if no other access control information is available. The default DACL grants Full Control to Creator Owner and System. For more information about the defaulting of access control information for new objects, see "DACLs for New Objects" later in this chapter.

Source The process that caused the access token to be created, such as Session Manager, LAN Manager, or Remote Procedure Call (RPC) Server.

Type A value indicating whether the access token is a primary or impersonation token. A primary token is an access token that represents the security context of a process. An impersonation token is an access token that a thread within a service process can use to temporarily adopt a different security context, such as the security context for a client of the service.

Impersonation Level A value that indicates to what extent a service can adopt the security context of a client represented by this access token.

Statistics Information about the access token itself. The operating system uses this information internally.

Restricting SIDs An optional list of SIDs added to an access token by a process with authority to create a restricted token. Restricting SIDs can limit a thread's access to a level lower than what the user is allowed.

Session ID A value that indicates whether the access token is associated with the Terminal Services client session.

Impersonation

Impersonation is the ability of a thread to execute in a security context that is different from the context of the process that owns the thread. Impersonation is designed to meet the security requirements of client/server applications. When running in a client's security context, a service "is" the client, to some degree. One of the service's threads uses an access token representing the client's credentials to obtain access to the objects to which the client has access.

The primary reason for impersonation is to cause access checks to be performed against the client's identity. Using the client's identity for access checks can cause access to be either restricted or expanded, depending on what the client has permission to do. For example, suppose a file server has files containing confidential information, and each of these files is protected by a DACL. To prevent a client from obtaining unauthorized access to information in these files, the service can impersonate the client before accessing the files.

Every process has a *primary token* that describes the security context of the user account associated with the process. The user associated with the process for an ordinary application is the human user who started the application. This is not the case for a service process. Services run under their own accounts and act as users in their own right. System services installed with the operating system run under the Local System account. Other services can be configured to run under this account, or they can be given separate accounts on the local system or in Active Directory. For information about installing and configuring domain services, see "Service Publication in Active Directory" in this book.

The primary token for a service is associated with the control thread in the service process and with all other threads acting on behalf of the service. It identifies the service's account, that account's groups, and its privileges. This information is used during access checks when the service requests access to objects needed for its own housekeeping.

When a service accepts a client, it creates a thread to do the work and associates the client's access token with the worker thread. The client's access token is known as an *impersonation token.* The impersonation token identifies the client, the client's groups, and the client's privileges. This information is used during access checks when the thread requests access to resources on the client's behalf. When impersonation is over, the thread reverts to using the primary token and returns to operating in the service's own security context rather than the client's.

The mechanism for impersonation is illustrated in Figure 12.8.

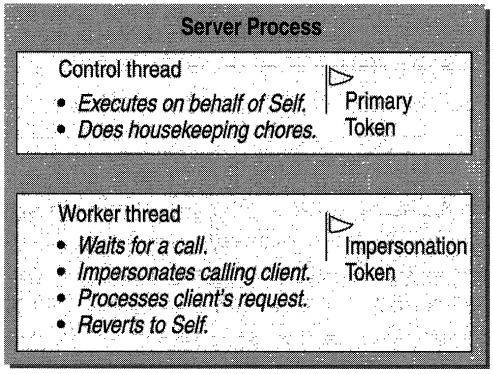


Figure 12.8 The Roles of a Service's Primary and Impersonation Tokens

Impersonation Levels

If impersonation succeeds, it means that the client has agreed to let the server "be" the client to some degree. A client process can control to what extent a service is able to act as the client by selecting an impersonation level when it connects to the service. By choosing an impersonation level, the client tells the service how far it can go in impersonating the client.

An impersonation level cannot be selected by users. It is specified as Security Quality of Service (SQoS) information in the code for client/server applications. There are four levels: anonymous, identify, impersonate, and delegate. The anonymous level has never been supported. Prior to Windows 2000, the only levels supported were identify and impersonate. Windows 2000 adds support for delegate.

The following is a brief description of each level:

Anonymous The client is anonymous to the service. The service can impersonate the client but the impersonation token does not contain any information about the client.

Identify The service can get the identity of the client and use this information in its own security mechanism, but it cannot impersonate the client.

Impersonate The service can impersonate the client. If the service is on the same computer as the client process, it can access network resources as the client. If the service is on a remote computer, it can impersonate the client only when accessing resources on the service's computer.

Delegate The service can impersonate the client not only when it accesses resources on the service's computer but also when it accesses resources on other computers. This level is supported only in Windows 2000 and later versions of the operating system.

Configuring Clients and Services for Delegation

Impersonation works at the delegate level only under the following conditions:

- The computers hosting the client process, the service process, and processes for any backend services must all be running Windows 2000 in a Windows 2000 domain. This is because delegation requires the Kerberos authentication protocol.
- The client's user account must be enabled for delegation.
- The service's account must be enabled for delegation.

To configure a user account for delegation, right-click the User object in Active Directory Users and Computers, click **Properties,** and then click the **Account** tab. In the **Account options** box, look for the **Account is sensitive and cannot be delegated** option. Make sure this option is not selected. Figure 12.9 shows the correct setting.

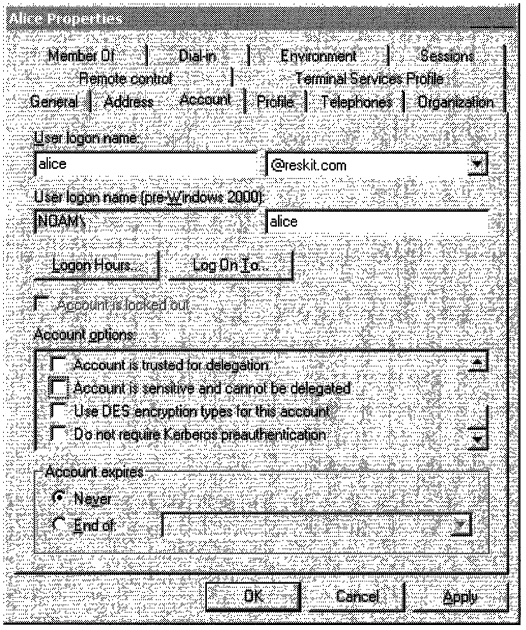


Figure 12.9 User Authentication Can Be Delegated

The way you configure the service account depends on whether the service runs under the Local System account on a computer or under its own domain user account. If the service is configured to run under the Local System account, the computer where the service runs must be trusted for delegation. To configure a computer account as trusted for delegation, you must have the **Enable computer and user accounts to be trusted for delegation** privilege on the computer. If you have this privilege, right-click the **Computer** object in Active Directory Users and Computers, click **Properties,** and then click the **General** tab. Check the box marked **Trust computer for delegation**. Figure 12.10 shows the correct setting.

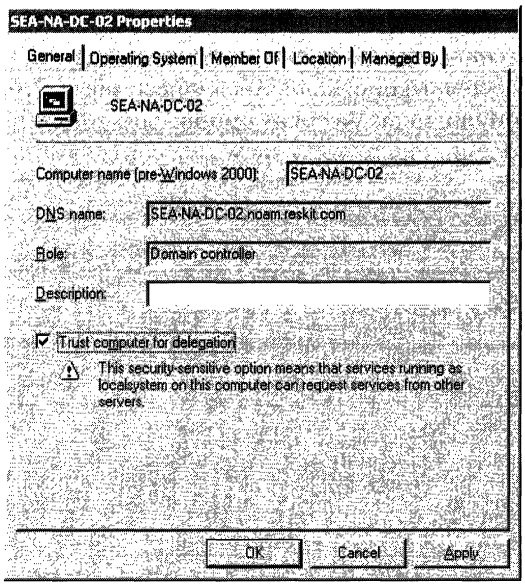


Figure 12.10 Computer Account Trusted to Act as a Delegate

Warning When you trust a computer for delegation, you enable delegation for all services that run under the Local System account on the computer. If an unwary administrator installs an untrusted service on the computer and configures it to run as Local System, it too can access network resources while impersonating other users. A better practice is to configure services that use delegation to run under their own domain user accounts managed by domain administrators.

If the service is configured to run under its own domain user account, the service's user account must be enabled to act as a delegate. To configure a service's user account, right-click the User object, select **Properties,** and then click the **Account** tab. In the **Account options** box, select the **Account is trusted for delegation** option. Figure 12.11 shows the correct setting.

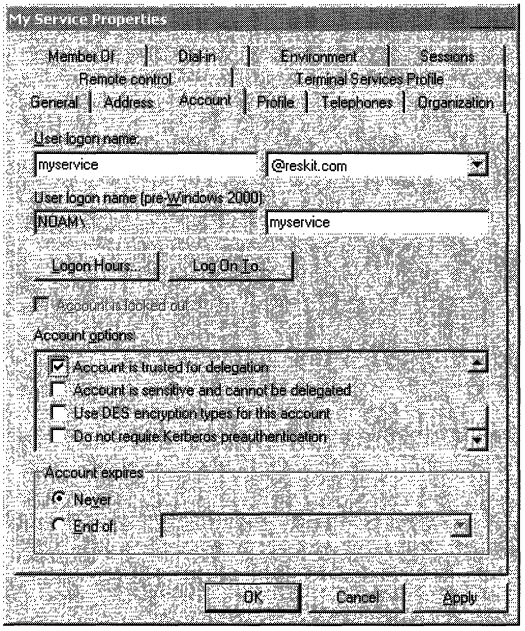


Figure 12.11 Service User Account Trusted to Act as a Delegate

SID Attributes in an Access Token

Each user and group SID in an access token can have one of two attributes that control how the system uses the SID in an access check. These attributes mark a SID either as one to be checked in all ACEs or as a SID to be checked only in ACEs that deny access. Table 12.4 lists the SID attributes.

Table 12.4 SID Attributes

Attribute	Description
SE_GROUP_ENABLED	A SID with this attribute is enabled for access checks. When the system performs an access check, it checks for ACEs that apply to the SID.
SE_GROUP_USE_FOR_DENY_ONLY	**Windows 2000 only**: A SID with this attribute is a deny-only SID. When the system performs an access check, it checks for ACEs that deny access to the SID. It ignores ACEs that allow access for the SID.

Both attributes are mutually exclusive. If one attribute is set, the other cannot be set. If neither attribute is set, the SID is ignored. Moreover, no process is ever allowed to remove a deny-only attribute from a SID.

Restricted Tokens

In Windows 2000, an application can start a child process in a restricted security context so that code executing in the child process has more limited access to resources than the application's user has. For example, when you use Internet-browsing software to display a Web page in an untrusted security zone, code associated with the Web page can execute with fewer privileges on your computer than you have. (This feature is not used by Microsoft Internet Explorer version 5.0 and earlier.) And when you get an email message with an attachment, double-clicking the attachment can invoke an application that similarly runs with limited access to other resources on the computer. (Microsoft Outlook 2000 does not currently use this feature.)

Applications can create restricted security contexts for child processes and impersonation threads by creating a restricted token and assigning it to the child process or impersonating thread. Restricted tokens are created by removing privileges, by applying the deny-only attribute to SIDs, or by adding a list of restricting SIDs to the original access token.

When a restricted process or thread tries to access a securable object, the system performs two access checks: one using the token's enabled and deny-only SIDs, and another using the list of restricting SIDs. Access is granted only if both access checks allow the requested access rights.

For information about how to write applications that use restricted tokens, see the Microsoft Platform Software Development Kit (SDK) link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Security Descriptors

Access control information associated with an object is contained in the object's *security descriptor*. When a user tries to do anything that can be done with the object, the operating system examines the object's security descriptor to determine whether the user is allowed to do what the user wants to do.

Exactly what information is included in a security descriptor depends on the type of object and how it was created. In general, security descriptors can include information about:

- Which user owns the object
- Which users and groups are allowed or denied access
- Which users' and groups' access should be audited
- How objects in a container inherit access control information from the container

Parts of a Security Descriptor

A security descriptor is a binary data structure of variable length. The data structure has the following parts:

Header The header contains a revision number and a set of control flags that describe characteristics of the security descriptor, such as the memory layout, which elements are present, and how particular elements were added or modified.

Owner The Owner field contains the SID for the object's owner. The owner of an object can modify permissions and give other users the right to take ownership.

Primary Group The Primary Group field contains the SID for the owner's primary group. This information is used only by the POSIX subsystem and is ignored by the rest of Windows 2000.

Discretionary Access Control List (DACL) The DACL is a list of zero or more access control entries (ACEs). Each ACE in the list has a header that specifies whether the ACE allows or denies access; a SID that specifies a particular user or group; and an access mask that lists the operations allowed or denied. The DACL's content is controlled by the object's owner. The owner can share control with other users by granting them Change Permissions (WRITE_DAC) permission.

System Access Control List (SACL) The SACL is similar to the DACL except that it is used to audit rather than control access to an object. When an audited action occurs, the operating system records the event in the security log. Each ACE in a SACL has a header that indicates whether auditing is triggered by success or failure or both; a SID that specifies a particular user or security group to monitor; and an access mask that lists the operations to audit. The SACL's content is controlled by security administrators for the local system. Security administrators are users who have been assigned the **Manage auditing and security log** (SeSecurityPrivilege) privilege. By default, this privilege is assigned to the built-in Administrators group.

Layout in Memory

A security descriptor's layout in memory can be either self-relative or absolute. Which of the two formats a particular security descriptor uses is indicated by a control flag in the security descriptor's header.

A self-relative security descriptor is stored in a contiguous block of memory, and the address for each part of the descriptor is expressed as an offset from the beginning of the memory block. To locate any part of the security descriptor, a process must be given only the address of the area in memory where the security descriptor begins. The process can figure out for itself where the part in question begins by adding the memory offset for the part to the memory address for the security descriptor. (Hence the name self-relative. The address for a part is relative to the address for the whole.) Figure 12.12 illustrates the layout of a security descriptor in self-relative format.

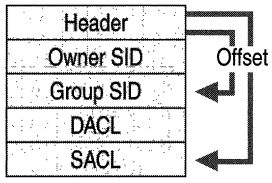


Figure 12.12 Security Descriptor in Self-Relative Format

Self-relative layout is used for security descriptors on objects that must be stored on disk, transmitted by a communications protocol, or copied in memory.

A security descriptor in absolute format contains pointers to its information, not the information itself. This format allows each component of the security descriptor to be allocated separately and assembled on the fly. It is used for objects that do not require persistent storage, such as objects that exist only during the lifetime of a single process. Figure 12.13 illustrates the layout of a security descriptor in absolute format.

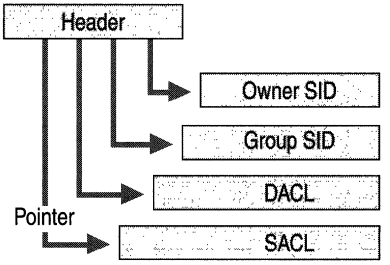


Figure 12.13 Security Descriptor in Absolute Format

Security Descriptor Control Flags

A security descriptor's header contains a set of control flags that qualify the meaning of the security descriptor or its components. In Windows 2000, control flags play an important role in the automatic propagation of inheritable security information from parent (that is, container) objects to child (that is, contained) objects.

Security descriptor control flags are stored in a bit field and are turned on or off by setting individual bits. Table 12.5 lists the security descriptor control flags.

Table 12.5 Security Descriptor Control Flags

Flag	Meaning
SE_DACL_AUTO_INHERITED	**Windows 2000:** Inheritable ACEs in this object's DACL have been propagated to existing child objects. This flag is not set in security descriptors for Windows NT, which does not support automatic propagation of inheritable ACEs.
SE_DACL_DEFAULTED	The DACL was provided by a default mechanism. This flag can affect how the operating system treats the DACL with respect to inheritance. The operating system ignores this flag if SE_DACL_PRESENT is not set.
SE_DACL_PRESENT	The security descriptor has a DACL. **Windows 2000:** If this flag is not set (that is, if the security descriptor has no DACL), SE_DACL_PROTECTEDmust be set. Otherwise, the operating system considers the security descriptor invalid.
SE_DACL_PROTECTED	**Windows 2000:** The security descriptor's DACL cannot be modified by inheritable ACEs. If this flag is not set, the security descriptor inherits information from the security descriptor on the parent object.
SE_GROUP_DEFAULTED	The primary group SID was provided by a default mechanism.

continued

Table 12.5 Security Descriptor Control Flags *(continued)*

Flag	Meaning
SE_OWNER_DEFAULTED	The owner SID was provided by a default mechanism.
SE_SACL_AUTO_INHERITED	**Windows 2000:** Inheritable ACE's in this object's SACL have been propagated to existing child objects. This flag is not set in security descriptors for Windows NT, which does not support automatic propagation of inheritable ACEs.
SE_SACL_DEFAULTED	The SACL was provided by a default mechanism. This flag can affect how the operating system treats the SACL with respect to inheritance. The operating system ignores this flag if SE_SACL_PRESENT is not set.
SE_SACL_PRESENT	The security descriptor has a SACL.
SE_SACL_PROTECTED	**Windows 2000:** The security descriptor's SACL cannot be modified by inheritable ACEs.
SE_SELF_RELATIVE	The security descriptor is in self-relative format with all information in a contiguous block of memory. If this flag is not set, the security descriptor is in absolute format.

Where Access Control Information Comes From

Access control information is first written to an object's security descriptor when the object is created. This information can later be modified. In both cases, the information that goes into a security descriptor can come from one of the following sources:

- The subject
- The object manager
- The parent object

When a subject creates a new object, it can assign the object a security descriptor. If the subject does not assign a security descriptor, the operating system uses access control information inherited from the parent object to create one. If no information is available to inherit, the operation system uses default access control information provided by the object manager for the particular type of object that the subject wants to create.

After an object is created, information in its security descriptor can be changed by the object's owner or by another user who has the owner's permission. Changes can also come from the parent object when that object's owner modifies its security descriptor. Every time the security descriptor on a container object is changed, the object manager propagates any changes marked as inheritable to all objects in the container.

Subjects

A subject is a thread executing in the security context provided by an access token. A subject's access token contains information that can be used in security descriptors for new objects that the subject creates and existing objects that the subject modifies. This information includes the following fields:

Owner This field of an access token contains the SID that identifies the security principal who, by default, becomes the owner of objects that the subject either creates or takes ownership of. Typically, the default owner is the individual user represented by the access token. If the user is a member of the Administrators group, the Default Owner field contains the SID for the group, not the user. If the user is a member of the Domain Admins group, objects that the user creates or takes ownership of in the domain's Active Directory are owned by the Domain Admins group, not by the user. When the SID in the Default Owner field of the subject's access token is copied to the Owner field of an object's security descriptor, SE_OWNER_DEFAULTED is set in the security descriptor's control flags.

Primary Group This field of an access token contains the SID for the user's primary group. When the SID in the Primary Group field of the subject's access token is copied to the Primary Group field of an object's security descriptor, SE_GROUP_DEFAULTED is set in the security descriptor's control flags.

Default DACL This field of an access token might contain a DACL, or it might be empty. If the subject's access token contains a DACL, it can be used as the DACL in a new object's security descriptor. If the Default DACL field is used in the security descriptor for a new object, SE_DACL_DEFAULTED is set in the security descriptor's control flags.

Object Managers

The object manager can provide default security information for new objects. Objects of different types have different requirements, thus each type has its own object manager. Table 12.6 lists common object types and the object manager for each type.

Table 12.6 Object Managers for Common Object Types

Object Type	Object Manager
Files and folders	NTFS
Shares	Server Service
Active Directory objects	Active Directory
Registry keys	The registry
Services	Service Control Manager
Printers	Print spooler
Terminals, window stations, desktops, and windows	Window Manager

For information about default permissions supplied by each object manager, see Windows 2000 Server Help and search for "Objects and Object Managers."

Parent Objects

Some objects can contain other objects. For example, an NTFS Folder object can contain File objects and other Folder objects. A registry key object can contain subkey objects. An Active Directory Organizational Unit (OU) object can contain other OU objects as well as User objects, Group objects, and Computer objects. Terminal objects contain Window Station objects that contain Desktop objects that contain Window objects. Any object contained by another object is called a *child object.* A child object's container is its *parent object.*

Child objects can inherit access control information from their parent object. For example, suppose the administrator for a server creates a file share with one folder, Public$. The administrator creates this folder so that users can have a place to store information that they want to share. With this purpose in mind, the administrator sets the permissions shown in Figure 12.14.

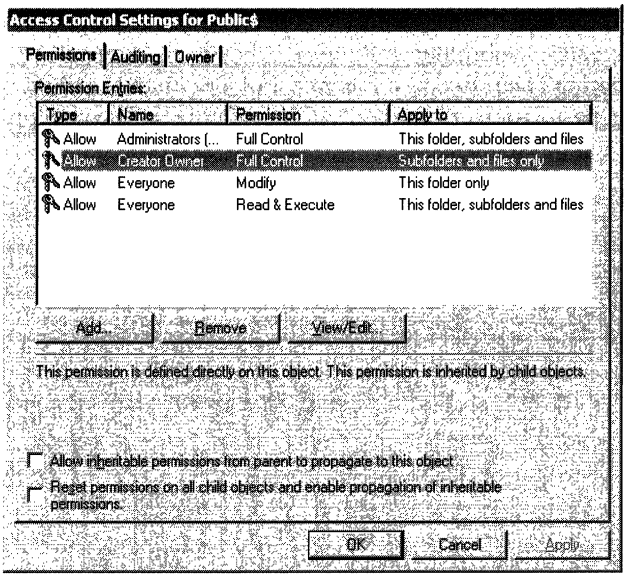


Figure 12.14 Access Control Settings for Public$ (Owner: Administrators)

None of the permissions shown in Figure 12.14 were acquired through inheritance. This is because the administrator cleared the checkbox for **Allow inheritable permissions from parent to propagate to this object**. Clearing the checkbox sets the security descriptor control flag SE_DACL_PROTECTED, which protects a child object's DACL by blocking inheritance from the parent object's DACL.

Permissions acquired through inheritance are called *inherited permissions*. Permissions that are not inherited, but are instead defined directly on an object, are called *explicit permissions*. One way to tell an explicit permission from an inherited permission is to select an entry in the **Permission Entries** list and read the text displayed after the list. In Figure 12.14, the second entry is selected, and the text after the list says **This permission is defined directly on this object**. In other words, the permission is explicit, not inherited.

The text in Figure 12.14 also says **This permission is inherited by child objects.** Permissions on a parent object that apply to child objects are called *inheritable permissions*. To see which of the permissions set on a parent object are inheritable, examine the **Apply to** column of **Permission Entries**. If **Apply to** says **This object only** (or, for folder objects, **This folder only**), then the permission is not inherited by child objects. Of the four permissions shown in Figure 12.14, three are inheritable and one is not.

To see how inheritable permissions become inherited permissions, suppose that Alice creates a subfolder in Public$. Alice is an engineer, so she names her folder Engineering Data. Because this new Folder object is a child of Public$, its DACL inherits permissions from the DACL on Public$. The new object's permissions are shown in Figure 12.15.

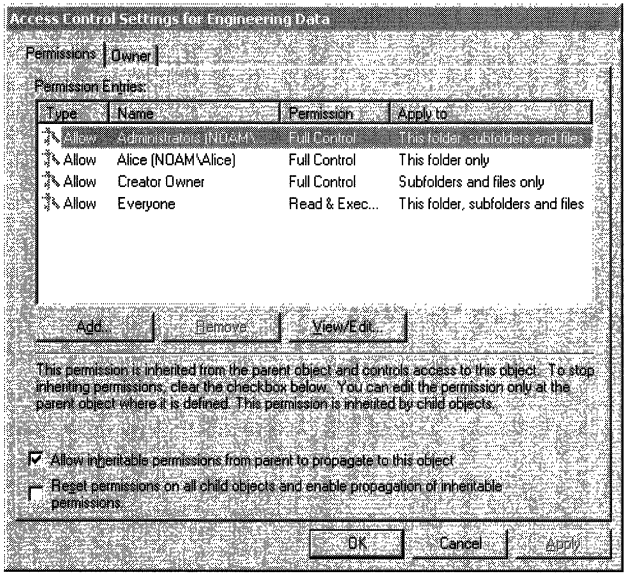


Figure 12.15 Access Control Settings for Engineering Data (Owner: Alice)

Note that Alice has not cleared the checkbox for **Allow inheritable permissions from parent to propagate to this object**, so inheritable permissions in the parent object's DACL are inherited by the child object's DACL. Inherited permissions are indicated in **Permission Entries** by a disabled (grayed-out) symbol at the beginning of each entry. The permission is still effective; all that is disabled is the ability to modify the entry. Because inherited permissions are defined on a parent object, they can be changed only by modifying the parent object's DACL.

Even though inherited permissions cannot be changed, the owner of a child object can add explicit permissions to the object's DACL. For example, suppose Alice decides that inherited permissions given to Creator Owner are too restrictive because they allow only the user who creates a file to make changes to the file. She wants all members of the Engineering group to be able to edit and add information to the Engineering Data folder, so she explicitly gives this group Modify permission for all objects within the folder. And suppose Alice also feels that people in her company's marketing department will misuse information in Engineering Data, so she decides to explicitly deny the Marketing group full control of (and therefore all access to) the folder, subfolders, and files. The result of Alice's changes to the access control settings is shown in Figure 12.16.

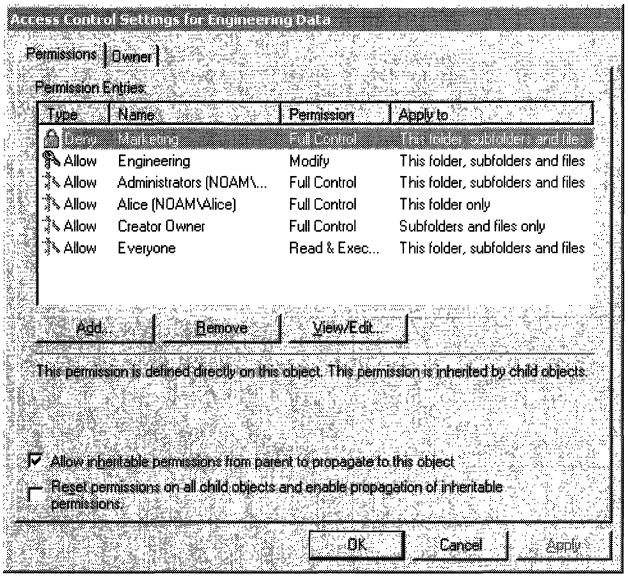


Figure 12.16 Modified Access Control Settings for Engineering Data

The list of permission entries in Figure 12.16 now includes two explicit permissions, both with enabled symbols indicating that the entries can be edited. Note that explicit permissions appear at the top of the list. Permissions are listed in the order in which they will be processed during an access check. Explicit permissions are listed before inherited permissions so that they are processed first. The assumption is that the owner of a child object adds explicit permissions in order to qualify inherited permissions. For example, in Figure 12.16 an inherited permission allows Everyone to read the folder, subfolders, and files. Alice has added an explicit permission that denies all access to a subset of the group Everyone—the Marketing group. The explicit deny entry is placed before any inherited entries, so it is processed before any inherited entries.

Alice applied both of her explicit permissions to **This folder, subfolders, and files** because she wants child objects within her folder to inherit the new permissions. If Engineering Data were on a server running Windows NT, the new explicit permissions would affect only new objects created in her folder. The access control model used in Windows NT supports inheritance only at the time an object is created. To change permissions on existing objects, Alice would need to take the additional step of setting explicit permissions on each existing subfolder and file. What is new in Windows 2000 is inheritance after the time of creation. New or changed inheritable permissions in the DACL on a parent object are automatically propagated to existing child objects every time the DACL on the parent object changes. In the case of Alice's folder, the entry denying Marketing permission to access Engineering Data is propagated to subfolders as soon as Alice clicks **Apply** in the **Access Control Settings** dialog box.

Automatic propagation of inheritable permissions is a powerful capability because it allows you to change permissions on an entire tree of objects by changing permissions on the top-level object in the tree. For example, suppose a member of the Administrators group realizes that information in the Public$ folder and its subfolders is proprietary, not public. In that case, it is inappropriate to allow Everyone access because that group includes all users who can access the network, even anonymous users and guests. The administrator decides that a more prudent choice is Authenticated Users, a group that includes only users who are authenticated by the server. To tighten security on the folder, the administrator simply needs to change Everyone to Authenticated Users for two entries in the permissions list. The change is shown in Figure 12.17.

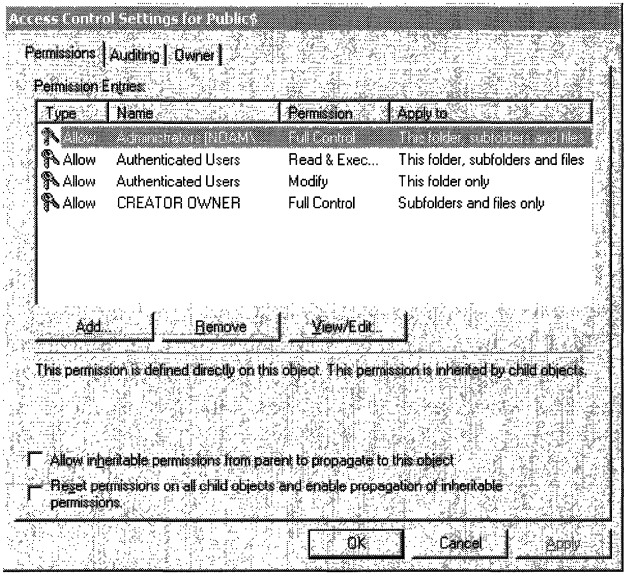


Figure 12.17 Modified Access Control Settings for Public$

When the administrator applies the changes to the access control settings on the parent object, all inheritable permissions in the parent object's DACL are propagated to DACLs on subfolders and files. The impact on Engineering Data is shown in Figure 12.18.

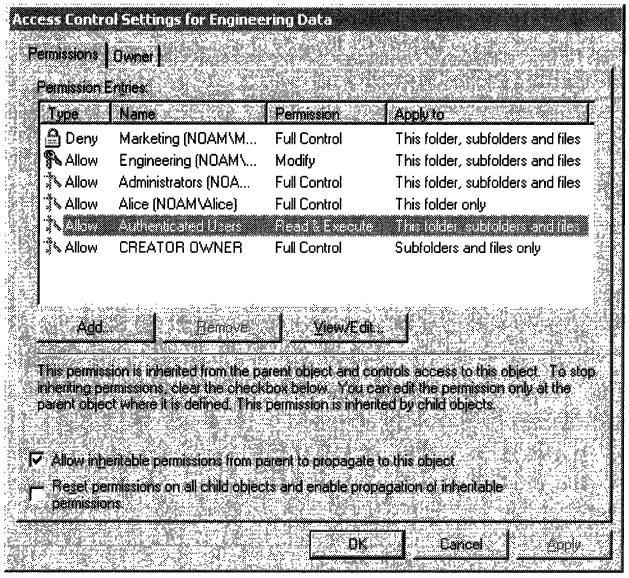


Figure 12.18 New Inherited Permissions for Engineering Data

Note that propagation of inheritable permissions from the Public$ folder to the Engineering Data folder does not change explicit permissions in the child's DACL. Only inherited permissions are replaced when inheritable permissions are propagated to existing child objects. However, if explicit permissions are also inheritable, the propagation process reapplies them as it moves down the tree. For example, both explicit permissions that Alice added to the DACL on her folder are inheritable by child objects in the folder. As the propagation process moves downward from Alice's folder, it picks up these additional inheritable permissions and applies them to the DACL of any child object that it finds.

The owner of a parent object can choose to overwrite explicit permissions defined on child objects. This is done by selecting the **Reset permissions on all child objects and enable propagation of inheritable permissions** checkbox in the Access Control Settings dialog box. When the owner of a parent object chooses this option, the propagation process removes explicit permissions from the DACLs on all child objects. It also sets the option **Allow inheritable permissions from parent to propagate to this object** on all child objects, removing any protection from inheritance that might have been set by the objects' owners.

How Owners are Assigned and Changed

Every object, whether it is in Active Directory or in an NTFS volume, has an owner, usually the user who created the object. The owner has an implied right to allow or deny other users permission to use the object, and this right cannot be withdrawn. Among other permissions, owners can give other users permission to Change Permissions (WRITE_DAC). This permission, unlike the owner's inherent right, can be withdrawn.

By default, a new object's owner is the security principal identified as the default owner in the access token attached to the creating process. When an object is created, the SID stored in the access token's Owner field is copied to the security descriptor's Owner field. The default owner is normally an individual—the user who is currently logged on. The only exceptions occur when the user is a member of either the Administrators group or the Domain Admins group. In both cases, the Owner field in the user's access token contains the SID for the group, not the SID for the individual user account. The assumption is that administrative accounts are used only to administer the system and not for any individual purpose. As a result, objects created by one administrator can be managed by other administrators in the same group.

If an administrative group such as Administrators owns an object, all members of the group share the owner's inherent right to change permissions for the object. This fact frequently surprises administrators. For example, suppose Alice logs on to an account in the Administrators group, creates a file, and then denies Bob permission to modify it. Because Alice is a member of the Administrators group, the group owns the file. If Bob is also a member of the Administrators group, he automatically has Change Permissions authority and can give himself permission to modify the file—despite Alice's effort to prevent him from modifying it.

Owners of NTFS objects can allow another user to take ownership by giving that user Take Ownership permission. Owners of Active Directory objects can do the same thing by giving another user Modify Owner permission. (Both permissions map to the same access right, WRITE_OWNER. The only difference between the two permissions is what they are called in the user interface.) In addition, certain users can take ownership without having permission if they have been assigned the **Take ownership of files or other objects** (SeTakeOwnershipPrivilege) privilege. By default, this privilege is assigned only to the Administrators group.

When a user takes ownership of an object, the default owner SID in the user's access token is copied to the owner field of the object's security descriptor. If a member of the Administrators group takes ownership (or, for Active Directory object, a member of the Domain Admins group), the default owner is the group, not the individual user. For example, Figure 12.19 shows the Owner tab for a Folder object created by Bob, a member of the Administrators group.

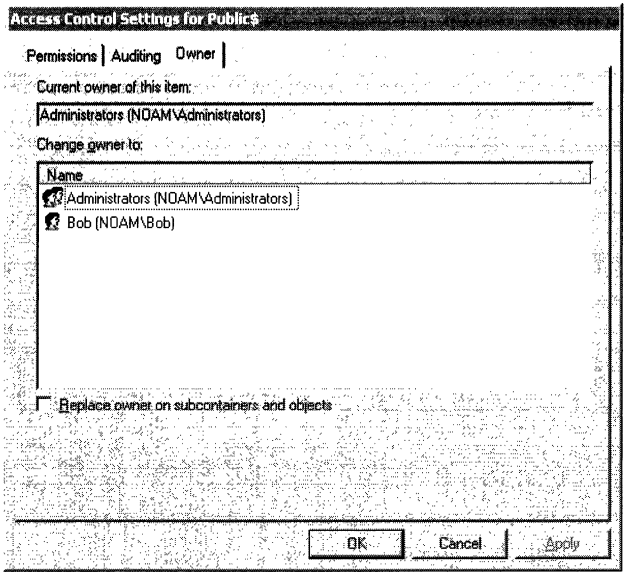


Figure 12.19 Owner Tab for a Folder Object

Even though Bob created the folder, the Owner tab shows the object's current owner as Administrators. When the object was created, the SID in the Default Owner field of Bob's access token was placed in the Owner field of the object's security descriptor. Because Bob is a member of Administrators, his access token specified Administrators as the default owner, so Administrators became the owner of the object. Figure 12.19 also shows that Bob has the option to take ownership of the object himself. He can simply select his name in the **Change owner to** list and then click **Apply**. However, if Bob takes ownership, another member of Administrators can always retake ownership on behalf of the entire group. In fact, members of Administrators can always take ownership of any object, no matter who originally owned the object. This capability is built into the operating system and cannot be removed.

Note that the Owner tab shown in Figure 12.19 also includes a checkbox for **Replace owner on all subcontainers and objects.** If Bob takes ownership of this folder, he can take ownership of all subfolders and files at the same time. Bob has this option because he is a member of Administrators. Ordinary users have the option only if, in addition to having Take Ownership permission for the parent object, they also have Take Ownership permission for all child objects.

The Owner tab shown in Figure 12.19 has no option for giving ownership to another individual. That capability is not exposed in the user interface as a safeguard against unscrupulous users who might take ownership, do something wrong, and then cover their tracks by giving ownership to someone else. However, the capability does exist for programs. If a process has WRITE_OWNER access to an object, it can write new information in the Owner field of the object's security descriptor.

You can track which users take ownership of objects by setting an audit control on take ownership events. For example, Figure 12.20 shows that auditing is enabled for all Take Ownership access to the top-level folder, Public$, and all subfolders and files. Whenever a user or group takes ownership of the parent object or any child object, the event is recorded in the security log.

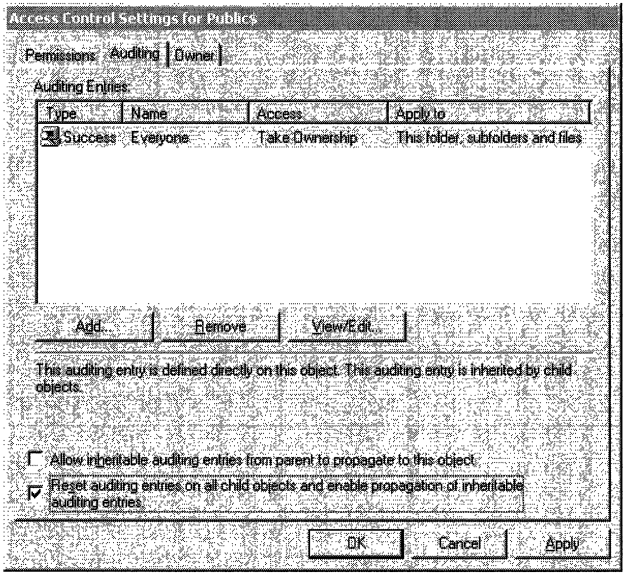


Figure 12.20 Auditing tab for a Folder Object

For information about how to enable auditing, see Windows 2000 Server Help.

Who Owns It?

To find out who owns any object, you must have Read Permissions (READ_CONTROL) access to the object. If you have the necessary authorization, getting the name of an object's owner is fairly simple.

To determine who owns most objects, right-click the object, then click **Properties**. In the Properties dialog box, click the **Security** tab, and then click **Advanced**. In the **Access Control Settings** dialog box, click the **Owner** tab. The name of the object's owner is shown after **Current owner of this item**.

For file and folder objects, you can see the ownership of several objects all at once. Open a command console, change to the directory in question, and then type DIR /Q. The output from this command lists the names of owners in one column, and the names of objects in the next. The name of the owner is omitted for any object for which you do not have Read Permissions access.

To determine who owns a registry object, run Regedt32. Select the object in question, then choose **Owner** from the **Security** menu. The Owner dialog box displays the name of the object and the name of its owner.

How Primary Groups Are Assigned and Changed

For domain accounts, the default primary group is Domain Users. You can change a particular user's primary group by editing the properties of the User object in Active Directory. For more information about changing a user's primary group, see Windows 2000 Server Help.

When a new object is created, the creating process can specify a SID for the object's Primary Group field. If the creating process does not specify a primary group, one is taken from the Default Primary Group field of the subject's access token.

A similar procedure is followed when a user takes ownership of an object. Normally, the thread acting on the user's behalf does not specify a primary group. When it does not, the SID in the Default Primary Group field of the subject's access token is copied to the Primary Group field of the object's security descriptor.

Access Control Lists

An access control list (ACL) is an ordered list of access control entries (ACEs) that define the protections that apply to an object and its properties. Each ACE identifies a security principal and specifies a set of access rights allowed, denied, or audited for that security principal.

An object's security descriptor can contain two ACLs:

- A discretionary access control list (DACL) that identifies the users and groups who are allowed or denied access
- A system access control list (SACL) that controls how access is audited

The data structure for an ACL is illustrated in Figure 12.21.

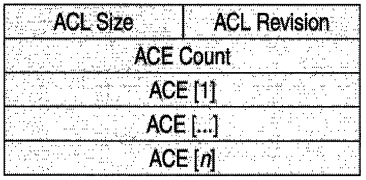


Figure 12.21 Structure of an ACL

The individual parts of an ACL are as follows:

ACL Size The number of bytes of memory allocated for the ACL. The size of an ACL varies with the number and size of its ACEs.

ACL Revision The revision number for the ACL's data structure. The structure of an ACL is the same for all revisions, but the structure of ACEs in the ACL can vary. The revision number for most objects is 2. The revision number for Active Directory objects is 4.

ACE Count The number of ACEs in the ACL. A value of zero means the ACL has no ACEs—it is empty, therefore access-checking can stop.

ACEs An ordered list containing zero or more ACEs. During an access check, ACEs are processed in the order in which they are listed.

Access Control Entries

All ACEs include the following access control information:

- A SID that identifies a user or group
- An access mask that specifies access rights
- A set of bit flags that determine whether child objects can inherit the ACE
- A flag that indicates the type of ACE

ACE Types

Windows 2000 supports six types of ACEs. Three are generic ACE types that can be present in ACLs attached to all securable objects. Table 12.7 lists generic ACE types. The three remaining ACE types are object-specific and can occur only in ACLs for Active Directory objects. Table 12.8 lists object-specific ACE types.

Table 12.7 Generic ACE Types

Type	Description
Access-denied	Used in a DACL to deny access.
Access-allowed	Used in a DACL to allow access.
System-audit	Used in a SACL to log attempts to access.

Table 12.8 Object-Specific ACE Types

Type	Description
Access-denied, object-specific	Used in a DACL to deny access to a property or property set, or to limit inheritance to a specified type of child object.
Access-allowed, object-specific	Used in a DACL to allow access to a property or property set, or to limit inheritance to a specified type of child object.
System-audit, object-specific	Used in a SACL to log attempts to access a property or property set, or to limit inheritance to a specified type of child object

Generic and object-specific ACEs are fundamentally alike. What sets them apart is the granularity of control they offer over inheritance and object access.

Generic ACEs offer limited control over the kinds of child objects that can inherit them. Essentially, they can distinguish only between containers and noncontainers. For example, the DACL on a Folder object in the NTFS file system can include a generic ACE that allows a group of users to list the folder's contents. This is an operation that can be performed only on container objects, so the ACE that allows the operation can be flagged as a CONTAINER_INHERIT _ACE. Only container objects in the folder (that is, only other Folder objects) inherit the ACE. Noncontainer objects (that is, File objects) do not.

Object-specific ACEs offer greater granularity of control over the types of child objects that can inherit them. For example, an OU object's ACL can have an object-specific ACE that is marked for inheritance only by User objects. Other types of objects, such as Computer objects, will not inherit the ACE. This capability is why object-specific ACEs are called object-specific. Their inheritance can be limited to specific types of child objects.

There are similar differences in how the two categories of ACE types control access to objects. Generic ACEs apply to an entire object. If a generic ACE gives a particular user read access, the user can read all information associated with the object—both data and properties. This is not a serious limitation for most object types. File objects, for example, have few properties, all used for describing characteristics of the object rather than for storing information. Most of the information in a File object is stored as object data, so there is little need for separate controls on a file's properties.

Object-specific ACEs can apply to any individual property of an object or to a set of properties. These ACE types are used only in ACLs for Active Directory objects, which, unlike other object types, store most of their information in properties. It is often desirable to place independent controls on each property of an Active Directory object, and object-specific ACEs make that possible. For example, when you define permissions for a User object, you can use one object-specific ACE to allow Principal Self (that is, the user) write access to the Phone-Home-Primary (homePhone) property, and you can use other object-specific ACEs to deny Principal Self access to the Logon-Hours (logonHours) property and other properties that set restrictions on the user account.

Structure of a Generic ACE

All three generic ACE types have the same data structure, which Figure 12.22 illustrates.

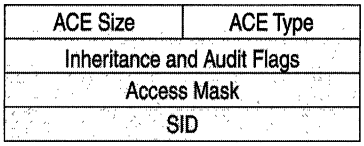


Figure 12.22 Structure of a Generic ACE

The individual parts of an ACE are as follows:

ACE Size The number of bytes of memory allocated for the ACE.

ACE Type Specifies whether the ACE allows, denies, or monitors access.

Inheritance/Audit Flags A set of bit flags that control inheritance and auditing. For information about inheritance flags, see "Inheritance" later in this chapter. Table 12.9 describes audit flags.

Table 12.9 Audit Flags

Flag	Meaning
FAILED_ACCESS_ACE_FLAG	Meaningful only in system-audit and system-audit object ACEs. The access mask specifies operations that should be logged when they fail.
SUCCESSFUL_ACCESS_ACE_FLAG	Meaningful only in system-audit and system-audit object ACEs. The access mask specifies operations that should be logged when they succeed.

Access Mask A 32-bit value whose bits correspond to access rights for the object. Bits can be set either on or off, but the setting's meaning depends on the ACE type. For example, if the bit that corresponds to the right to read permissions is turned on, and the ACE type is Deny, then the ACE denies the right to read the object's permissions. If the same bit is set on but the ACE type is Allow, then the ACE grants the right to read the object's permissions.

SID Identifies a user or group whose access is controlled or monitored by this ACE.

Structure of an Object-Specific ACE

Figure 12.23 illustrates the structure of an object-specific ACE.

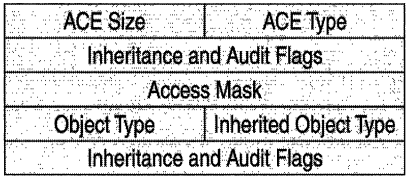


Figure 12.23 Structure of an Object-Specific ACE

The fields for ACE Size, ACE Type, Inheritance/Audit Flags, Access Mask, and SID are identical to like elements in the data structure for a generic ACE. The key differences between a generic and an object-specific ACE are as follows:

Object Flags

Object flags indicate whether Object Type or Inherited Object Type are present. Table 12.10 shows the three flags that are possible.

Table 12.10 Object Flags

Flag	Meaning
0 (no flags)	Neither Object Type nor Inherited Object Type are present. In this case, the ACE applies to the entire object. It is, in effect, a generic ACE.
ACE_OBJECT_TYPE_PRESENT	ACE applies to a property, property set, or extended right, or it controls the ability to create a particular type of child object.
ACE_INHERITED_OBJECT _TYPE_PRESENT	ACE can be inherited only by a particular type of child object.

Object Type

Object Type contains a GUID that identifies one of the following:

- **A type of child object**. The ACE controls who can create a particular type of child object within a container. The SID portion of the ACE identifies a user or group who can create this type of child object. The ACE's access mask contains the object-specific access right ADS_RIGHT_DS_CREATE_CHILD.
- **A property or property set**. The ACE controls the ability to read or write a particular property or property set. The SID portion of the ACE identifies a user or group who can read or write the property or property set. The ACE's access mask contains either ADS_RIGHT_DS_READ_PROP or ADS_RIGHT_DS_WRITE_PROP.
- **An extended right.** The ACE controls the right to perform the operation associated with the extended right. The SID portion of the ACE identifies a user or group who has the extended right. The ACE's access mask contains ADS_RIGHT_DS_CONTROL_ACCESS.

Inherited Object Type

Inherited Object Type contains a GUID that identifies the type of child object that can inherit the ACE. Inheritance is also controlled by the ACE's Inheritance Flags and by any protection against inheritance placed on the child object in its Security Descriptor Control Flags.

DACLs for Newly Created Objects

The operating system uses the following rules to set the DACL in the security descriptors for most types of new securable objects:

1. The object's DACL is the DACL from the security descriptor specified by the the creating process. The operating system merges any inheritable ACEs into the DACL unless SE_DACL_PROTECTED is set in the security descriptor control flags. The operating system then sets the SE_DACL_PRESENT security descriptor control flag.
2. If the creating process does not specify a security descriptor, the operating system builds the object's DACL from inheritable ACEs in the parent object's DACL. The operating system then sets the SE_DACL_PRESENT security descriptor control flag.
3. If the parent object has no inheritable ACEs, the operating system asks the object manager to provide a default DACL. The operating system then sets the SE_DACL_PRESENT and SE_DACL_DEFAULTED security descriptor control flags.

4. If the object manager does not provide a default DACL, the operating system checks the subject's access token for a default DACL. The operating system then sets the SE_DACL_PRESENT and SE_DACL_DEFAULTED security descriptor control flags.
5. If the subject's access token does not have a default DACL, the new object is assigned no DACL, which allows unconditional access by everyone. The SE_DACL_PRESENT security descriptor control flag is not set.

DACLs for Newly Created Active Directory Objects

The method used to build a DACL for a new Active Directory object is slightly different from the method used to build DACLs for other object types. There are two key differences, which are as follows:

- The rules for creating a DACL distinguish between generic inheritable ACEs and object-specific inheritable ACEs in the parent object's security descriptor. Generic inheritable ACEs can be inherited by all types of child objects. Object-specific inheritable ACEs can be inherited only by the type of child object to which they apply.
- The Active Directory schema can provide a security descriptor. Each object class defined in the schema has a defaultSecurityDescriptor attribute. If neither the creating process nor inheritance from the parent object provides a DACL for a new Active Directory object, the operating system uses the DACL in the default security descriptor specified by the schema.

The operating system uses the following rules to set the DACL in the security descriptor for new Active Directory objects:

1. The object's DACL is the DACL from the security descriptor specified by the the creating process. The operating system merges any inheritable ACEs into the DACL unless SE_DACL_PROTECTED is set in the security descriptor control flags. The operating system then sets the SE_DACL_PRESENT security descriptor control flag.
2. If the creating process does not specify a security descriptor, the operating system checks the parent object's DACL for inheritable object-specific ACEs that apply to the type of object being created. If the parent object has inheritable object-specific ACEs for the object type, the operating system builds the object's DACL from inheritable ACEs, including both generic and object-specific ACEs. It then sets the SE_DACL_PRESENT security descriptor control flag.
3. If the parent object has no inheritable object-specific ACEs for the type of object being created, the operating system uses the default DACL from the Active Directory schema for that object type. It then sets the SE_DACL_PRESENT and SE_DACL_DEFAULTED security descriptor control flags.

4. If the Active Directory schema does not specify a default DACL for the object type, the operating system checks the subject's access token for a default DACL. It then sets the SE_DACL_PRESENT and SE_DACL_DEFAULTED security descriptor control flags.
5. If the subject's access token does not have a default DACL, the new object is assigned no DACL, which gives unconditional access to everyone. The SE_DACL_PRESENT security descriptor control flag is not set.

SACLs for Newly Created Objects

The operating system uses the following rules to set the SACL in the security descriptors for new securable objects:

1. If the creating process provides an explicit SACL, the operating system places it in the object's security descriptor. The operating system merges any inheritable ACEs into the SACL unless SE_SACL_PROTECTED is set in the security descriptor control flags. It then sets the SE_SACL_PRESENT security descriptor control flag.
2. If the creating process does not provide an explicit SACL, the operating system builds the object's SACL from inheritable ACEs in the parent object's SACL. It then sets the SE_SACL_PRESENT security descriptor control flag.
3. If the parent object has no inheritable ACEs, the operating system asks the object manager to provide a default SACL. It then sets the SE_SACL_PRESENT and SE_SACL_DEFAULTED security descriptor control flags.
4. If the object manager does not provide a default SACL, the new object is assigned no SACL.

Inheritance

Inheritance is the process that propagates ACEs in a parent object's ACL to a child object's ACL. In Windows 2000, inheritable ACEs can be propagated from parent to child when one of the following events takes place:

- A new child object is created
- The DACL on the parent object is modified
- The SACL on the parent object is modified

In this scheme, any object can be the child of another object. Only container objects can be parents. And, just as in human genetics a parent can carry recessive traits that are not evident in the parent, so it is in Windows 2000. The ACL for a container object can carry ACEs that are not effective on the container but are present only for the purpose of inheritance—only so that they can be passed down to subsequent generations of objects until they reach a noncontainer child object, where they become effective ACEs.

The mechanism for inheritance depends on two things: a set of inheritance flags in each ACE and a set of inheritance rules built into the operating system.

Inheritance Flags

The header for an ACE contains a set of inheritance flags that control how the ACE is inherited and how the ACE affects a child object that inherits it. Table 12.11 lists the inheritance flags.

Table 12.11 Inheritance Flags

Flag	Meaning
INHERITED_ACE	**Windows 2000**: The ACE was inherited from a parent object's DACL or SACL. This flag is not set in an explicit ACE—an ACE defined directly on the object.
INHERIT_ONLY_ACE	Indicates that this is an inherit-only ACE. The ACE is ignored during an access check but can be propagated to child objects. If this flag is not set, the ACE is an effective ACE—an ACE that is processed during an access check. Both effective and inherit-only ACEs can be inherited. Whether an ACE is inherited depends on the state of the flags for OBJECT_INHERIT_ACE and CONTAINER_INHERIT_ACE.
CONTAINER_INHERIT_ACE	Container objects inherit this ACE as an effective ACE. When the ACE is inherited by a container object, the operating system clears the flag for INHERIT_ONLY_ACE.

continued

Table 12.11 Inheritance Flags *(continued)*

Flag	Meaning
OBJECT_INHERIT_ACE	Non-container objects inherit this ACE as an effective ACE. When the ACE is inherited by a non-container object, the operating system clears the flag for INHERIT_ONLY_ACE. Container objects also inherit this ACE but only for the purpose of inheritance. When the ACE is inherited by a container object, the operating system sets the flag for INHERIT_ONLY_ACE.
NO_PROPAGATE_INHERIT_ACE	If a child object inherits an ACE where this flag is set, the operating system clears the flags for OBJECT_INHERIT_ACE and CONTAINER_INHERIT_ACE. This prevents the ACE from being inherited by subsequent generations of objects.

Rules of Inheritance

The operating system interprets the inheritance flags and other inheritance information according to the rules of ACE inheritance outlined in Table 12.12. These rules function the same for both DACLs and SACLs. When the operating system propagates inheritable ACEs to child objects, it observes the preferred (or *canonical*) order. After ACEs have been propagated, the system sets the INHERITED_ACE flag in all inherited ACEs.

Table 12.12 Rules of Inheritance

Parent ACE Inheritance Flags	Effect on Child ACL
No flags	None.
OBJECT_INHERIT_ACE only	**Noncontainer child objects**: Inherited as an effective ACE. **Container child objects**: Containers inherit an inherit-only ACE unless the NO_PROPAGATE_INHERIT_ACE flag is also set.
CONTAINER_INHERIT_ACE only	**Noncontainer child objects**: No effect on the child object. **Container child objects**: The child object inherits an effective ACE. The inherited ACE is inheritable unless the NO_PROPAGATE_INHERIT_ACE flag is also set.

continued

Table 12.12 Rules of Inheritance *(continued)*

Parent ACE Inheritance Flags	Effect on Child ACL
CONTAINER_INHERIT_ACE and OBJECT_INHERIT_ACE	**Noncontainer child objects**: Inherited as an effective ACE. **Container child objects**: The child object inherits an effective ACE. The inherited ACE is inheritable unless the NO_PROPAGATE_INHERIT_ACE flag is also set.

If an inherited ACE is an effective ACE for the child object, the system maps any generic rights to the specific rights for the child object. Similarly, the system maps generic SIDs, such as CREATOR_OWNER, to the appropriate SID. If an inherited ACE is an inherit-only ACE, any generic rights or generic SIDs are left unchanged so that they can be mapped appropriately when the ACE is inherited by the next generation of child objects.

When a container object inherits an ACE that is both effective on the container and inheritable by its descendants, the container might inherit two ACEs. This occurs if the inheritable ACE contains generic information. The container inherits an inherit-only ACE containing the generic information, and an effective ACE in which the generic information is mapped.

An object-specific ACE has an Inherited Object Type field that can contain a GUID to identify the type of object that can inherit the ACE. If the field does not contain a GUID, the inheritance rules are the same as for a standard ACE. If the field contains a GUID, the ACE is inheritable by objects that match the GUID if OBJECT_INHERIT_ACE is set, and by containers that match the GUID if CONTAINER_INHERIT_ACE is set.

Order of ACEs in a DACL

The preferred order of ACEs in a DACL is called the *canonical* order. For Windows 2000, the canonical order is the following:

- All explicit ACEs are placed in a group before any inherited ACEs.
- Within the group of explicit ACEs, access-denied ACEs are placed before access-allowed ACEs.
- Inherited ACEs are placed in the order in which they are inherited. ACEs inherited from the child object's parent come first, then ACEs inherited from the grandparent, and so on up the tree of objects.

Figure 12.24 shows the canonical order.

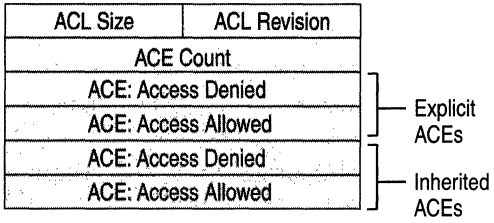


Figure 12.24 Canonical Order of ACEs

The canonical order ensures that an explicit access-denied ACE is enforced regardless of any explicit access-allowed ACE. This means that the object's owner can define permissions that allow access to a group of users, such as Everyone, but deny access to a subset of the group, such as Marketing. If the object's ACEs are in canonical order, the ACE that denies Marketing comes before the ACE that allows Everyone. During an access check, the operating system steps through the ACEs in the order in which they appear in the object's DACL, so the deny ACE is processed before the allow ACE. As a result, users who are members of the Marketing group are denied access. Everyone else is allowed access to the object.

The canonical order also ensures that all explicit ACEs are processed before any inherited ACE. This is consistent with the concept of *discretionary* access control: access to a child object is at the discretion of the child's owner, not the parent's owner. The owner of a child object can define permissions directly on the child that modify the effects of inherited permissions. For example, suppose a parent object has an inheritable ACE that denies access to Marketing. And suppose the owner of a child object defines an explicit ACE that allows access to a subset of Marketing, let's say Bob. If the child object's ACEs are in canonical order, the explicit ACE that allows Bob access comes before any inherited ACE, including the inherited ACE that denies access to Marketing. During an access check, the operating system reaches the ACE that allows Bob access before it gets to the ACE that denies access to Marketing. As a result, Bob is allowed access to the object even though he is a member of the Marketing group. Other members of the group are denied access.

What Happens When You Upgrade

When you upgrade a computer from Windows NT 4.0 to Windows 2000, the new canonical order is applied to DACLs on existing objects the first time after the upgrade that their security descriptors are modified. Because Windows 2000 introduces automatic propagation of ACEs to existing objects, the DACLs on all objects in the hierarchy below the modified object are also converted to the new canonical order. The security descriptors for objects that have been converted contain the security descriptor control flags SE_DACL_AUTO_INHERITED and SE_SACL_AUTO_INHERITED.

The operating system uses the following rules for propagating inherited ACEs to child objects.

- If a child object with no DACL inherits an ACE, the result is a child object with a DACL containing only the inherited ACE.
- If a child object with an empty DACL inherits an ACE, the result is a child object with a DACL containing only the inherited ACE.
- If you remove an inheritable ACE from a parent object, automatic inheritance removes any copies of the ACE inherited by child objects.
- If automatic inheritance results in the removal of all ACEs from a child object's DACL, the child object has an empty DACL rather than no DACL.

No DACL vs Empty DACL

A security descriptor with *no DACL* gives unconditional access to everyone. A security descriptor with an empty DACL gives no access to anyone.

In Windows NT, if you were a developer who wanted to allow everyone unconditional access to an object, your code could create the object without a DACL. In Windows 2000, developers can still create objects with no DACL, but code that creates an object with no DACL must now set the SE_DACL_PROTECTED security descriptor control flag. If that flag is not set, the object will acquire a DACL through inheritance, and the result is likely to be something less than unconditional access for everyone. In fact, if the parent object has no inheritable permissions, the child could end up with an empty DACL—which would allow no access to anyone, exactly the opposite of what the object's creator intended.

If you are developing for Windows 2000 and want to give everyone unconditional access to an object that your code creates, then assign the object a DACL with one ACE that grants Everyone full control.

What Happens When You Convert FAT to NTFS

When you convert a FAT volume to the NTFS file system, Windows 2000 sets permissions on all folders and files in the converted volume. The permissions set on existing folders are shown in Table 12.13. The permissions set on existing files are shown in Table 12.14. All permissions are set in explicit ACEs.

Table 12.13 DACL on Existing Folders After Converting from FAT to NTFS

Type	Name	Permission	Apply to
Allow	System	Full Control	This folder only
Allow	Administrators	Full Control	This folder only
Allow	Everyone	Full Control	This folder, subfolders and files

Table 12.14 DACL on Existing Files After Converting from FAT to NTFS

Type	Name	Permission
Allow	System	Full Control
Allow	Administrators	Full Control
Allow	Everyone	Full Control

Only the last ACE in a folder's DACL is inheritable, so this is the only ACE that is inherited by new objects that are created in the converted volume. Table 12.15 shows the DACL for files and folders created after the conversion.

Table 12.15 DACL on New Files and Folders After the Conversion

Type	Name	Permission
Allow	Everyone	Full Control

This DACL would not be a problem on computers running Windows NT. However, it does pose a problem on computers running Windows 2000. For example, suppose you edit permissions on the converted volume's root, and you remove the inheritable permission that allows Everyone full control of the folder, subfolders, and files. In Windows 2000, changes to inheritable ACEs are propagated down the tree of objects; changes to inheritable ACEs on the root object in a tree are propagated to the entire tree. Objects that existed before the conversion have explicit ACEs allowing Everyone full control, not inheritable ACEs, so they would not be affected by the change in permissions on the root object. But files and folders created after the conversion have an inherited ACE allowing Everyone full control. This ACE would be removed from all files and folders created after the conversion, and each of these objects would end up with an empty DACL, which allows no access to anyone.

You could recover from this error by adding a new inheritable ACE to the root object and propagating that change down the tree of objects. However, it is always better to preclude the possibility of error rather than try to recover from one. In this case, you can avoid trouble by changing the non-inheritable ACEs on the root object to inheritable ACEs. It is recommended that you do this immediately after converting a volume from FAT to NTFS.

▶ **To avoid creating empty DACLs on a converted volume**

1. Open Windows Explorer, and then locate the drive symbol for the converted volume.
2. Right-click the drive symbol, click **Properties**, click the **Security** tab, and then click **Advanced**.
3. In **Permissions**, double-click Administrators.
4. In **Apply onto**, select "This folder, subfolders and files."
5. Click **OK**.
6. In **Permissions**, double-click System.
7. In **Apply onto**, select "This folder, subfolders and files."

Warning Do not check **Reset permissions on all child objects and enable propagation of inheritable permissions**. Propagation of inheritable permissions is already enabled by default. Checking this option clears the SE_DACL_PROTECTED security descriptor control flag in every child object's security descriptor and removes permissions that are explicitly defined on the objects. Neither action is necessary for this procedure.

8. Click **OK** until all dialog boxes are closed.

After you make this change, the new inheritable ACEs are propagated to child objects below the root folder, and they are inherited by any new subfolders and files that are created in the tree.

Access Check and Audit Generation

When a subject attempts to access an object, the object manager calls the function AccessCheckAndAuditAlarm to determine if access is allowed, denied, or audited. The function approaches the job in two phases. First, it determines whether the subject is allowed or denied access. Then it determines whether it needs to generate an auditing entry in the security log.

Access-Checking

The goal of an access check is to determine whether the subject is authorized to do what the subject wants to do. This determination is made by the function AccessCheckAndAuditAlarm, which considers the following information:

- The subject's access token
- The subject's desired access mask
- The object's security descriptor

The subject's desired access mask is a data structure 32 bits long, each bit corresponding to a particular access right. Bits are turned on for rights that the subject wants, and they are turned off for rights that the subject does not want. For more information about access masks, see "Access Masks" earlier in this chapter.

When access-checking is complete, AccessCheckAndAuditAlarm returns a granted access mask to the object manager. The granted access mask is a 32-bit structure identical to the desired access mask, except that all bits are initially turned off. As the access-checking process finds authorization for each right requested in the desired access mask, the bit corresponding to that right is turned on in the granted access mask and turned off in the desired access mask. When all bits in the desired access mask are turned off, access-checking stops. The granted access mask is then returned to the calling process.

AccessCheckAndAuditAlarm uses the following rules to determine the subject's authorization:

1. If the object's security descriptor has no DACL, then the granted access mask is set to match the desired access mask, and access-checking stops. The subject receives the access that it requests.
2. If the desired access mask is empty, access-checking stops. The subject gains no access to the object.
3. If the bit for the right to access the SACL is turned on in the desired access mask, the subject's access token is checked for the **Manage auditing and security log** (SeSecurityPrivilege) privilege. If the access token contains this privilege, the bit for SACL access is turned off in the desired access mask and on in the granted access mask.
4. If the bits for Read Permissions, Change Permissions, or Modify Owner are turned on in the desired access mask, then the Owner SID in the object's security descriptor is compared to SIDs in the access token's User and Groups fields. If a match is found, the bits are turned off in the desired access mask and on in the granted access mask.

5. The object's DACL is evaluated by examining each ACE in sequence, starting with the first ACE.
   - If an ACE's inheritance flags are marked INHERIT_ONLY, the ACE is skipped.
   - If the SID in an ACE does not match any SID in the subject's access token, the ACE is skipped.
   - If the ACE type is access-denied, the rights denied in the ACE's access mask are compared to the rights requested in the subject's desired access mask. If there are any matches, all bits are turned off in both the desired access mask and the granted access mask, and access-checking stops. The subject gains no access to the object.
   - If the ACE type is access-allowed, the rights in the ACE's access mask are compared to the rights in the subject's desired access mask. If there are any matches, corresponding bits are turned off in the desired access mask and turned on in the granted access mask.
   - If any bits in the desired access mask are still turned on, access-checking continues with the next ACE.
   - If the end of the DACL is reached before all bits are turned off in the desired access mask, then access is implicitly denied. Any bits that have been turned on in the granted access mask are turned off, and the granted access mask is returned to the calling process. The subject gains no access to the object.

Auditing

The goal of auditing is to generate entries in the security log for successful or failed attempts to access an object. AccessCheckAndAuditAlarm cannot determine whether an attempted access succeeds or fails until it completes an access check, so access-checking takes place first. After the access check is complete, the function determines what needs to be logged by considering the following information:

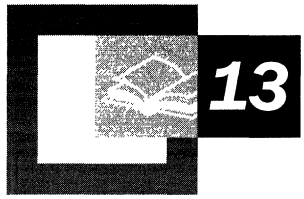
- The subject's access token
- The desired access mask presented by the subject
- The granted access mask resulting from the access check
- The object's SACL

Audit-checking is much simpler than access-checking. Only a fraction of the total number of access requests requires an entry in the security log—only those requests that you want to monitor. Security logs become unusable when they contain too much information, so you quickly learn to be judicious when setting audit controls. As a result, SACLs generally have fewer ACEs than DACLs.

AccessCheckAndAuditAlarm uses the following rules to evaluate ACEs in an object's SACL:

1. If an ACE's inheritance flags are marked INHERIT_ONLY, the ACE is skipped.
2. If the SID in an ACE does not match any SID in the subject's access token, the ACE is skipped.
3. If a bit is turned on in the ACE's access mask but the corresponding bit is not turned on in the desired access mask, the ACE is skipped. The only relevant ACEs are those that match rights that the subject has requested.
4. If an ACE's audit flags contain SUCCESSFUL_ACCESS_ACE_FLAG, the ACE's access mask is compared with the granted access mask. If a bit is turned on in the ACE's access mask and the same bit is turned on in the granted access mask, a successful access event is recorded in the security log.
5. If an ACE's audit flags contain FAILED_ACCESS_ACE_FLAG, the ACE's access mask is compared with the granted access mask. If a bit is turned on in the ACE's access mask and the same bit is turned off in the granted access mask, a failed access event is recorded in the security log.
6. When the process reaches the end of the SACL, audit-checking stops.

CHAPTER 13



Choosing Security Solutions That Use Public Key Technology

Designing an appropriate security system to protect your organization's confidential and proprietary information requires developing a set of appropriate solutions for specific risk scenarios. Many of the distributed security features that are available in the Microsoft® Windows® 2000 operating system to provide strong and scalable solutions for network security use public key technology. Understanding public key technology and the Windows 2000 network and information security options can help you use them effectively.

In This Chapter

Related Information in the Resource Kit

- For more information about the basic concepts of cryptography and public key technology as well as the implications of cryptography export restrictions, see "Cryptography for Network and Information Security" in this book.
- For more information about the Windows 2000 public key infrastructure and Certificate Services, see "Windows 2000 Certificate Services and Public Key Infrastructure" in this book.
- For more information about designing, testing, and deploying a public key infrastructure, see "Planning Your Public Key Infrastructure" in the *Microsoft® Windows® 2000 Server Resource Kit Deployment Planning Guide*.

Weaknesses in Open Network Security

The open nature of today's enterprise promotes information exchange and collaboration on an unprecedented level. However, the very nature of open networks makes it more difficult for organizations to protect valuable network resources and business information.

In a recent Computer Security Institute survey, 64 percent of the 520 respondents reported computer security breaches in a 12-month period—an alarming 16 percent increase over the previous year. Yet, perhaps even more alarming, 241 respondents disclosed corporate losses that totaled $136 million—a 36 percent increase over the previous year. There is clearly a need for stronger security on intranets and on the Internet.

Although Internet Protocol (IP) networks such as private intranets and the public Internet provide for open communication between users and computers, they are vulnerable to compromised security and malicious attack by intruders. Information sent over IP networks is open to anyone who can gain access to the network. Authorized network users, as well as intruders, can view, modify, and disrupt IP network communications easily by using software tools that are readily available on the Internet. Both internal users and external intruders can exploit the open nature of IP networks to do any of the following:

- Steal proprietary information and eavesdrop on private communications.
- Impersonate users, clients, or servers on the network.
- Intercept and modify information before it arrives at its destination.
- Intercept and redirect information to other destinations.
- Cause denial-of-service attacks that exploit the inherent weakness of IP networks or bugs in protocol implementation and cause harm to organizations and users on the networks.

Both the software industry and the Internet community have developed a wide range of new technologies and standards to address these problems and improve the security and privacy of information on open networks.

Windows 2000 Security Technologies

Windows 2000 includes a variety of distributed security technologies that make it possible for you to provide strong, scalable cryptography-based security for open networks or closed networks. One of the cornerstones of many modern electronic cryptographic technologies is public key technology. Many of the public key security functions of distributed security systems require a public key infrastructure (also known as a PKI). Windows 2000 includes a public key infrastructure that can support a wide range of public key information security solutions. You can use public key technology in conjunction with other security technologies to provide comprehensive protection for intranets, extranets, and the Internet.

The following Windows 2000 distributed security systems use public key technology:

- A network logon authentication that uses the Kerberos v5 authentication protocol, including logging on with smart cards (a permitted extension to the Kerberos protocol).
- A routing and remote access service that supports secure remote access to network resources. Routing and Remote Access supports the following:
  - Integration with Active Directory™, the Windows 2000 directory service that makes it possible to manage remote user authentication through the use of domain network user accounts and Group Policy settings.
  - Remote Authentication Dial-in User Service (RADIUS), which makes it possible to manage remote user authentication through a variety of authentication protocols.
  - User authentication that is based on the Extensible Authentication Protocol and Transport Layer Security (EAP-TLS). Supports the authentication of users through public key certificates and the smart card logon process.
  - Confidential communication over public Internet lines by using the Layer 2 Tunneling Protocol (L2TP) and the Point-to-Point Tunneling Protocol (PPTP).
  - Remote network access and logging on through the virtual private networks and public Internet service providers.
- Microsoft® Internet Information Services, which supports Web site security through certificate mapping and secure channel communications with the Secure Sockets Layer (SSL) protocol, Transport Layer Security (TLS) protocol, and Server Gated Cryptography (SGC) protocol.

- IP Security, which supports IP-level, end-to-end authentication, integrity, anti-replay, and encrypted communication over open IP networks, including the Internet.
- Encrypting File System, which makes it possible for a user to encrypt folders and files for safekeeping and allows an administrator to recover files when the user's private key is damaged or lost.

In addition, Windows 2000 distributed security technologies support a wide range of open standards for network and information security, as recommended by the Internet Engineering Task Force (IETF) and other standards bodies. For example, the Windows 2000 public key infrastructure is based on the open standards that are recommended by the Public Key Infrastructure (X.509) (PKIX) working group of the IETF. Because Windows 2000 security is based on open standards, the security solutions you implement can operate with many standards-compliant, third-party operating systems and security products.

Public key security in Windows 2000 is based on industry-standard public key technologies, such as the Diffie-Hellman (DH) algorithm, the RSA cryptographic algorithms developed by RSA Data Security, and the Digital Signature Algorithm. Windows 2000 security also makes use of the industry-standard, X.509 version 3 digital certificates that are issued by the certification authorities that you choose to trust. Many Windows 2000 security features use public key technology as well as certificates to provide authentication, integrity, confidentiality, and nonrepudiation for network and information security.

Network Security Solutions

You can use Windows 2000 to deploy a variety of information and network security solutions that take advantage of the benefits of public key technology and certificates.These benefits include the following:

- Logging on with smart cards for local and remote network logon authentication through the use of extensions to the Kerberos v5 protocol and the EAP-TLS protocol.
- Smart card authentication and storage of certificates and private keys.
- Secure e-mail through the use of Secure/Multipurpose Internet Mail Extensions (S/MIME) clients and optional secure mail servers.
- Secure Web communication with Internet Information Services through the use of SSL, TLS, or SGC.
- Secure access to Web site resources with Internet Information Services through the use of certificate mapping to network user accounts.

- Digitally signed software that ensures the authenticity and integrity of the software you distribute on an intranet or on the Internet.
- Protection of folders and files with Encrypting File System through the use of file encryption, including the protection of portable computers for mobile users.
- Optional authentication for IP Security communications that is based on certificates.
- Cryptographic security through the use of a Federal Information Processing Standard (FIPS) 140-1–compliant cryptographic service provider (CSP).
- Enhancement of security for secure mail and secure Web communication through the use of industry-standard smart cards or FORTEZZA Crypto Cards.
- Custom applications and certificate services that meet special security needs.

You can deploy Windows 2000 Certificate Services certification authorities (CAs) to issue and manage certificates for your organization. You can also obtain certificate services from a variety of third-party vendors.

The following sections describe how you can use certificates and Windows 2000 distributed security technologies to improve network and information security. For more information about each technology, see Windows 2000 Server Help.

Secure Mail

Standard Internet mail is sent as plaintext over open networks with no security. In today's increasingly interconnected network environments, the open nature of Internet mail poses many problems for mail security. Intruders can monitor your mail servers and network traffic to obtain proprietary or sensitive information. You also risk exposure of proprietary and confidential business information when you send mail over the Internet from within your organization. Messages sent over the Internet can be intercepted and read by eavesdroppers who are monitoring Internet traffic or even by legitimate administrators of the mail servers and connectors that process and route the messages.

Even in organizations with security policies that prohibit the exposure of proprietary business information on the Internet, employees sometimes forward their office mail over the public Internet to their personal mail accounts. Employees can also inadvertently send proprietary mail to the wrong mail alias or to a mail alias that includes the addresses of people who do not have a need to know the information in the message.

Another form of intrusion is impersonation. On IP networks, anyone can impersonate mail senders by using readily available tools to counterfeit the originating IP address and mail headers. When you use standard Internet mail, you can never be sure who really sent a message or whether the contents of the message are valid. Moreover, malicious attackers can use mail to cause harm to the recipient computers and networks (for example, by sending attachments that contain viruses).

For these reasons, many organizations have placed a high priority on implementing secure mail services that provide confidential communication, data integrity, and nonrepudiation. However, until recently, many of these secure mail systems have been proprietary or have not been scalable for global communication.

In Windows 2000, you can use secure mail to ensure the integrity of messages and to enable confidential mail communication within your organization.

Standards-based Mail Clients

The S/MIME Secure Mail working group of the IETF developed the open S/MIME standard to extend the original Multipurpose Internet Mail Extensions (MIME) standard. The S/MIME standard enables the digital signing and encryption of confidential mail. Secure mail can be exchanged between S/MIME clients that run on any platform or operating system. Secure mail clients can send S/MIME messages over the Internet without regard to the types of mail servers that handle the messages between the origin of the message and the final destination because all cryptographic functions are performed on the clients, not on the servers. Mail servers treat S/MIME messages as standard MIME. The only function of Internet mail servers is to route MIME messages; they do not alter the contents of messages in transit.

Microsoft supports S/MIME in the Microsoft® Outlook® 98 messaging and collaboration client as well as in Microsoft® Outlook Express version 4 and Outlook Express version 5. Microsoft is also working with the S/MIME Mail Security working group of the IETF, as well as with other vendors, to ensure maximum interoperability of Outlook 98 and Outlook Express with other S/MIME applications. Many of the other leading mail clients from third-party vendors also support S/MIME.

Secure mail with S/MIME uses the industry-standard X.509 version 3 certificates and public key technology. To provide message authentication, data integrity, and nonrepudiation, secure mail clients can sign messages with the sender's private key before sending the messages. The recipients then use the sender's public key to verify the message by checking the digital signature. Clients require a valid secure mail certificate before they can send signed mail. Recipients must have a copy of the originator's secure mail certificate (which contains the public key) before they can verify the originator's signature.

In addition, secure mail clients can send and receive confidential mail. Clients generate random secret bulk (symmetric) encryption keys and use the secret key to encrypt messages for confidentiality. Then they protect the secret bulk encryption key by encrypting it with the public key of each recipient and sending the encrypted key along with the encrypted message to each recipient. Message originators must have a copy of the recipient's secure mail certificate (which contains the public key) before they can send confidential mail. Recipients use their private keys to decrypt the secret bulk encryption key; then they use the secret key to decrypt the message.

For more information about symmetric key encryption, public key encryption, and digital signatures, see "Cryptography for Network and Information Security" in this book.

Secure Mail Clients

By using secure mail, senders are assured that the integrity of their messages is preserved and that only the intended recipients can read the encrypted mail. Recipients are assured that the message is genuine and originated from the sender. For secure mail to work, each mail client must have a valid certificate for secure mail and each client must trust the root CA in the certification path of the other client's secure mail certificates. Certificates can be published in Lightweight Directory Authentication Protocol (LDAP) directories, public folders, and Web pages to facilitate the distribution of certificates and public keys. In Windows 2000, secure mail certificates are published to Active Directory for the user account that is issued a certificate. You also have the option of configuring Certificate Services to publish certificates to public folders, Web pages, or other LDAP-compliant directory services. Users with mail clients that support LDAP, such as Outlook 98 or Outlook Express, can browse directory services to locate and obtain the published certificates of others.

Secure mail clients must trust the certificates from other correspondents. You can configure secure mail for your organization to trust secure mail certificates that are issued by CAs in your organization or to trust secure mail certificates that are issued by third-party CAs. If you trust only the secure mail root CAs in your organization, secure mail communications are limited to transactions between employees. However, you can enable secure mail transactions with third parties by trusting their secure mail root CAs.

The strength of the encryption cryptography that is available for secure mail clients depends on the current export or import restrictions for cryptography that are required by many governments. The actual cryptographic strength that is available to your mail clients depends on the cryptography restrictions that apply for the locality where the mail clients are deployed and for the locality where the mail clients are installed. In general, mail clients with exportable technology provide much weaker security than mail clients with nonexportable cryptography.

You can deploy Certificate Services so that it issues secure mail certificates that work with S/MIME-compliant secure mail clients such as Outlook 98 or Outlook Express. You can deploy Web enrollment pages to enroll users and issue secure mail certificates.

In addition, you can use a mail service, such as the Microsoft® Exchange Server version 5.5 client/server messaging and groupware, to provide management services for secure mail. You can deploy Exchange Server and use the Key Management server (KM server) to manage secure mail certificate enrollment for Certificate Services. You can also use KM server to provide key recovery services as described in the following section.

For more information about trust for CAs and the certification path, see "Windows 2000 Certificate Services and Public Key Infrastructure" in this book.

Key Management and Key Recovery Services

Because the S/MIME secure mail standard is client-based, it does not specify key management or key recovery requirements. However, you can use a key management service, such as KM server, to provide secure mail recovery services for keys that are not used to sign messages. Key recovery services maintain copies of nonsigning private keys in a central, protected database, where only authorized security administrators can obtain copies of the keys and restore them to their owners when necessary. Keys are maintained and transmitted in a password-protected, encrypted format to ensure their confidentiality.

Important When a copy exists of a private key that is used for digital signing, the integrity and the nonrepudiation provided by the signing key are compromised. If anyone else can gain access to and use the private key, they can impersonate the owner of the key. Then he or she could, for example, forge the digital signature and send counterfeit messages. Therefore, private keys that are used for digital signing should never be copied (exported) or stored in a key recovery system.

Without key recovery, an encrypted message that is sent to a user cannot be decrypted if the private key has somehow been corrupted or destroyed. For example, if a user's hard disk crashes and cannot be restored and the user has secure mail, other people can still send encrypted messages to the user. But if the private key cannot be restored, the user cannot read the encrypted messages. Therefore, the nonsigning private keys that are used for key recovery systems should be maintained in a protected database for at least the lifetime of the corresponding secure mail certificates. When the certificate expires, the public key cannot be used any longer to send secure mail.

The private keys for secure mail are used for encrypting secret bulk encryption keys for confidential mail and for the digital signing of messages. If you do not use a key recovery system, you can use one certificate for both confidential mail and signing mail. However, if you provide key recovery services, you must issue two certificates to users—one that is used only for confidential mail and the other that is used only for signing messages. The private key that is used for confidential mail can be exported and stored for key recovery. The private key that is used for signing mail cannot be exported and is not stored by the key recovery service.

Although the Windows 2000 public key infrastructure does not provide key recovery or dedicated certificate enrollment services for secure mail, Exchange Server 5.5 provides the KM server for managing certificate mail enrollment and for recovering private keys. The KM server maintains a protected key recovery database that contains all private keys for certificates that have been issued for secure mail. It stores private keys until the certificate expires or is revoked and the public key is no longer used. Administrators can use the KM server to recover private keys and securely restore the keys to their owners—for example, if the private key has been damaged on the owner's computer. Starting with Exchange Server version 5.5 Service Pack 1, the KM server uses Certificate Services to issue the secure mail certificates. Earlier versions of Exchange did not use Certificate Services; instead, the KM server issued the now obsolete X.509 version 1 certificates. To use Certificate Services with Exchange, install a CA, and then install the Exchange Policy module. The CA uses the Exchange Policy module to issue secure mail certificates upon receipt of valid requests for certificates from the KM server.

For more information about the Exchange Policy module and how to use Certificate Services with Exchange Server, see Certificate Services Help and Exchange Server Help.

Secure Web Communications

Internet communications that are based on the Transfer Control Protocol/Internet Protocol (TCP/IP), such as the Hypertext Transfer Protocol (HTTP), Telnet, and File Transfer Protocol (FTP), are not secure because all communication occurs in plaintext. Confidential or sensitive information that is transmitted with these protocols can easily be intercepted and read unless the information is protected by encryption technology.

In addition, because any Web client can send HTTP requests to a Web server and exploit weaknesses in the HTTP protocol or its implementation, Web servers that use only standard HTTP to communicate with Web clients are easy targets for denial-of-service attacks and other types of attacks. Moreover, Web clients that communicate by using standard HTTP are easy targets for unauthorized Web servers, which can impersonate legitimate Web sites and which might contain either virus-laden software for download by users or malicious scripts and programs.

In Windows 2000, you can deploy Internet Information Services to provide secure Web communication channels to ensure the integrity and confidentiality of Web communications in your organization. For more information about Internet Information Services, see Internet Information Services Help and the *Microsoft® Windows® 2000 Server Resource Kit Internet Information Services Resource Guide*.

Protocols for Secure Web Communication

Secure Web communication protocols provide a way to authenticate clients and servers on the Web and to protect the confidentiality of communication between clients and servers. A variety of secure communication standards that use public key technology have been developed, including Secure Hypertext Transfer Protocol (SHTTP), IP Security (IPSec), PPTP, and L2TP. The leading general-purpose, secure Web communication protocols are SSL 3.0 and the open TLS protocol that is based on SSL. The SSL and TLS protocols are widely used to provide secure channels for confidential TCP/IP communication on the Web.

One disadvantage of SSL and TLS, however, is that the strength of the cryptography that is used for secure channels is subject to government export and import restrictions. For example, the strength of symmetric key encryption that is used by technology that is nonexportable is much higher (128 bits) than the strength of the symmetric key cryptography that is used by technology that is exportable (40 bits or 56 bits). Both servers and clients must use the same cryptographic strength and the same cryptography algorithms when they communicate over a secure channel. At the beginning of SSL and TLS sessions, the server chooses the strongest cryptography that is available to both the server and the client. Maximum security for secure SSL and TLS communication is available only between servers and clients that can both support the higher-strength nonexportable cryptography.

For secure Web communication with banks and other financial institutions, other specialized protocols that use strong cryptography have been developed (as allowed by import or export restrictions on cryptography). Qualifying institutions can use these special protocols to provide strong cryptography for Web transactions, and at the same time circumvent the import and export restrictions that apply for SSL and TLS. Two of the leading secure Web communication protocols of this type are the Secure Electronic Transaction (SET) protocol and the SGC protocol. Internet Information Services supports SGC; Microsoft® Wallet (which is available for the Microsoft® Internet Explorer 5 Internet browser) supports SET.

The SGC protocol is an extension of SSL, which requires a special SGC certificate to enable strong, 128-bit secure communication for the Web server. Internet Explorer and many other Web clients support SGC for both exportable and nonexportable versions of Web clients. Web clients do not need certificates for SGC communication. However, to use SGC communication with a Web server, you must obtain an SGC server certificate from an authorized, commercial CA. The commercial CA that issues your SGC certificate verifies that you are qualified to use SGC. Currently, many financial institutions and institutions in other specific industries can qualify for SGC certificates. For more information about SGC and qualifying institutions, see the Microsoft Security Advisor link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Benefits of Secure Web Communication

SSL and TLS can provide the following protection for Web communication:

- Server authentication that is based on the server's authentication certificate. Clients can identify the server by its certificate and can choose to communicate with authenticated servers. Clients can detect whether an unauthorized entity is trying to impersonate a legitimate Web server.
- Optional, mutual authentication between servers and clients that is based on both clients and servers having trusted certificates. Both servers and clients can choose to trust only those certificates that are issued by a specific CA.
- Confidential Web communications that use an encrypted, secure channel. The server and the client negotiate the cryptographic algorithms that are to be used. They also negotiate the secret, shared session key that is used for secure communication. By default, secure Web communication uses the longest secret session key that is supported by both the server and the client. You can configure Internet Information Services to require strong (128-bit) session keys, or you can allow secure communication with shorter keys that are supported by exportable Web clients.

- Data integrity that is based on Hash Message Authentication Codes (HMACs). An intruder cannot tamper with the data because the transmitted information is accompanied by message digests that must be verified before the information is accepted by the receiving client or server.

Internet Information Services and Web Communication

Windows 2000 Server includes Internet Information Services, which you can use to deploy Web sites on your intranets, extranets, or the Internet. You can use the Internet Information Services console in Microsoft Management Console (MMC) to configure and manage your Web sites, including configuring them for SSL and TLS Web communication. You can use the Internet Information Services console in Microsoft Management Console (MMC) to configure

For secure Web communication in which the client authenticates the server, only the server must have a valid authentication certificate. For optional secure Web communication that requires authentication of both the server and the client, the Web server and the Web client both must have valid certificates. If either of them does not have a valid authentication certificate, the secure channel is not completed.

You can use Certificate Services to issue client and server authentication certificates that work with both Internet Information Services and Internet Explorer, as well as with many other Web server and Web client products. You can also use other certificate services to provide certificates for secure Web communication.

For secure Web communication on an intranet or on extranets, you can often reduce the costs of certificate services by deploying your own CAs. You control the certificates that are trusted for authentication. However, for secure Web communication on the Internet, clients are usually more likely to trust a server certificate that has been issued by a commercial CA than a server certificate that has been issued by your organization. Furthermore, it is generally very costly to manage your own certificate services for secure Web client authentication certificates for public users on the Internet. Therefore, for Internet communication, consider obtaining Web server authentication certificates from commercial CAs. For Internet Web clients, consider trusting client authentication certificates from leading commercial CAs.

When you configure Internet Information Services to require client authentication certificates, the only Web clients that can communicate with the Web site are those whose authentication certificates are valid and contain a certification path with a root CA that has its certificate in the Trusted Root Certification Authorities store. But you can create certificate trust lists (CTLs) to specify other CAs that are trusted for each Web site. Because root CAs with certificates in the Trusted Root Certification Authorities store are trusted for a wide range of security functions , consider using CTLs whenever you want to trust a root CA from outside your organization. For example, you might create CTLs to trust client authentication certificates that are issued by business partners or by a commercial CA.

Secure Web Communication Options

By default, secure Web communication for Internet Information Services requires authentication of the server to clients, but clients are not required to have a certificate for authentication to the server. You have the option of configuring Internet Information Services to prevent clients from communicating with Web sites if they do not have valid and trusted authentication certificates, which reduces the risk of denial-of-service attacks. You can require secure channels for all Web communication to protect confidential information, or you can use secure channels to transmit only selected information, such as personal information and credit card numbers.

The users of your Web sites on the Internet are usually customers that use a mixture of Web browsers, including Web browsers that support strong cryptography and Web clients that do not support strong cryptography. With Internet Information Services, the default configuration for SSL and TLS communication enables secure channels to default to the lowest strength of cryptography that is supported by each user's Web browser.

You have the option of configuring Web sites to require the maximum 128-bit cryptographic strength for SSL and TLS clients so that Web clients that support only the weaker cryptography cannot communicate with the Web site. For example, you might configure Internet Information Services to require strong encryption on your intranet or extranets. Of course, you must also deploy Web clients and Web servers in your organization to support the stronger cryptography requirements.

If you meet the SGC qualification requirements, you also might be able to use SGC on the Internet to provide strong 128-bit encrypted communication with clients that support SGC. To use SGC with Internet Information Services, you must obtain an SGC server certificate from a commercial CA. Or you can configure Internet Information Services to use SSL, TLS, and SGC. If the Web client supports SGC, SGC is used; otherwise SSL or TLS is used.

Secure channels require that information be both encrypted and decrypted on both the server and the client. Most of the client computers in use today can handle the encryption and decryption load of secure channels easily. However, multiple, concurrent secure channels can place a large load on a Web server. To improve the performance of secure-channel Web servers, you can use very fast multiple processor computers and crypto-accelerator boards. You can also design Web sites to use secure channel sessions for only a portion of the content on the Web site. For example, you can design Web pages to require secure channels only for forms that are used to submit confidential information, such as credit card numbers or payroll information.

Secure Access to Web Site Resources

You can use the Internet Information Services console to provide granular security for your Web sites. You can configure security for an entire site as well as for individual folders and files in a site. You can set general security for a Web site and then add other security requirements for specific folders and files in the site. Internet Information Services provides the following methods for controlling access to Web site resources:

- Anonymous access
- Authenticated access
- IP addresses and domain names
- Certificate mapping to user accounts

In addition, Internet Information Services uses access control lists (ACLs) to control access to Web resources that are installed on NTFS file systems.

Anonymous Access

When Internet Information Services is configured for anonymous access, the server logs on each Web user with an anonymous guest account, which can be any valid Windows 2000 user account. Windows 2000 Server provides a built-in local account for anonymous access: IUSR_*Server* (where *Server* is the name of the server computer on which Internet Information Services is installed). Internet Information Services provides a default set of user rights and permissions for anonymous-access Web sites. For example, by default, the IUSR_*Server* account allows general users read permission for most Web site resources. However, administrators and the system have full control of Web site resources. Instead of using the IUSR_*Server* account, you can select another user account for use by anonymous-access users if you want.

When anonymous access is enabled, users can usually log on and request Web resources without having to enter their user name and password. However, if the resources that they are requesting have NTFS ACL restrictions, they are required to submit their user name and password before being granted access to the resources. Anonymous access is generally suitable for Web sites that contain information for public consumption, but not for Web sites that contain proprietary or confidential information.

Authenticated Access

When Internet Information Services is configured for authenticated access, a user must enter a valid Windows user account name and password to gain access to the Web resources that he or she has requested. Authentication options determine whether a user's password is sent over the network as plaintext, as a message digest, or as ciphertext. Authenticated access is appropriate generally for providing low-level security on Web sites, but secure SSL and TLS channels and certificate mapping provide much stronger security.

IP Addresses and Domain Names

You can list IP addresses and domain names to grant access to selected Web resources or to deny access to selected Web resources by listing individual computers or groups of computers on the basis of subnet address or domain names. However, extensive use of domain names can slow Web performance because of time-consuming Domain Name System (DNS) lookups. Controlling Web access through IP addresses and domain names also provides only weak security because attackers can easily forge the originating IP address and domain name information to gain access to your Web site. Security is weakened further if you grant Web site access to the IP addresses or domain names for proxy services because, in this situation, you have extended access to all clients who connect to your Web site through these proxy services.

Certificate Mapping to User Accounts

You can map certificates to Windows 2000 user accounts to control access to selected Web resources. Certificate mapping provides for strong security that is based on the Web client ownership of a valid authentication certificate. When certificate mapping is enabled, Internet Information Services authenticates users on the basis of mapped certificates, and it grants rights and permissions that are based on the mapped user account. Certificate mapping can be one-to-one or many-to-one.

One-to-One Certificate Mapping For one-to-one certificate mapping, you map individual certificates to the corresponding Windows 2000 user accounts for the owners of the certificates. Internet Information Services authenticates users with the listed certificates and grants rights and permissions that are based on the user account information. One-to-one mapping works only for clients with Windows 2000 user accounts.

Many-to-One Certificate Mapping For many-to-one certificate mapping, you create rules that define the certificate criteria for mapping. Mapping rules check the information that is contained in users' certificates, such as the user's organization and the issuing CA, to determine whether the information matches the criteria in the rules. When the information in users' certificates match the rules, users are usually mapped to a particular user account that you specify. With many-to-one mapping, a user's rights and permissions for Web site resources are controlled on the basis of the rights and permissions of the mapped user account. You can also configure Internet Information Services to refuse Web access to users with certificates that match the mapping rules. You can use many-to-one certificate mapping to control access for any Web client that owns a valid authentication certificate. For example, you might use many-to-one mapping to grant access to Web resources on an extranet on the basis of Web client ownership of a specific type of certificate that has been issued by a specific commercial CA or a business partner's CA.

Administering one-to-one mapping manually requires more administrative effort than administering many-to-one mapping. Therefore, if you use one-to-one mapping for large numbers of clients, consider developing custom Active Server Pages Web enrollment pages to automate the mapping process. For more information about developing custom enrollment Web pages that map certificates automatically, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

In addition, consider using many-to-one mapping to control Web site access when it is feasible to do so. Use one-to-one mapping as necessary to control Web site access when you have a relatively small number of clients. For example, you might use one-to-one mapping to grant Web site access to several administrators and then use many-to-one mapping to grant Web site access to a large number of employees from your organization, selected consultants, and selected employees of business partners.

You can use separate many-to-one certificate mappings for each group to which you want to grant Web site access. You can configure user accounts that grant different sets of rights and permissions on the basis of clients' ownership of valid certificates that match the mapping rules. For example, you can map your employees to a user account that grants them full read access to the Web site, but map consultants and employees of business partners to other accounts that allow access only to nonconfidential information and selected proprietary information.

NTFS Access Control Lists

NTFS ACLs extend the granular security that is available for Web sites. When Web sites are installed on NTFS file systems, user rights and access permissions for Web resources are controlled by file system ACLs. You can configure these lists to control access to individual Web sites, folders, or files. You can use Windows Explorer to grant rights and permissions for user accounts and security groups. When folders and files have ACL restrictions, Internet Information Services prompts users to enter their Windows 2000 user names and passwords for authenticated access, even when anonymous access is enabled for the requested resource. Because this security is not available for file allocation table (FAT) or FAT32 file systems, consider using NTFS for all of your Web sites to provide the additional security benefits of NTFS file and folder security.

Digitally Signed Software

Software that is downloaded from the Internet to users' computers can contain programs such as viruses and Trojan horses that are designed to cause malicious damage or provide clandestine network access to intruders. As networks become more interconnected, malicious software and viruses also become a threat to intranets. To help counter this growing threat, you can digitally sign the software that you distribute on your intranets or the Internet to ensure its integrity and to assure others that the software can be trusted. Signed software ensures that users can verify the origin of the software, as well as verify that no one has tampered with it.

Microsoft developed the Microsoft® Authenticode® technology, which enables developers to digitally sign software. The last thing developers do before they release software is digitally sign the software. Any modification to the software after it is signed invalidates the digital signature. By using Authenticode technology, code signers who own valid X.509 version 3 code-signing certificates can sign software with their private key. Several other third-party code signing technologies also use digital certificates to enable code signing.

Code Signing Within Your Organization

Executable programs, scripts, and ActiveX® controls that are distributed in Windows 2000 domains should be digitally signed by trusted developers. To protect your network from malicious programs and viruses, you can configure Internet Explorer to specify security settings for the Internet, local intranet, Trusted sites, and Restricted sites security zones. You can specify security settings that prevent users from downloading and running unsigned software from any security zone. You can also configure Internet Explorer to trust specific software publishers so that any software that is signed by these publishers is downloaded automatically without notifying the users. For more information about Internet Explorer security, see the *Microsoft® Windows® 2000 Server Resource Kit Internet Explorer Resource Guide*.

In addition, you can configure Public Key Group Policy to specify the CAs for code signing that are trusted in your organization. You can trust software publishing certificates that are issued either by commercial CAs or by your CAs. You can also create and use CTLs to establish trust in the domain for code-signing certificates.

You can use Certificate Services to issue code-signing certificates to the developers who sign software for distribution on your intranet.

Code Signing on the Internet

When software is distributed over the Internet, users are more likely to trust software that is signed by a publisher whose code-signing certificates ("software publisher certificates") have been issued by a reputable commercial CA. Using commercial CAs also removes the liability placed on your organization when you assume the responsibilities of a commercial CA for external software distribution. Therefore, if you distribute software on the Internet, consider obtaining the services of a commercial CA to issue digital signing certificates to your external software developers.

Consider providing special protection for the private keys that are used to sign code. If someone obtains access to a private key for code signing, they can impersonate your organization, distribute signed but defective or malicious code, and damage your organization's reputation. Some third-party vendors offer smart card solutions that enable code signing with smart cards. You can establish a smart card program for code signers that provides additional protection for their private keys.

Automating Code Signing and Software Distribution

You can build custom applications to automate code signing and the distribution of software within your organization or to external Web sites. Internal and external developers or program managers who have valid code-signing certificates can use custom applications to submit code to be signed automatically and processed for distribution. Deploying code-signing applications includes the following activities:

- Installing CAs that issue code-signing certificates or obtaining certificate services from a commercial CA.
- Developing the custom applications for code signing and software distribution.
- Issuing code-signing certificates to the appropriate developers or program managers.
- Configuring the software distribution infrastructure and services.

For example, you might use Active Server Pages (ASP) technology and Internet Information Services to build code-signing and software distribution Web sites. You might configure one-to-one certificate mapping to grant permission for use of the Web site to users who have valid code-signing certificates. Users who do not have valid code-signing certificates are not permitted to use the site to submit code for signing and distribution.

Security with Encrypting File System

Windows 2000 includes Encrypting File System (EFS), which is a security technology that enables an individual user to encrypt files so that the files cannot be read by others.

EFS uses an encryption attribute to designate files for EFS protection. When a file's encryption attribute is on, EFS stores the file as encrypted ciphertext. When an authorized user opens an encrypted file in an application, EFS decrypts the file in the background and provides a plaintext copy to the application. The authorized user can view or modify the file, and EFS saves any changes transparently as ciphertext. Other users are denied permission to view or modify EFS-encrypted files. EFS-protected files are bulk encrypted to provide confidentiality even from intruders who bypass EFS and attempt to read files by using low-level disk tools.

Because EFS operates in the background at the system level, applications can save temporary files as plaintext to non-EFS-protected folders and inadvertently compromise confidentiality. Therefore, encryption usually must be enforced at the folder level rather than the file level. This means that you do not encrypt individual files, but instead designate folders as EFS-protected folders. All files that are added to EFS-protected folders are encrypted automatically. To specify EFS protection for a folder, use the properties page for the folder in Windows Explorer.

EFS is supported only for the version of NTFS that is included with Windows 2000. It does not work with any other file system, including the previous versions of NTFS. For more information about EFS, see "Encrypting File System" in this book.

File Encryption and Public Key Technology

For EFS to work, the EFS user must have a valid EFS user's certificate, and at least one EFS recovery agent account must have a valid EFS recovery certificate. EFS does not require a CA to issue certificates because EFS automatically generates its own certificates to users and to default recovery agent accounts. The EFS private key is generated and managed by Microsoft Cryptographic Application Programming Interface (CryptoAPI) in conjunction with the base Microsoft CSP.

When EFS encrypts a file, it does the following:

1. Generates a bulk symmetric encryption key.
2. Encrypts files by using the bulk encryption key.
3. Encrypts the bulk encryption key by using the EFS user's public key.
4. Stores the encrypted bulk key in a special field called the data decryption field (DDF), which is attached to the EFS file.

EFS can then use the user's private key to decrypt the bulk encryption key and decrypt the file as necessary. Because only the user has the private key, others cannot unlock the DDF.

In addition, EFS enables designated recovery agent accounts to decrypt and recover the file in case the user's private key is lost or damaged. For each designated recovery agent account, EFS does the following:

1. Encrypts the bulk encryption key by using the public key from each recovery agent certificate.
2. Stores the encrypted bulk key in a special field called the data recovery field (DRF), which is attached to the EFS file.

The data recovery field can contain information for multiple recovery agent accounts. Every time a file system operation is complete for a file, such as viewing, opening, copying, or moving the file, EFS generates and saves a new DRF with the most current public keys for the current recovery agent certificates. You can designate recovery agent accounts by configuring Encrypted Data Recovery Agents Group Policy settings.

Encrypted Data Recovery Policy

You might want to recover encrypted files, for example, when an employee is terminated for cause or when a user's private key for EFS is damaged. You can use the command-line tool, Cipher, to recover files on a recovery computer where a current recovery agent account, certificate, and private key are located. To recover a file, a recovery administrator must log on to the recovery computer as the recovery agent account and then use Cipher to decrypt the file. Cipher only works for the recovery agent accounts that are listed in the files DRF. Cipher also only works if the private key for recovery is installed on the computer.

Encrypted Data Recovery Agent Group Policy settings are a subset of Public Key Group Policy. You can configure Encrypted Data Recovery Agent settings to designate recovery agent accounts for domains, organizational units (also known as OUs), or stand-alone computers. Trusted recovery administrators that you designate can then use the recovery agent accounts to recover EFS encrypted files for the domains or organizational units where the EFS recovery settings apply.

When Group Policy is downloaded to computers, the Encrypted Data Recovery Agent Group Policy settings contain the certificates for each designated recovery agent account within the scope of the policy. EFS uses the information in the current Encrypted Data Recovery Agent Group Policy settings to create and update DRFs. A recovery agent certificate contains the public key and information that uniquely identifies the recovery agent account.

By default, the domain Administrator user's account on the first domain controller that is installed in the domain is the recovery agent for computers that are connected to the network. On stand-alone computers, the local Administrator user account is the default EFS recovery agent. EFS generates EFS recovery certificates automatically for default Administrator accounts.

Note You can disable EFS for a domain, organizational unit, or stand-alone computer by applying an "empty" Encrypted Data Recovery Agents policy setting. Until Encrypted Data Recovery Agent settings are configured and applied through Group Policy, there is "no" policy, and the default recovery agents are used by EFS. However, EFS must use the recovery agents that are listed in the Encrypted Data Recovery Agents Group Policy after the settings have been configured and applied. If the policy that is applied is empty, EFS does not operate.

The Windows 2000 Resource Kit includes the tool Efsinfo.exe, which you can use to view information about the recovery agent accounts. You can use Efsinfo to verify what recovery accounts are current for an encrypted file.

Recovery Agent Accounts

The default EFS recovery agent accounts might meet the needs of some organizations. However, other organizations might want to issue EFS recovery certificates to designate other recovery agent accounts for more flexible EFS recovery management. For example, you can issue recovery certificates to dedicated EFS recovery computers rather than to the default Administrator user account on domain controllers. You can also configure Encrypted Data Recovery Agents settings for portable computers so that they use the same recovery agent certificates when they are connected to the domain and when they are operated as stand-alone computers.

Rather than attempting to manage EFS recovery on a domain-wide basis, consider designating recovery agent accounts to manage recovery for each organizational unit. You can designate several recovery accounts to recover users' files as necessary for users within that organizational unit.

You can deploy Certificate Services to issue EFS user certificates to users and EFS recovery certificates to recovery accounts. If an enterprise issuing CA is available, EFS requests EFS certificates from the enterprise CA instead of generating its own certificates. Using an enterprise CA to manage EFS certificates provides the benefits of centralized certificate management and enables you to publish certificate revocation lists (CRLs) for EFS certificates.

Old EFS files for which there have been no file operations, such as viewing, opening, copying, or moving for a long period of time might contain out-of-date recovery agent information. Therefore, it is recommended that you maintain a recovery agent archive to ensure that EFS files can be recovered by using obsolete recovery agent certificates and their private keys. For more information about EFS recovery, see "Windows 2000 Certificate Services and Public Key Infrastructure" and "Encrypting File System" in this book.

Security for Portable Computers

One of the most common uses of EFS is the encryption of data files on portable computers, which protects data if the computer is stolen. EFS protection is usually applied by using the properties page for the folder in Windows Explorer. However, you can also use the EFS command-line tool, Cipher, to apply EFS protection.

To provide EFS security for a portable computer, do the following:

Make sure the user's My Documents folder is empty, and then apply EFS protection to this folder. Therefore, all new files that are stored in the EFS-protected folder are encrypted, and all new subfolders that are created in the EFS-protected folder are protected. Users can create as many folders in My Documents as they need. If you add encrypted files to the folder, the user cannot read those files.

Apply EFS protection for temporary folders that are used by applications. Applications work with only plaintext because EFS operates in the background. If EFS protection is not applied to the folders where the temporary files are stored, applications could save their temporary files as plaintext to the folders. You can also configure applications to store temporary files in EFS-protected folders.

Configure NTFS file system ACLs to prevent users from creating non-EFS-protected folders and from changing EFS settings.

Use the Windows 2000 System Key (SysKey) to protect EFS private keys. SysKey uses strong encryption techniques to increase the protection of users' protected stores, including users' private keys for EFS.

Security with IP Security

Windows 2000 incorporates a suite of protocols called IP Security (IPSec). IPSec is based on standards that were developed by the IP Security Protocol (IPSEC) working group of the IETF. IPSec operates at the IP network and TCP/UDP transport levels and is transparent to the operating system and applications. IPSec provides end-to-end security between sending and receiving computers on IP networks. IPSec can be configured to perform one or more of the following security functions:

- Authenticates the sender of IP data packets on the basis of Kerberos authentication, digital certificates, or a shared secret key (password).
- Ensures the integrity of the IP data packets that are transmitted over the network.

- Encrypts all data that is sent over the network for full confidentiality.
- Hides the originating IP addresses from observation while they are en route.

For Kerberos clients, you usually configure IPSec security policy to authenticate clients on the basis of Kerberos authentication. However, you can also configure IPSec policy to use digital certificates for authenticating non-Kerberos clients (for example, business partner clients on your extranet). Certificates provide the strongest security for authenticating non-Kerberos clients for IPSec communication. The use of shared secrets (passwords) for authentication provides relatively weak security and, thus, is not generally used except when necessary to test IPSec or when necessary to provide interoperability with some third-party IPSec clients. You can configure IPSec security policy for each domain or for each local computer by defining a list of rules and filters that are to be applied to regulate secure communication with specific IPSec clients.

You can deploy Certificate Services to issue IPSec authentication certificates, or you can obtain certificate services from third parties. You must configure IPSec security policy to specify the CAs that you trust and that are to issue the IPSec certificates for authentication. For example, you might deploy Certificate Services to provide IPSec certificates to non-Windows 2000–based clients in your organization and configure IPSec security policy to trust certificates that are issued by appropriate CAs. You might also choose to trust certificates that are issued by third-party CAs (for example, the CA for a business partner) for IPSec communications with computers on an extranet.

For more information about IPSec, see "Internet Protocol Security" in the *Microsoft® Windows® 2000 Server Resource Kit TCP/IP Core Networking Guide.*

Security with Smart Cards

Smart cards are credit card–sized plastic cards that contain integrated circuit cards. Smart cards are tamperproof and can be used to store users' certificates and private keys. Smart cards can perform sophisticated public key cryptography operations, such as digital signing and key exchange.

You can deploy smart cards and smart card readers to provide stronger user authentication and nonrepudiation for a range of security solutions, including logging on over a network, secure Web communication, and secure mail.

Benefits of Smart Cards

The benefits of using smart cards include the following:

- Because private keys are stored on the tamper-resistant smart card rather than on a less secure medium such as the user's hard disk, smart cards provide stronger security for user authentication and nonrepudiation.
- Because cryptographic operations are isolated from the operating system, smart cards are not susceptible to attacks on the operating system (such as buffer overflow attacks and memory dump attacks,which might reveal private keys or other cryptographic secrets).
- Because logon credentials accompany the user, you can issue a single smart card to each network user to provide a single set of logon credentials for logging on to local and remote networks, which can reduce the cost of administering separate user accounts for logging on to a network and logging on remotely.

In addition, logging on with a smart card provides much stronger security than other types of network logon processes that depend on traditional passwords. Furthermore, because the administrative support that is required to manage user passwords is a significant cost for most large organizations, you can deploy smart cards to reduce the cost of supporting users who forget their passwords or let their passwords expire. Smart cards use personal identification numbers (PINs) rather than passwords. The smart card is protected from misuse by the PIN, which is known only to the smart card's owner. To use the smart card, the user inserts the card in a smart card reader that is attached to a computer and, when prompted, enters the PIN. The smart card can be used only by someone who possesses the smart card and knows the PIN.

Personal Identification Numbers vs. Passwords

PINs offer significantly more protection than standard network passwords. Passwords (or derivations such as hashes) travel on the wire and are subject to brute-force attacks (key search attacks), in which an attacker tries all possible password combinations until he or she finds the password. Passwords are also subject to dictionary attacks, in which an attacker tries known words in the dictionary and numerous common password names in an attempt to guess the password. Because most users prefer easily remembered passwords, dictionary attacks are often a shortcut to finding a password in significantly less time than brute force attacks would take to find the same password. Therefore, the strength of a password depends largely on how long the password is, how well it is protected from being revealed by the owner, how well it is protected if it is "sniffed" on the network, and how hard it is to guess. Even good passwords that are protected by cryptography on the wire and are not subject to dictionary attacks can be broken by brute force in a few weeks or months by an attacker who sniffs the password on the wire.

In contrast, PINs never travel on the network, so they cannot be sniffed. In addition, dictionary attacks or brute-force attacks can be attempted only by someone in physical possession of the card. However, even when an attacker has the card in his or her possession, the smart card locks after only a few failed attempts by the attacker to guess the PIN. So dictionary attacks and brute-force attacks on smart cards are not feasible even then.

Another benefit of smart cards is that policies for PINs can be less restrictive than policies for network passwords. In general, good network passwords have to be changed often and require long, complex composition. Because users are more likely to write down their long, hard-to-remember network passwords, network security is weakened. Good PINs, however, can be changed infrequently and can be relatively short. Because users are more likely to remember their short PINs, network security is less likely to be broken by a misplaced written PIN.

Network Smart Card Logon Process

Windows 2000 supports logging on with a smart card for the network logon process by using extensions to the Kerberos v5 protocol. For logging on to a network, users usually press CTRL+ALT+DEL to initiate the Windows 2000 secure logon sequence. When the smart card logon process is enabled, a user inserts the smart card to initiate the Windows 2000 secure logon sequence. The user is then prompted to enter the PIN for the smart card. If the user's PIN and smart card credentials are valid, the user is logged on and granted rights and permissions for the user account.

When an administrator enrolls for a smart card logon certificate on behalf of the user, Windows 2000 automatically maps the smart card certificate to the user's account in Active Directory. Therefore, smart card certificates for logging on to the network must be issued by a trusted enterprise CA.

If you deploy smart cards for logging on to the network in a domain and allow some users to log on without smart cards (for example, with CTRL+ALT+DEL for Windows 2000–based clients or with NTLM for clients based on Microsoft® Windows® 98 and Microsoft® Windows NT®), the security of the network becomes only as good as the weakest password in the system. For maximum network logon security, deploy Windows 2000 and smart cards for all users and require that smart cards be used for logging on to all computers in your domains, including logging on from a remote location.

Note You can use smart cards for logging on to computers even when the computers are offline and disconnected from the domain. When computers are configured for the smart card logon process, the smart card is used to authenticate users when they log on to a local computer or a network. Therefore, you can deploy smart cards for portable computers that are used by mobile users without requiring either separate logon credentials or separate logon processes for online and offline operation.

Remote Access Logon Process

Windows 2000 Server includes routing and remote access services to authenticate remote access network users. Routing and Remote Access supports smart card logon authentication by using the EAP-TLS extension of the Point-to-Point Protocol (PPP). When EAP-TLS is enabled, a remote access user is prompted to insert the smart card and enter the PIN during network logon authentication. If the user's PIN and smart card credentials are valid, the user is logged on and granted rights and permissions for the appropriate network user account. For more information about EAP-TLS, see "Internet Authentication Service" in the *Microsoft® Windows® 2000 Server Resource Kit Internetworking Guide*.

Other Smart Card Applications

A variety of third-party vendors manufacture Personal Computer Smart Card (PC/SC)–compliant smart cards and smart card readers that work with computers that are running the operating systems Windows 2000, Microsoft® Windows® 95, Windows 98, or Windows NT version 4.0. You can deploy smart cards and smart card readers in your organization to provide stronger security for Web communication, Web site access, and secure e-mail by using smart card–aware applications.

Smart card–aware applications, such as Microsoft Internet Explorer 5 and Outlook 98, can use smart cards to perform public key cryptography operations. For example, Internet Explorer can use smart cards to authenticate users for secure SSL and TLS communication sessions on the Web. Outlook 98 can use smart cards to perform S/MIME operations for secure mail.

Third-party vendors provide a variety of smart card applications that might meet your needs. For example, some third-party vendors provide code-signing applications that use smart cards to digitally sign software by using Authenticode technology. You can also build your own custom applications that use smart cards. For example, you can develop your own code-signing application to sign code by using smart cards.

Note Some third-party file encryption products support smart cards. EFS, however, does not support smart cards because EFS is designed to work transparently without the need for user interaction.

Smart Card Enrollment

You can use Certificate Services Web pages and the Smart Card Enrollment Station (available from the Advanced Certificates Request Web page) to issue smart card certificates on behalf of users. Security administrators can centrally issue and manage the smart card program to provide a high level of network user assurance. If you allow users to request their own smart card certificates, it weakens the overall security provided by smart cards.

You can choose to allow smart card certificates to be renewed automatically for Windows 2000–based clients. However, to ensure the highest levels of network security, some organizations might want to re-issue smart cards and PINs on a periodic basis. PINs can be changed only when smart card certificates are issued or renewed by the smart card CSP.

For more information about the Smart Card Enrollment Station, see "Windows 2000 Certificate Services and Public Key Infrastructure" in this book.

Smart Card Compatibility

Windows 2000 supports Plug and Play smart cards and smart card readers that have been authorized to display the Microsoft "Designed for Windows" logo. The "Designed for Windows" logo ensures that smart card products work in Windows 2000, and ensures interoperability between smart cards and smart card readers from different vendors.

Microsoft provides drivers and support for a variety of Windows-compatible logo smart card readers. Some vendors might provide drivers for noncompliant smart card readers that do not work with the smart cards of other vendors. Some vendors might also provide noncompliant smart cards that do not work with the smart card readers of other vendors. To ensure maximum continuing support and interoperability of smart cards and smart card readers, it is recommended that you deploy only Windows-compatible logo smart cards and smart card readers with Windows 2000.

For more information about the "Designed for Windows" logo program, see the Microsoft Hardware Testing link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. For the list of currently compatible smart card products, see the Microsoft Windows Hardware Compatibility List link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Smart Card Options

When you design your public key infrastructure and plan the deployment of smart cards, you have the option of doing any of the following to create a secure system.

Force Users to Use the Smart Card Logon Process Allowing the CTRL+ALT+DEL secure logon sequence for smart card users defeats the purpose of using smart cards. During the transition to smart cards, you must enable both logon methods until users are trained and the smart card logon process has been tested for your domains. Thereafter, however, you can configure individual user accounts (but not security groups) so that the CTRL+ALT+DEL secure logon process is disabled and users are forced to use their smart cards to log on to their computers. To configure individual user accounts, use the Active Directory Users and Computers console (a snap-in to MMC).

Force Systems to Lock Upon Removal of the Smart Card When a user walks away from a computer with an active logon session and the user fails to secure the computer by logging off or locking the computer, an intruder might use the computer for malicious purposes. If you are requiring the use of smart cards for logging on to computers, you can force the systems to lock when users remove their smart cards from the readers. Use this option as necessary to meet your security needs, especially when computers are used in an environment with easy access by the public. You can configure Security options under Security Settings in Group Policy to force groups of computers to lock upon the removal of smart cards.

Combine Smart Cards and Employee Badges Many organizations issue card keys and identification badges to their employees. You can add employee card keys and photographs to smart cards to provide a single solution for both building and network access. Such combination cards can be used to grant physical access to buildings and secure rooms, as well as to grant network logon access. Combination cards also can be used for electronic payment debit systems—for example, to pay for employee purchases at the organization's cafeteria or store. For more information about combining card keys and picture badges with smart cards, contact smart card vendors.

FIPS-140-1 Security and FORTEZZA Crypto Cards

Windows 2000 supports FIPS 140-1 and FORTEZZA Crypto Cards, two federal cryptographic standards that are important to the protection of United States government communications. For more information about the FIPS 140-1 and FORTEZZA Crypto Cards cryptographic standards, see the Microsoft Security Advisor link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

FIPS 140-1

The FIPS 140-1 standard was developed by the National Institute of Standards and Technology (NIST). FIPS 140-1, titled "Security Requirements for Cryptographic Modules," specifies the United States government's requirements for proper design and implementation of hardware and software cryptographic modules that perform cryptographic operations for sensitive but unclassified information. FIPS 140-1 has been adopted by the Canadian Communication Security Establishment and the American National Standards Institute. FIPS 140-1 is widely regarded as a de facto standard for cryptographic modules, and has been incorporated into International Standard 15408, "Evaluation Criteria for Information Technology Security," of the International Standards Organization.

NIST certifies modules that are FIPS 140-1 compliant. Vendors can submit hardware and software cryptographic modules, such as FORTEZZA Crypto Cards, to NIST for certification testing. FIPS 140-1 provides for increasing levels of security, from Level 1 through Level 4. These levels cover a wide range of applications and environments where cryptographic modules are used.

All Windows 2000 cryptographic service providers (CSPs) are FIPS 140-1 Level 1 compliant for use by organizations that require FIPS 140-1–level certification. For more information about Microsoft CSPs and FIPS 140-1, see the Microsoft Security Advisor link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

FORTEZZA Crypto Cards

The FORTEZZA Crypto Cards are Personal Computer Memory Card International Association (PCMCIA) cards developed by the National Security Agency. They are tamper-resistant, hardware-based security tokens that provide cryptographic services such as data confidentiality, user authentication, and data integrity. The way FORTEZZA Crypto Cards function is similar to the way that smart cards and smart card readers function, but FORTEZZA Crypto Cards have more memory and more powerful processors, and they implement the cryptographic algorithms that were chosen for the Defense Message System of the United States Department of Defense.

Like smart cards, FORTEZZA Crypto Cards can be used for secure mail and secure Web communications. FORTEZZA Crypto Cards are usually used to protect sensitive but unclassified information. However, enhanced versions of FORTEZZA Crypto Cards are also available to protect classified information.

Microsoft supports FORTEZZA Crypto Cards for secure mail by using the Defense Message System–compliant versions of Exchange Server and the Outlook 98 messaging and collaboration client. Windows 2000, Internet Explorer, and Internet Information Services also support FORTEZZA Crypto Cards for secure Web communications.

FORTEZZA Crypto Cards require the installation of PCMCIA interfaces on desktop computers and are much more expensive to deploy than industry-standard smart cards and smart card readers. Smart cards provide nearly the same level of security as FORTEZZA Crypto Cards, but for much less cost. Therefore, some United States government agencies are deploying industry-standard smart cards to provide strong security for mail and Web communication as well as interoperability with industry-standard public key information security and messaging systems. For example, the United States Department of Defense has proposed two components of their public key infrastructure—a FORTEZZA-based High Assurance Messaging System and a smart card–based Medium Assurance Messaging System.

High Assurance Messaging Systems use expensive FORTEZZA Crypto Cards and FIPS 140-1–compliant cryptographic modules to provide high-level information security. However, FORTEZZA-based systems are not compatible with industry-standard public key information security systems.

Medium Assurance Messaging Systems use inexpensive, industry-standard smart cards and public key infrastructure to provide medium-level information security. Moreover, non-Department of Defense organizations can conduct secure communications with Department of Defense agencies by using industry standard messaging and information security systems, without the need to invest in expensive FORTEZZA technology.

FORTEZZA Crypto Cards are available from a variety of National Security Agency–approved vendors. For more information about FORTEZZA Crypto Cards, contact the card vendors.

Custom Security Applications

You can deploy a variety of public key security solutions with the standard components and features included in the Windows 2000 public key infrastructure. However, if necessary, you also can develop custom applications with Certificate Services to meet special needs.

You might develop custom Policy modules and custom Exit modules to integrate certificate services with legacy databases and third-party directory services. For example, you might develop an application that validates certificate requests from user information that is contained in a legacy database or a third-party directory service.

You might also develop a custom application that issues special types of certificates. For example, you might develop an application that processes electronic documents to generate digital thumbprints of the documents and then store the thumbprints in timestamp certificates. You might maintain these timestamp certificates in a document registry database to provide integrity for the original document contents. When a document is compared to the digital thumbprint in the registry database, any tampering with or modification to the document since it was registered are identified. You might use a document registry like this to provide an online quality assurance audit trail for the products you manufacture and, thus, ensure the integrity of electronic test and certification documentation.

In addition, you might develop a custom certificate enrollment and renewal application with ASP pages. For example, you might modify the standard Certificate Services Web pages to add or delete features. You might also develop custom Web pages that integrate with third-party services or other applications that you develop.

It is recommended that you design and test all custom applications in the lab and in pilot programs before deploying the applications in your production environments. For more information about custom public key security applications with Certificate Services, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Public Key Technology Standards and Interoperability

To maximize the interoperability of Windows 2000 public key infrastructure with third-party applications that use public key technology, the Windows 2000 public key infrastructure is based on the standards recommended by the PKIX working group of the IETF. Other standards that have been recommended by the IETF also have a significant impact on public key infrastructure interoperability, including standards for TLS, S/MIME, and IPSec. Microsoft is collaborating with the other members of the IETF to develop these open standards and is committed to ensuring that its public key infrastructure products are compliant with the standards recommended by the IETF.

For cryptographic technology, the de facto standards are the Public Key Cryptography Standards (PKCS) developed and maintained by RSA Data Security, Inc. PKCS technologies are broadly deployed in products today and provide a basic, but well-tested and well-understood framework for interoperability. In fact, when the PKIX working group proposed another standard for certificate management, the S/MIME working group created its own proposal that is based on PKCS. This response is typical of the IETF process for developing standards and reflects awareness of market conditions. Microsoft has incorporated these de facto standards into its Windows 2000 public key infrastructure to maximize interoperability with third-party public key security products.

Standards, however, do not ensure interoperability between commercially available PKIX-compliant products because public key technology is still in an early stage of development. Historically, commercial product development has outpaced the collaborative process. This has been especially true for public key technology. Currently, the IETF has several working groups actively developing proposed standards for public key technology, but many of the applications to which these standards would apply are shipping already as products. Moreover, no standard can anticipate every application requirement and dependency, so standards often get watered down in implementation. Interoperability, then, is the result of standards tempered over time by vendors working together to ensure the interoperability of their products.

Today, SSL and TLS and S/MIME work well across the products of many vendors. As standards develop and vendors work together, more and more components of public key infrastructure become compatible. One day, complete interoperability might be taken for granted. However, at this time, the only way to know for sure how well products and features from different vendors work together is to test them in the lab.

How to Choose Appropriate Security Solutions

You can use Windows 2000 distributed security technologies to provide a wide range of strong, scalable, cryptography-based solutions to protect network and information security. Cryptography-based security solutions can meet a wide range of security needs in your organization. The levels of security you choose for your organization are determined by the value of the information that you want to protect and the costs to implement the various levels of security.

In general, making appropriate decisions about information security involves the following activities:

- Analyzing business information and communication.
- Defining communication and information security goals.
- Determining communication and information security requirements.

- Assessing security risks.
- Assessing the costs and benefits of various security measures.
- Testing each security solution that you choose to develop to ensure that it is the right choice for your organization.

Analyzing Business Information and Communication

Analyze your organization's existing business information and communication system to identify areas of compromised or weak security and to determine how you might improve your information security. The process includes the following:

- Defining information classification levels and the security policies for public, proprietary, and highly classified or sensitive information in the organization.
- Evaluating all of the information that your organization stores (such as files and databases) on your networks, including both extranets and the Internet.
- Evaluating all of the information that your organization sends over internal networks and over the Internet or extranets.

Defining Communication and Information Security Goals

After you analyze your organization's existing business information and communication, define realistic security goals for the organization. This is an important step in controlling the overall cost of the security measures that you eventually implement. Realistic security goals help ensure that you are providing acceptable levels of security at acceptable costs.

Following are examples of realistic communication and information security goals:

- Provide strong network logon authentication and at the same time reduce the Help desk costs that are associated with supporting users who forget their passwords or who let their passwords expire.
- Provide increased Internet security by preventing users from downloading or from using nontrusted and nonsigned content from the Internet.
- Provide increased intranet security by preventing users from downloading or from using nontrusted and nonsigned content from the intranet.
- Provide integrity and nonrepudiation for general business e-mail messages that are sent within your organization and enable users to send confidential e-mail messages as needed.
- Provide integrity, nonrepudiation, and confidentiality for all business e-mail messages between members of the executive management and trusted executive staff.

- Provide integrity, nonrepudiation, and confidentiality for all business e-mail messages that are sent over the Internet.
- Provide strong user authentication for project Web sites that are used for product development and project collaboration.
- Provide authentication, integrity, and confidentiality for online cost accounting transactions.
- Provide for a strong remote network logon process by using a single set of user network logon credentials to reduce the administrative overhead of maintaining separate local and remote network logon accounts for the same users.

Setting unrealistic security goals (for example, specifying an unnecessarily high level of security) can result in security requirements that cost too much to implement or maintain. Unrealistic security goals can also exceed the limits of existing technology and performance capabilities. For example, setting a goal to provide IP-level authentication, integrity, and confidentiality for all of your network communication might be achievable in a few years, but it is generally not feasible with today's network infrastructures and existing IPSec technology. IPSec can place a substantial load on network traffic, and many clients and applications do not yet support IPSec.

Determining Communication and Information Security Requirements

After you establish your security goals, determine what levels of security are required for specific types of information to meet your goals. This involves the following activities:

- Creating information and communication scenarios that reflect the types of information you must protect and the flow of information between entities on your networks.
- Determining the security levels that are required for each scenario.
- Defining the technical requirements for meeting the security goals.

For example, you might develop three different scenarios that reflect the information transfer in your company. One scenario might be private e-mail communication between executive staff members. A second scenario might be Web-based project collaboration and information sharing for classified product development projects. A third scenario might be network traffic and communication between members of the legal department.

In the first scenario, you might determine that communication between executive staff members requires secure, confidential e-mail messages that can be read only by the executive staff and a few other approved staff. You might also determine that the technology that is implemented must be strong and safe from attacks.

In the second scenario, you might determine that only approved members of the project team can have access to Web site content. You might also determine that some types of highly classified information require very secure protection when they are transmitted over the network.

In the third scenario, you might determine that all network traffic between computers in the legal department must be very securely protected. You might also determine that communication with computers outside the legal department is not to be allowed.

On the basis of the requirements that you identify during this process, you can choose security measures that meet your requirements at the acceptable level of risk and acceptable level of cost.

Assessing Security Risks

Estimate the risk of attacks on your network resources from both internal and external sources. The risk of attack and subsequent compromise of a system is affected primarily by the following factors:

- How secure your network is from external and internal attacks. If your networks are connected to the Internet, there is always some risk of external attack; if you lack adequate firewall and proxy services, the risk of external attack is very high. If workstations and servers are stored in secure locations, the risk of internal attack is usually relatively low. However, if unauthorized users have access to workstations or if network servers are not stored in secure data centers, the risk of internal attack is much higher.
- How valuable your network resources are to attackers. One network might be at high risk for attack because of the highly valuable financial information that is available on the network. Another network might be at low risk for attack because only public information is available. The value of an attack includes more than the monetary value of the resources on the network—an external attacker, for example, might place a high value on the prestige of breaking into your organization's networks.
- How high the cost of an attack is to the attackers. For example, the cost of launching an attack against a network that is secured by on-wire encrypted communication can be very high. However, the cost of launching an attack on a network where on-wire communication is in plaintext can be very low.

In general, only attackers that have cryptanalysis equipment and skills, a significant incentive to launch the attack, and considerable time to invest in the attack choose to pursue costly attacks on resources that are protected by strong cryptographic security technology.

To choose appropriate security solutions, you must weigh the risk of potential attacks and the potential damage of successful attacks against the costs and benefits of the security solutions you propose to deploy.

For more information about risk factors for cryptography-based security, see "Cryptography for Network and Information Security" in this book.

Assessing the Costs and Benefits of Security Solutions

The overall cost of a specific security solution includes the following:

- Resources necessary to plan, design, test, and deploy the system you choose.
- Resources necessary to administer and maintain the solution after deployment.
- Resources necessary to educate users about the new system and teach them how to use the new technology.
- Resources necessary to support users for the new technology (for example, the Help desk).
- Lost user productivity that is a result of the restrictions that are imposed by the security system.
- Increased load and reduced performance of computers and networks that result from the increased load of cryptographic operations.

Assess the costs of security solutions to determine which ones provide reasonable security benefits at acceptable costs and performance for your organization. The cost to implement and maintain security systems by using Windows 2000 distributed security technologies can vary considerably and depends on your security goals and requirements. The two examples that follow illustrate this.

If you deploy smart cards to implement smart card network logon authentication, you also have to deploy a smart card program to enroll users and to support users who lose or leave their smart cards at home. The smart card security policies you choose affect the costs of the program significantly. If employees who misplace their smart cards cannot quickly and easily obtain temporary access to the network, they lose productivity. However, if you allow employees to log on to the network without smart cards, network security is reduced. The smart card program and policies you choose affect user support costs, employee productivity, and overall network security.

If you use secure Web communications to provide strong confidentiality for information about classified projects, you can deploy and manage certificate services to issue and maintain digital certificates for all project members who need to use the classified Web sites. The cost associated with this system includes the resources that are required for implementing and maintaining certificate services. In addition, high volumes of secure Web traffic place a heavy performance burden on Web server processors. Therefore, you might have to install expensive cryptographic accelerator boards in your Web servers to achieve acceptable Web performance. The overall cost varies according to the number of users whom you need to support and the volume of confidential communication that is handled by your organization. If you can realistically keep the amount of confidential Web communication to a low volume, the additional cost of cryptographic accelerators might not be necessary.

Testing Your Proposed Security Plan

No matter what security solutions you choose, always test them in test laboratories and in pilot programs before you deploy them in your production environment. Testing ensures that you have designed the right system for the needs of your organization. Always test a system in simulations of the actual network environments that exist in your organization.

During testing, be sure the system not only performs properly but also provides the level of security that you expect. Test all features and functions of the system. Develop and use formal test plans; these help ensure that all security measures are, in fact, thoroughly tested.

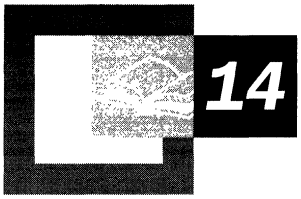
Many organizations now employ special security teams to launch real-world security attacks on their network security to discover and report weaknesses in their systems, in addition to testing their proposed systems in test laboratories and pilot programs. Through these efforts, organizations frequently discover and correct security weaknesses before real attackers have a chance to discover and exploit them. Consider implementing an ongoing program for field testing and improving your network and information security systems. Regular field testing can help you achieve a high level of assurance for the security of your information.

Additional Resources

- For more information about IETF drafts and recommendations, see the Internet Engineering Task Force (IETF) link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.
- For more information about Public Key Cryptography Standards (PKCS), see the RSA Data Security link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.
- For more information about the FIPS 140-1 standard, see "Security Requirements for Cryptographic Modules," National Institute of Standards and Technology (NIST) link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.
- For more information about FIPS 140-1 as a de facto international standard for cryptographic modules, see the International Organization for Standardization link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. Search for "International Standard 15408: Evaluation Criteria for Information Technology Security."
- For more information about the Microsoft® "Designed for Windows" Windows-compatible logo program and a list of currently compatible smart card products, see the Microsoft Windows Hardware Compatibility List link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.
- For more information about the security technologies in Microsoft products, including cryptography export restrictions and licensing requirements, see the Microsoft Security Advisor link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.
- For more information about the effect of export restrictions on the cryptography technologies you use for security solutions, see "Cryptography for Network and Information Security" in this book.

CHAPTER 14

Cryptography for Network and Information Security



Cryptography is a cornerstone of the modern electronic security technologies used today to protect valuable information resources on intranets, extranets, and the Internet. Microsoft® Windows® 2000 includes a wide range of distributed security technologies that you can deploy to provide cryptography-based network and information security. Understanding the basic concepts, components, and risks of cryptography-based security is important in choosing and planning appropriate network and information security systems for your organization.

In This Chapter

Related Information in the Resource Kit

- For information about security solutions that use public key technology, see "Choosing Security Solutions That Use Public Key Technology" in this book.
- For more information about the Windows 2000 public key infrastructure and Certificate Services, see "Windows 2000 Certificate Services and Public Key Infrastructure" in this book.
- For more information about designing, testing, and deploying a public key infrastructure, see "Planning Your Public Key Infrastructure" in the *Microsoft® Windows® 2000 Server Resource Kit Deployment Planning Guide*.

What Is Cryptography?

Cryptography is the science of providing security for information. It has been used historically as a means of providing secure communication between individuals, government agencies, and military forces. Today, cryptography is a cornerstone of the modern security technologies used to protect information and resources on both open and closed networks.

Background

Traditionally, cryptography has been used to pass coded messages between parties to ensure communication secrecy. Cryptography systems have used processes, techniques, and mechanisms to provide for secure communications between authorized parties while preventing unauthorized parties from monitoring communications or counterfeiting messages.

In its simplest form, cryptography substitutes or transposes letters to create a coded message, traditionally called a *cipher*, which is used to transform a readable message called *plaintext* (also called cleartext) into an unreadable, scrambled, or hidden message called *ciphertext*. Only someone with a decoding key can convert the ciphertext back into its original plaintext. The originator of a coded message must share the decoding key in a secure manner with intended recipients who are authorized to know the contents of the coded message. If unauthorized parties can somehow intercept or figure out the decoding key, security is compromised because they can convert the ciphertext into plaintext and read the contents of the message.

Anyone who can intercept both the ciphertext of a coded message and the decoding key can read secret communications. Furthermore, anyone who knows the cipher and has the decoding key might be able to impersonate the originator and send false messages. Therefore, systems of cryptography must provide reliable methods for securely sharing decoding keys, while keeping them unavailable to unauthorized parties. Others might know the cipher used to code a message and might have access to the coded message, but only authorized recipients are able to decode the contents of the message with the shared decoding key. Systems of cryptography also include techniques and mechanisms for verifying that originators of coded messages are authentic as well as ways to ensure that messages have not been altered en route.

Objective of Modern Cryptography Systems

No system of cryptography, called a *cryptosystem,* can be considered absolutely unbreakable or beyond compromise. However improbable a successful attack might seem, there is always some facet of the cryptosystem that can be attacked. The history of cryptography is full of examples of cryptosystems that were once considered invincible, and yet people were able to break the security and compromise them. Because cryptographers are not omniscient, they cannot design cryptosystems that are guaranteed to have no weaknesses or that are impervious to unforeseeable methods of attack. Furthermore, cryptosystems must be implemented in the real word, so they are subject to real-world limitations and constraints. All information security systems, including cryptography-based security, have weak links that can be attacked and potentially exploited to compromise the system.

The objective of modern cryptosystems is not to provide perfect or risk-free security. Rather, the objective of cryptography-based security is to protect information resources by making unauthorized acquisition of the information or tampering with the information more costly than the potential value that might be gained. Because the value of information usually decreases over time, good cryptography-based security protects information until its value is significantly less than the cost of illicit attempts to obtain or tamper with the information. Good cryptography, when properly implemented and used, makes attempts to violate security cost-prohibitive.

For example, many modern cryptosystems make it very difficult but not impossible for an attacker to determine the decoding key. Even though the key might be eventually determined by a skilled decoder, given enough time and effort, cryptosystems can still provide ample security to protect valuable information. By the time an attacker can feasibly determine the right decoding key, the cost of the attack would be much higher than the value of the information that is being protected by the key.

For well-designed and analyzed cryptosystems with no known flaws, the primary defense against attack is the length of the encoding key. All cryptosystems with encoding keys shorter than the plaintext message are subject to exhaustive search attacks where the attacker tries all possible combinations of the key until the key is found. For large encoding keys, an exhaustive search for the key usually requires sophisticated and expensive computing equipment to conduct the search, and the search can take hundreds, thousands, or even trillions of years to complete. Cryptosystems can defend against exhaustive key search attacks by simply making the encoding key long enough to make the search computationally infeasible or cost-prohibitive.

Another objective of all information security systems, including cryptography-based security systems, is to protect information resources at less cost than the value of the information that is being protected. A cryptography-based security system must provide information security at acceptable costs. Determining acceptable costs involves weighing the cost of the security versus the benefits of the security. For more information about evaluating the costs and benefits of security solutions, see "Choosing Security Solutions That Use Public Key Technology" in this book.

Security Functions of Cryptography

Cryptography is most often associated with the confidentiality of information that it provides. However, cryptography can offer the following four basic functions:

Confidentiality Assurance that only authorized users can read or use confidential information. Without confidentiality, anyone with network access can use readily available tools to eavesdrop on network traffic and intercept valuable proprietary information. Intruders who gain illicit network rights and permissions can steal proprietary information that is transmitted or stored as plaintext. Therefore, cryptosystems use techniques and mechanisms to ensure information confidentiality. For example, unauthorized users might be able to intercept information, but the information is transmitted and stored as ciphertext and is useless without a decoding key that is known only to authorized users.

Authentication Verification of the identity of the entities that communicate over the network. Without authentication, anyone with network access can use readily available tools to forge originating Internet Protocol (IP) addresses and impersonate others. Therefore, cryptosystems use various techniques and mechanisms to authenticate both the originators and recipients of information. For example, online entities can choose to trust communications with other online entities based on the other entities ownership of valid digital authentication credentials.

Integrity Verification that the original contents of information have not been altered or corrupted. Without integrity, someone might alter information or information might become corrupted, and the alteration could be undetected. Therefore, many cryptosystems use techniques and mechanisms to verify the integrity of information. For example, an intruder might covertly alter a file, but change the unique digital thumbprint for the file, causing other users to detect the tampering by comparing the changed digital thumbprint to the digital thumbprint for the original contents.

Nonrepudiation Assurance that a party in a communication cannot falsely deny that a part of the actual communication occurred. Without nonrepudiation, someone can communicate and then later either falsely deny the communications entirely or claim that it occurred at a different time. For example, without nonrepudiation, an originator of information might falsely deny being the originator of that information. Likewise, without nonrepudiation, the recipient of a communication might falsely deny having received the communication.

To provide nonrepudiation, systems must provide evidence of communications and transactions, so that involved parties cannot easily refute the evidence. For example, someone might deny sending an e-mail message, but the messaging system adds a timestamp and digitally signs the message with the message originator's digital signature. Because the message contains a timestamp and a unique signature, there is strong evidence to identify both the message's originator and the date and time of origin. If the message originator later denies sending the message, the false claim is easily refuted. Likewise, to provide nonrepudiation for mail recipients, mail systems might generate mail receipts that are dated and signed by the recipients.

Cryptography-based security technologies commonly use one or more of these functions to provide network and information security. Additionally, the security functions provided by cryptosystems sometimes overlap. For example, cryptosystems that provide nonrepudiation often provide authentication as a byproduct. Your security goals and requirements determine which functions you need to provide.

What Cryptography Technology Cannot Do

It is important to understand what cryptography-based security technology can and cannot do. In general, cryptography-based security systems provide ample security when used properly within the capabilities and limitations of the cryptography technology. However, cryptography technology only provides part of the overall security for your networks and information. The overall strength of security systems depends on many factors, such as the suitability of the technology, adequate security procedures and processes, and how well people use the procedures, processes, and technology. To put it another way: security is only as good as the weakest link in the entire security system.

You might use the best security technology available to provide all of the security functions of cryptography online, but all your security efforts will be defeated if someone (perhaps an intruder or an employee) can easily walk into offices and obtain valuable proprietary information that has been printed out as plaintext hard copy. You should not rely on cryptography-based security technology to overcome other weaknesses and flaws in security systems.

For example, if you transmit valuable information as ciphertext over your networks to protect confidentiality, but store the information as plaintext on origination or recipient computers, the overall security is only as good as the security provided for the computers where the plaintext is stored. If you provide low levels of network security for the computers or maintain them in nonsecure facilities, your information security might be very low. However, if you protect the computers with strong firewalls and maintain them in secure facilities, overall information security is much higher.

Role of Cryptanalysis

A discipline related to cryptography is *cryptanalysis*, defined as the science of interpreting ciphertext. These two disciplines combined form the science of *cryptology*. The cryptographer's goal is to provide security for information by developing strong cryptosystems, while the cryptanalyst's goal is to discover weaknesses or flaws in cryptosystems and break the security provided by those systems. Professional cryptanalysts perform an important role in evaluating and corroborating the strength of cryptosystems. In fact, cryptosystems are generally not considered secure until they withstand significant cryptanalysis.

Cryptanalysts can use powerful computing equipment and a variety of procedures, processes, and techniques to launch attacks against cryptosystems. In fact, a good cryptanalyst can even determine plaintext from samples of ciphertext without even knowing the cipher that was used to produce it.

Cryptanalysis can also be used illegitimately for illicit gain. Knowledgeable intruders can use cryptanalysis techniques as part of their attacks against your cryptography-based security systems. When properly implemented, standard cryptography-based security technologies can provide ample protection against a wide range of attacks, including common cryptanalysis techniques. However, to obtain highly valuable information, skilled intruders or trained espionage agents with access to powerful computing resources might have the incentive to launch expensive and highly sophisticated cryptanalyst attacks. Stopping sophisticated cryptanalyst attacks requires highly secure systems that use strong cryptography-based security technologies.

Basic Components of Modern Cryptography

Modern electronic cryptosystems use complex mathematical algorithms and other techniques and mechanisms to provide network and information security. Cryptography-based security technologies commonly use one or more of the following basic components to provide security functions:

- Encryption algorithms
- Message digest functions
- Hashed Message Authentication Code (HMAC) functions
- Secret key exchange algorithms
- Digital signatures

Windows 2000 distributed security technologies use these components of cryptology for a variety of security functions. For more information about how these components are used with security technologies in Windows 2000, see the appropriate sections of the following publications:

- *Microsoft® Windows® 2000 Professional Resource Kit*
- *Microsoft® Windows® 2000 Server Resource Kit*
- Windows 2000 Server Help
- *Microsoft® Platform Software Development Kit*

Encryption

Traditionally, ciphers have used information contained in secret decoding keys to code and decode messages. The process of coding plaintext to create ciphertext is called *encryption,* and the process of decoding ciphertext to produce the plaintext is called *decryption.* Modern systems of electronic cryptography use *digital keys* (bit strings) and mathematical algorithms (*encryption algorithms*) to encrypt and decrypt information.

There are two types of encryption: symmetric key encryption and public (asymmetric) key encryption. Symmetric key and public key encryption are used, often in conjunction, to provide a variety of security functions for network and information security.

Symmetric Key Encryption

Encryption algorithms that use the same key for encrypting and for decrypting information are called *symmetric-key algorithms.* The symmetric key is also called a secret key because it is kept as a shared secret between the sender and receiver of information. Otherwise, the confidentiality of the encrypted information is compromised. Figure 14.1 shows basic symmetric key encryption and decryption.

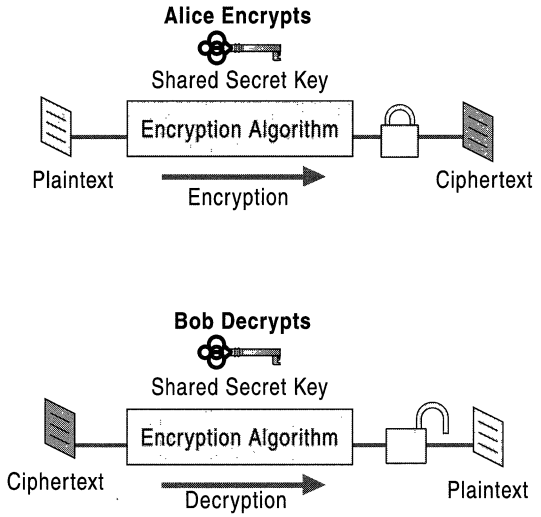


Figure 14.1 Encryption and Decryption with a Symmetric Key

Symmetric key encryption is much faster than public key encryption, often by 100 to 1,000 times. Because public key encryption places a much heavier computational load on computer processors than symmetric key encryption, symmetric key technology is generally used to provide secrecy for the bulk encryption and decryption of information.

Symmetric keys are commonly used by security protocols as *session keys* for confidential online communications. For example, the Transport Layer Security (TLS) and Internet Protocol security (IPSec) protocols use symmetric session keys with standard encryption algorithms to encrypt and decrypt confidential communications between parties. Different session keys are used for each confidential communication session and session keys are sometimes renewed at specified intervals.

Symmetric keys also are commonly used by technologies that provide bulk encryption of persistent data, such as e-mail messages and document files. For example, Secure/Multipurpose Internet Mail Extensions (S/MIME) uses symmetric keys to encrypt messages for confidential mail, and Encrypting File System (EFS) uses symmetric keys to encrypt files for confidentiality.

Cryptography-based security technologies use a variety of symmetric key encryption algorithms to provide confidentiality. For more information about the specific encryption algorithms that are used by security technologies, see the applicable documentation for each technology. For more information about how the various symmetric key algorithms differ, see the cryptography literature that is referenced under "Additional Resources" at the end of this chapter.

Public Key Encryption

Encryption algorithms that use different keys for encrypting and decrypting information are most often called *public-key algorithms* but are sometimes also called *asymmetric key algorithms*. Public key encryption requires the use of both a *private key* (a key that is known only to its owner) and a *public key* (a key that is available to and known to other entities on the network). A user's public key, for example, can be published in the directory so that it is accessible to other people in the organization. The two keys are different but complementary in function. Information that is encrypted with the public key can be decrypted only with the corresponding private key of the set. Figure 14.2 shows basic encryption and decryption with asymmetric keys.

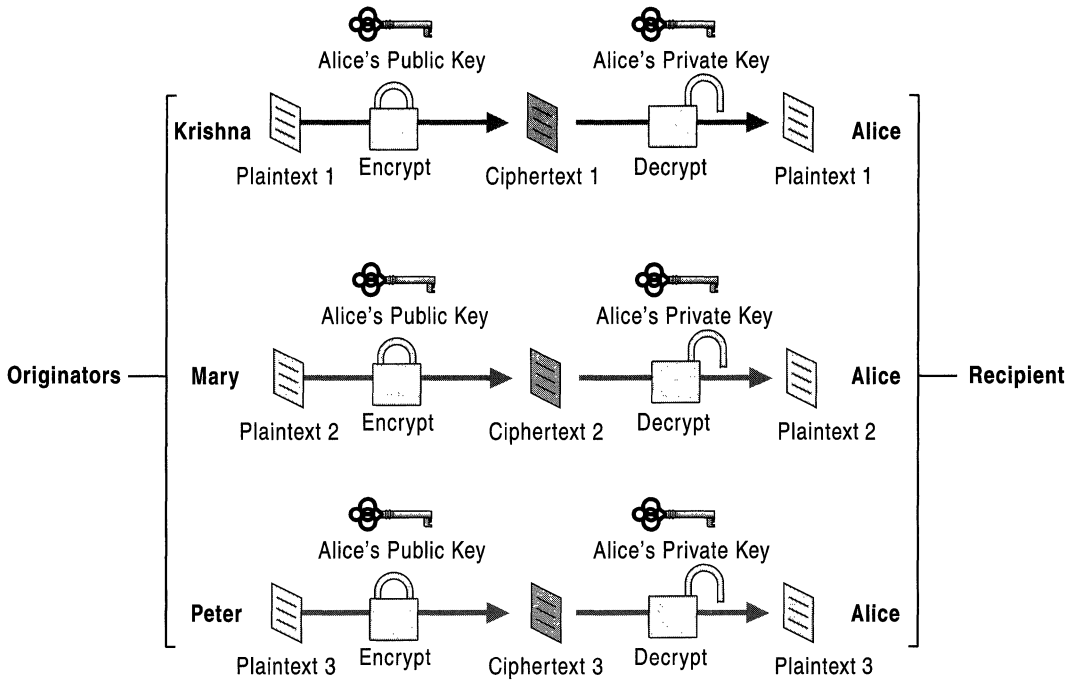


Figure 14.2 Encryption and Decryption with Asymmetric Keys

The RSA digital signature process also uses private keys to encrypt information to form digital signatures. For RSA digital signatures, only the public key can decrypt information encrypted by the corresponding private key of the set.

Today, public key encryption plays an increasingly important role in providing strong, scalable security on intranets and the Internet. Public key encryption is commonly used to perform the following functions:

- Encrypt symmetric secret keys to protect the symmetric keys during exchange over the network or while being used, stored, or cached by operating systems.
- Create digital signatures to provide authentication and nonrepudiation for online entities.
- Create digital signatures to provide data integrity for electronic files and documents.

Message Digest Functions

Message digest functions, also called *hash functions*, are used to produce digital summaries of information called message digests. *Message digests* (also called *hashes*) are commonly 128 bits to 160 bits in length and provide a digital identifier for each digital file or document. Message digest functions are mathematical functions that process information to produce a different message digest for each unique document. Identical documents have the same message digest; but if even one of the bits for the document changes, the message digest changes. Figure 14.3 shows the basic message digest process.

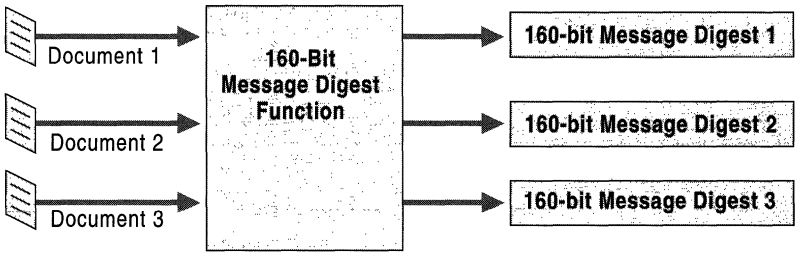


Figure 14.3 Example of the Message Digest Process

Because message digests are much shorter than the data from which the digests are generated and the digests have a finite length, duplicate message digests called *collisions* can exist for different data sets. However, good message digest functions use one-way functions to ensure that it is mathematically and computationally infeasible to reverse the message digest process and discover the original data. Finding collisions for good message digest functions is also mathematically and computationally infeasible but possible given enough time and computational effort. However, even if an attacker discovers a collision, it is highly improbable that the collision could be useful. For example, assume that an English message produces a message digest with a value of *n,* and an attacker somehow manages to computationally generate a second set of data that also produces a message digest of *n*. The second set of data would have to be in the English language and form a coherent and germane message for an attacker to be able to use it for an illicit purpose, such as sending a counterfeit message in the place of the original message. With the best message digest functions in use today, the probability that a second set of collision data would be in a known language or form a coherent message is minuscule.

Message digests are commonly used in conjunction with public key technology to create digital signatures or "digital thumbprints" that are used for authentication, integrity, and nonrepudiation. Message digests also are commonly used with digital signing technology to provide data integrity for electronic files and documents.

For example, to provide data integrity for e-mail messages, message digests can be generated from the completed mail message, digitally signed with the originator's private key, and then transmitted with e-mail messages. The recipient of the message can then do the following to check the integrity of the message:

- Use the same message digest function to compute a digest for the message.
- Use the originator's public key to verify the signed message digest.
- Compare the new message digest to the original digest.

If the two message digests do not match, the recipient knows the message was altered or corrupted. Figure 14.4 shows a basic integrity check process with a digitally signed message digest.

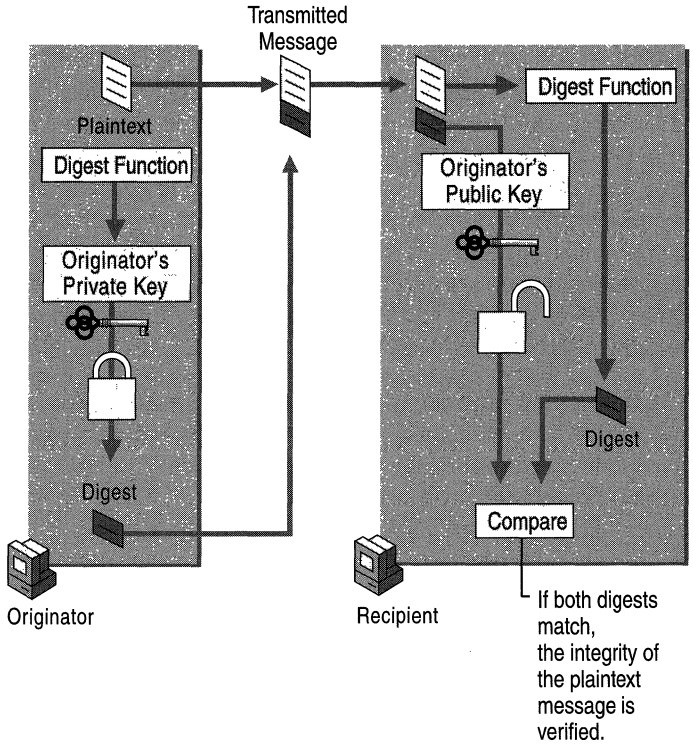


Figure 14.4 Example of an Integrity Check with a Digitally Signed Message Digest

Because the message digest is digitally signed with the sender's private key, it is not feasible for an intruder to intercept the message, modify it, and create a new valid encrypted message digest to send to the recipient. Another method of ensuring the integrity of data is to use message digests with a Hashed Message Authentication Code (HMAC) function, as described later in this chapter.

Two of the most commonly used message digest algorithms today are MD5, a 128-bit digest developed by RSA Data Security, Inc., and SHA-1, a 160-bit message digest developed by the National Security Agency. The SHA-1 algorithm is generally considered to provide stronger cryptographic security than MD5, because it uses a longer message digest and it is not vulnerable to some attacks that can be conducted against MD5.

Hashed Message Authentication Code Functions

A Hashed Message Authentication Code (HMAC) is a message authentication function for Internet communications described in RFC 2104 of the Network Working Group of the Internet Engineering Task Force (IETF). HMAC uses standard message digest functions, such as MD5 and SHA-1. In general, HMAC MD5 provides better performance for secure communications, while HMAC SHA-1 provides stronger cryptographic security.

HMAC is widely used by Internet security technologies, such as the TLS and IPSec protocols, to verify the integrity of transmitted data during secure communications. HMAC generates a message digest for each block of transmitted data and uses a random secret symmetric key to encrypt the message digests. The secret key is securely shared between the parties involved in the secure communications. (Secure secret key exchange is done with key exchange algorithms, which are described later in this chapter.) When data is received, the secret key is necessary to decrypt the message digest and perform the data integrity check.

The cryptographic strength of the HMAC depends on the underlying strength of the message digest used and how securely the secret key is exchanged. An intruder does not know the secret key and cannot tamper with the data en route or counterfeit the message digest. HMAC provides data integrity and protection against tampering in a manner similar to digital signatures, but it does not require communicating parties to have public and private keys. HMACs also provide better performance for bulk online communications than public key digital signing technologies.

Secret Key Exchange

For symmetric key cryptography to work for online communications, the secret key must be securely shared with authorized communicating parties and protected from discovery and use by unauthorized parties. Public key cryptography can be used to provide a secure method for exchanging secret keys online. Two of the most common key exchange algorithms are the following:

- Diffie-Hellman Key Agreement algorithm
- RSA key exchange process

Both methods provide for highly secure key exchange between communicating parties. An intruder who intercepts network communications cannot easily guess or decode the secret key that is required to decrypt communications. The exact mechanisms and algorithms that are used for key exchange varies for each security technology. In general, the Diffie-Hellman Key Agreement algorithm provides better performance than the RSA key exchange algorithm.

Diffie-Hellman Key Agreement

Public key cryptography was first publicly proposed in 1975 by Stanford University researchers Whitfield Diffie and Martin Hellman to provide a secure solution for confidentially exchanging information online. Figure 14.5 shows the basic *Diffie-Hellman Key Agreement* process.

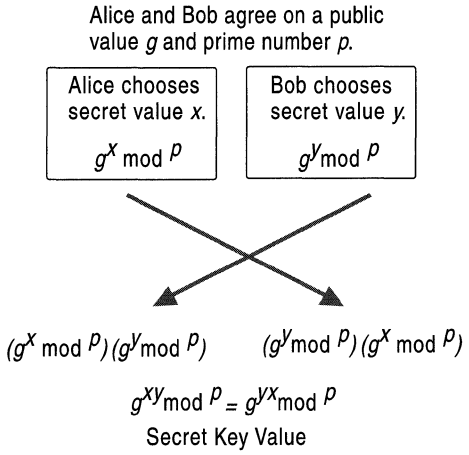


Figure 14.5 Diffie-Hellman Key Agreement

Diffie-Hellman key agreement is not based on encryption and decryption, but instead relies on mathematical functions that enable two parties to generate a shared secret key for exchanging information confidentially online. Essentially, each party agrees on a public value g and a large prime number p. Next, one party chooses a secret value x and the other party chooses a secret value y. Both parties use their secret values to derive public values, $g^x \bmod p$ and $g^y \bmod p$, and they exchange the public values. Each party then uses the other party's public value to calculate the shared secret key that is used by both parties for confidential communications. A third party cannot derive the shared secret key because they do not know either of the secret values, x or y.

For example, Alice chooses secret value x and sends the public value $g^x \bmod p$ to Bob. Bob chooses secret value y and sends the public value $g^y \bmod p$ to Alice. Alice uses the value $g^{xy} \bmod p$ as her secret key for confidential communications with Bob. Bob uses the value $g^{yx} \bmod p$ as his secret key. Because $g^{xy} \bmod p$ equals $g^{yx} \bmod p$, Alice and Bob can use their secret keys with a symmetric key algorithm to conduct confidential online communications. The use of the modulo function ensures that both parties can calculate the same secret key value, but an eavesdropper cannot. An eavesdropper can intercept the values of g and p, but because of the extremely difficult mathematical problem created by the use of a large prime number in mod p, the eavesdropper cannot feasibly calculate either secret value x or secret value y. The secret key is known only to each party and is never visible on the network.

Diffie-Hellman key exchange is widely used with varying technical details by Internet security technologies, such as IPSec and TLS, to provide secret key exchange for confidential online communications. For technical discussions about Diffie-Hellman key agreement and how it is implemented in security technologies, see the cryptography literature that is referenced under "Additional Resources" at the end of this chapter.

RSA Key Exchange

The Rivest-Shamir-Adleman (RSA) algorithms available from RSA Data Security, Inc., are the most widely used public key cryptography algorithms. For RSA key exchange, secret keys are exchanged securely online by encrypting the secret key with the intended recipient's public key. Only the intended recipient can decrypt the secret key because it requires the use of the recipient's private key. Therefore, a third party who intercepts the encrypted, shared secret key cannot decrypt and use it. Figure 14.6 illustrates the basic RSA key exchange process.

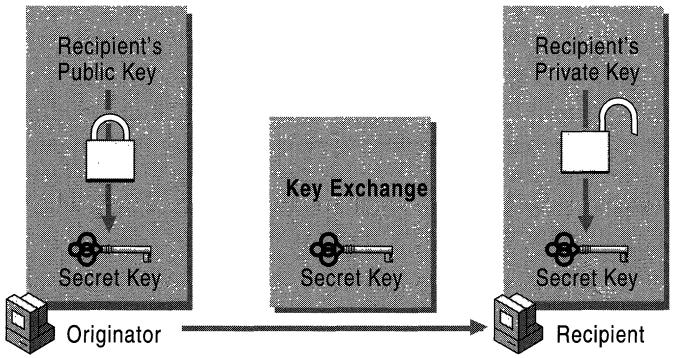


Figure 14.6 Basic RSA Key Exchange

The RSA key exchange process is used by some security technologies to protect encryption keys. For example, EFS uses the RSA key exchange process to protect the bulk encryption keys that are used to encrypt and decrypt files.

Digital Signatures

Just as handwritten signatures or physical thumbprints are commonly used to uniquely identify people for legal proceedings or transactions, so digital signatures ("digital thumbprints") are commonly used to identify electronic entities for online transactions. A *digital signature* uniquely identifies the originator of digitally signed data and also ensures the integrity of the signed data against tampering or corruption.

One possible method for creating a digital signature is for the originator of data to create the signature by encrypting all of the data with the originator's private key and enclosing the signature with the original data. Anyone with the originator's public key can decrypt the signature and compare the decrypted message to the original message. Because only someone with the private key can create the signature, the integrity of the message is verified when the decrypted message matches the original. If an intruder alters the original message during transit, the intruder cannot also create a new valid signature. If an intruder alters the signature during transit, the signature does not verify properly and is invalid.

However, encrypting all data to provide a digital signature is impractical for three reasons:

- The ciphertext signature is the same size as the corresponding plaintext, so message sizes are doubled, consuming large amounts of bandwidth and storage space.
- Public key encryption is slow and places heavy computational loads on computer processors, so network and computer performance can be significantly degraded.
- Encrypting the entire contents of information produces large amounts of ciphertext, which can be used for cryptanalysis attacks, especially known plaintext attacks (where certain parts of the encrypted data, such as e-mail headers, are known beforehand to the attacker).

Digital signature algorithms use more efficient methods to create digital signatures. The most common types of digital signatures today are created by signing message digests with the originator's private key to create a digital thumbprint of the data. Because only the message digest is signed, the signature is usually much shorter than the data that was signed. Therefore, digital signatures place a relatively low load on computer processors during the signing process, consume insignificant amounts of bandwidth, and produce small amounts of ciphertext for cryptanalysis. Two of the most widely used digital signature algorithms today are the RSA digital signature process and the Digital Signature Algorithm (DSA).

RSA Data Security Digital Signature Process

In the RSA digital signature process, the private key is used to encrypt only the message digest. The encrypted message digest becomes the digital signature and is attached to the original data. Figure 14.7 illustrates the basic RSA Data Security digital signature process.

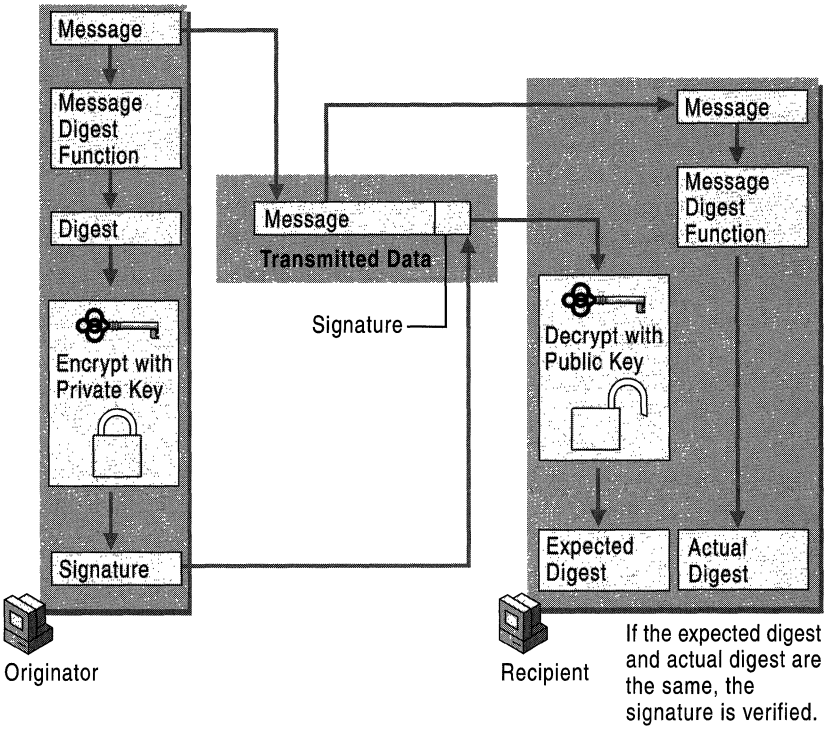


Figure 14.7 Basic RSA Data Security Digital Signature Process

To verify the contents of digitally signed data, the recipient generates a new message digest from the data that was received, decrypts the original message digest with the originator's public key, and compares the decrypted digest with the newly generated digest. If the two digests match, the integrity of the message is verified. The identify of the originator also is confirmed because the public key can decrypt only data that has been encrypted with the corresponding private key.

Digital Signature Security Standard

Another widely used technology for creating digital signatures is the Digital Signature Security Standard (DSS) that was developed by the National Security Agency and adopted by the United States government as its digital-signature standard. DSS defines the Digital Signature Algorithm (DSA), which functions in a manner similar to RSA. Although similar to RSA, DSA does not encrypt message digests with the private key or decrypt the message digest with the public key. Instead, DSA uses special mathematical functions to generate a digital signature composed of two 160-bit numbers that are derived from the message digest and the private key. DSA uses the public key to verify the signature, but the verification process is more complex than RSA.

The digital signature processes for DSA and RSA are generally considered to be of equal strength. However, DSA requires the use of the SHA-1 message digest function to ensure strong digital signatures. RSA can be used with other message digest functions (besides SHA-1) that might produce weaker digital signatures. Because the DSA signature verification process increases computer processor load significantly, relative to the verification process for RSA (all other conditions being equal), the RSA digital signature process generally provides better overall performance.

Because DSA is used only for digital signatures and makes no provisions for data encryption (for example, to provide secure secret key exchange), DSA is usually not subject to the export or import restrictions commonly imposed on RSA cryptography technology. Therefore, DSS digital signature technology can often be used when RSA digital signature technology cannot be used because of government-imposed export or import restrictions.

Uses for Digital Signatures

Anyone with the public key can use it to perform a validity check of digital signatures created by the private key. Only a digital signature created by the appropriate private key decrypts and validates properly with the public key. If a different private key was used to sign the data, the validity check fails. If the contents of digitally signed data or the digital signature have been tampered with or are corrupted, the validity check also fails. Valid digital signatures can be used to perform the following functions:

- Authenticate online entities.
- Verify the authorship or origin of digital data.
- Ensure the integrity of digital data against tampering.

Many security technologies use digital signatures. For example, Microsoft® Authenticode® can be used to digitally sign software programs, safeguarding them when they are distributed on the intranet or Internet to help counter the threat of software tampering and the spread of viruses and other malicious code. Likewise, the S/MIME protocol can be used to digitally sign e-mail messages to ensure the integrity of mail communications.

Basic Components of a Public Key Infrastructure

A public key infrastructure (PKI) provides the framework of services, technology, protocols, and standards that enable you to deploy and manage a strong and scalable information security system based on public key technology. The basic components of a public key infrastructure include digital certificates, certificate revocation lists, and certification authorities. Before public key cryptography can be widely used and easily managed on public networks, a public key infrastructure must be in place. Without a public key infrastructure, public key technology is not generally suitable for large-scale enterprise deployment.

This section describes the basic concepts and components of public key infrastructures that are based on the open standards recommended by the Public-Key Infrastructure (X.509) (PKIX) working group of the Internet Engineering Task Force (IETF). PKIX-compliant public key infrastructures can provide a high level of interoperability between public key security products from different vendors. For more information about the components and technology used in the Windows 2000 public key infrastructure, see "Windows 2000 Certificate Services and Public Key Infrastructure" in this book. For more information about the public key infrastructure recommendations of the PKIX working group, see the Internet Engineering Task Force (IETF) link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Ways to Establish Trust for Private Key and Public Key Sets

Public key cryptography uses public keys and private keys to provide authentication, confidentiality, integrity, and nonrepudiation. However, by themselves, public keys and private keys cannot provide proof that the public key and private key set belong to an alleged key set owner. There has to be a way to verify the identity of the owner of a public key and private key set with a high degree of confidence. There also has to be a way to establish trust for the public key and private key sets of geographically dispersed entities on intranets and the Internet.

On a public network, Alice can send a message to Bob, which is signed by her private key and accompanied by the corresponding public key. Bob can then use the public key to verify the integrity of the message, but how can he verify that the sender is really Alice? Perhaps an intruder has intercepted Alice's original message and substituted a counterfeit message to Bob along with a counterfeit public key. The intruder might also be impersonating Alice by forging her network IP address. In addition, if Bob receives a message from Alice, how does he know he can trust her? How does he know that she won't falsely deny sending the message?

Peer-to-Peer Trust

For Bob to trust Alice's private key and public key, he needs assurance of her identity and verification that the public key is indeed hers. If Bob knows Alice and obtains her public key in person, he can use her public key to verify digital signatures created by her private key with a high level of trust and assurance. If an intruder attempts to impersonate Alice, Bob can use her public key to detect whether the message is signed by her private key and whether it was altered. However, such peer-to-peer trust is generally limited to a small group of people who work within the same locality or who know each other well. Peer-to-peer trust is effective for a small circle of people who choose to trust each other's public keys, but it is not scalable to the public Internet or to large and often geographically dispersed enterprises.

Trust in Certification Authorities

To establish widespread trust for private key and public key sets on open networks, there must be trustworthy authorities that can certify the identities of individuals, organizations, and computers on the network and also provide assurance that public keys and private keys correspond to these entities. These trustworthy authorities, called certification authorities (CAs), positively verify the identity of each online entity and provide credentials to identify online identities, so that others can have a high level of assurance that the online entities are who they claim to be.

To establish trust on the Internet, intranets, or extranets, a public key infrastructure uses electronic credentials called digital certificates that are issued by CAs. A digital certificate provides proof that the entity named in the certificate is the owner of the public key and private key set. Therefore, other entities on the network can have a high level of assurance that a public key really belongs to the owner of the private key.

If Bob trusts a CA on the basis of its reputation, he can then choose to trust Alice's public key when it is certified by the CA, even if he does not know Alice. Bob can be confident that Alice is who she claims to be during online communication. Furthermore, Bob can be confident that Alice cannot falsely deny sending a message that was signed with her private key.

PKIX-compliant public key infrastructures, such as the public key infrastructure in Windows 2000, use digital certificates issued by CAs rather than peer-to-peer trust to establish trust on intranets, extranets, and the Internet for online entities and their public key and private key sets.

Digital Certificates

Digital certificates, similar to identification cards, are electronic credentials that are used to certify the online identities of individuals, organizations, and computers. Certificates are issued and certified by CAs. PKIX-compliant public key infrastructures support industry standard X.509 version 3 certificates.

Functions Like a Traditional Identification Card

Digital certificates function similarly to identification cards such as passports and drivers' licenses. Identification cards are issued by recognized government authorities. When someone requests an identification card, a government authority verifies the identity of the requester, certifies that the requester meets all requirements to receive the card, and then issues the card. When an identification card such as a driver's license is presented to others, they can verify the identify of its owner because the card provides the following security benefits:

- It contains personal information to help identify and trace the owner.
- It contains the photograph and the signature of the rightful owner to enable positive identification.
- It contains the information that is required to identify and contact the issuing authority.
- It is designed to be tamper resistant and difficult to counterfeit.
- It is issued by an authority that can revoke the identification card at any time (for example, if the card is misused or stolen).
- It can be checked for revocation by contacting the issuing authority.

Issued by Certification Authorities

Like a driver's license, digital certificates are issued by CAs to provide proof for verifying the identity of online entities. However, instead of containing a photograph and the signature of the certificate's owner, a certificate binds the owner's public key to the owner's private key.

A certificate contains information that identifies the certificate's owner (called the subject) as an entity on the network. A certificate also contains the owner's public key. Furthermore, a certificate identifies the CA (called the issuer) that issued the certificate. A CA uses its private key to digitally sign each certificate it issues. To create the digital signature, the CA generates a message digest from the certificate, encrypts the digest with its private key, and includes the digital signature as part of the certificate. Anyone can use the message digest function and the CA's public key to verify the certificate's integrity. If a certificate becomes corrupted or someone tampers with it, the message digest for the altered certificate does not match the digest in the CA's digital signature. Figure 14.8 shows how a certificate is signed by the issuing CA.

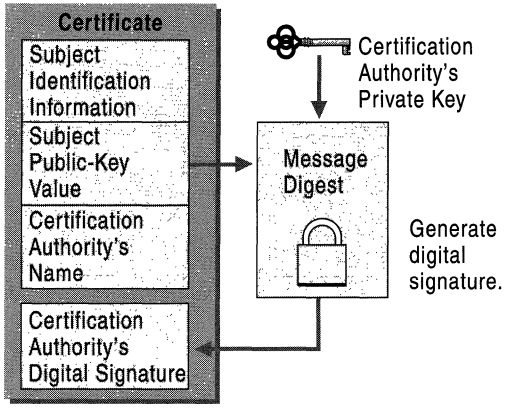


Figure 14.8 Digital Signature for a Certificate

A certificate is public information that is available to anyone. Certificates are commonly distributed by means of directories, public folders, e-mail, and Web pages. Because the certificate owner's public key is contained in a certificate, distributing a certificate also distributes the public key. Others can choose to trust a certificate owner's private key based on the reputation of the CA that issued the certificate and based on confidence in the certificate issuing practices of the CA.

Contents of X.509 Version 3 Certificates

PKIX-compliant public key infrastructures, including the public key infrastructure in Windows 2000, support X.509 version 3 certificates. Figure 14.9 shows the contents of X.509 version 3 certificates.

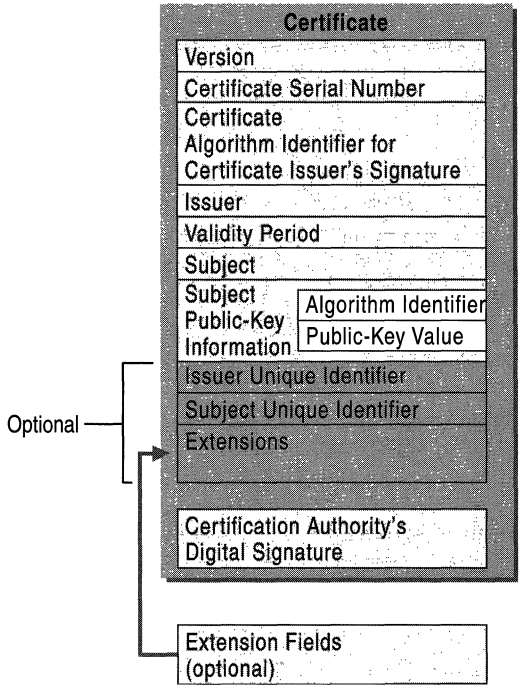


Figure 14.9 X.509 Version 3 Certificate

The contents of X.509 version 3 certificates are described in Table 14.1.

Table 14.1 Description of X.509 Version 3 Certificate Contents

Certificate Field	Description
Version	Version of the certificate format; for example, version 3.
Certificate Serial Number	The unique serial number that is assigned by the issuing CA. The CA maintains an audit history for each certificate so that certificates can be traced by their serial numbers. Revoked certificates also can be traced by their serial numbers.

(continued)

Table 14.1 Description of X.509 Version 3 Certificate Contents *(continued)*

Certificate Field	Description
Certificate Algorithm Identifier	The public key cryptography and message digest algorithms that are used by the issuing CA to digitally sign the certificate.
Issuer	The name of the issuing CA. The name can be listed in one or more of the following formats: X.500 directory name, Internet e-mail address, fully qualified domain name (FQDN), X.400 e-mail address, and URL.
Certificate Algorithm Identifier	The public key cryptography and message digest algorithms that are used by the issuing CA to digitally sign the certificate.
Issuer	The name of the issuing CA. The name can be listed in one or more of the following formats: X.500 directory name, Internet e-mail address, fully qualified domain name (FQDN), X.400 e-mail address, and URL.
Validity Period	The certificate's start and expiration dates. These define the interval during which the certificate is valid, although the certificate can be revoked before the designated expiration date.
Subject	The name of the subject (owner) of the certificate. The name can be listed in one or more of the following formats: X.500 directory name, Internet e-mail address, fully qualified domain name (FQDN), X.400 e-mail address, and URL.
Subject Public-Key Information	The public key and a list of the public key cryptography algorithms. The algorithms are for the tasks for which the public key set can be used, such as digital signing, secret key encryption, and authentication.
Issuer Unique Identifier	Optional information for uniquely identifying the issuer, when necessary.
Subject Unique Identifier	Optional information for uniquely identifying the subject, when necessary.
Extensions	Additional information that can be specified for optional use by public key infrastructures. Common extensions include a list of specific uses for certificates (for example, S/MIME secure mail or IPSec authentication), CA trust relationship and hierarchy information, a list of publication points for revocation lists, and a list of additional attributes for the issuer and subject.
Certification Authority's Digital Signature	The CA's digital signature, which is created as the last step in generating the certificate.

Uses of the Public Key and Private Key Set

X.509 version 3 certificates contain information in the Subject Public-Key Information field that specifies the cryptography operations for which the public key and private key set can be used. Public key security systems commonly support the following basic cryptography operations:

- Digital signing of electronic data to verify data origin and the integrity of data.
- Authentication of entities that are communicating over networks.
- Secret key encryption to protect symmetric secret encryption transmitted and shared over networks.

The public key and private key set can be used to provide a variety of specific security functions for information security technologies. These specific functions of certificates are listed in the Extensions field. Common specific security functions for public key technology include the following:

- Secure mail to provide authentication, confidentiality, integrity, and nonrepudiation for e-mail communications.
- Secure Web communications to provide authentication, integrity, and confidentiality between Web clients and servers.
- Code signing to provide integrity and nonrepudiation for executable code to be distributed on the Internet or intranets.
- Local network logon or remote access logon to authenticate users of network resources.
- IPSec authentication to authenticate clients that do not use Kerberos authentication or shared secret passwords for IPSec communications.

Certificate Management

A public key infrastructure provides the foundation, components, and features that are necessary for managing certificates and private keys throughout the certificate lifecycle. The certificate lifecycle includes the following:

- Certificate issue
- Certificate revocation
- Certificate renewal
- Certificate expiration

To manage the certificate lifecycle, a public key infrastructure must provide mechanisms to support the following management activities:

- Enroll users and computers for certificates.
- Distribute certificates for public use.
- Publish certificate revocation lists (CRLs).
- Renew certificates.
- Maintain a certificate audit trail.

Certificate Enrollment

Users and computers must enroll to request and receive certificates from a CA. The enrollment process varies with the CA and its policies. One of the most common certificate enrollment methods is to use Web pages for certificate requesters to submit their certificate requests. When a request is received by a CA, the CA verifies whether the requester is qualified to receive the certificate and either approves or denies the request.

The certificate issuing policies for CAs vary depending on the level of verification the CA must do to verify the certificate requester's identity. Some certificate requests might be approved or denied quickly because the CA merely checks to see if the requester has a valid Internet e-mail address. Other certificate requests might take longer because the CA must verify the requester's identity through third-party agencies or by conducting background checks.

Some public key infrastructures, such as the Windows 2000 public key infrastructure, can automate the enrollment for certain types of certificates. For example, in Windows 2000, you can optionally configure Public Key Group Policy to automatically enroll Windows 2000 computers for computer certificates.

Certificate Distribution

When certificates are issued, they must be distributed to the requester as well as to distribution points where other users can have access to them, as necessary. Many public key infrastructures, including the Windows 2000 public key infrastructure, can be configured to automatically distribute certificates through directories, Web pages, public folders, and e-mail.

Certificate Revocation Lists

CAs publish certificate revocation lists (CRLs) to identify certificates that have been revoked (for example, when a certificate user has left the organization or when a private key has been compromised). During the certificate validation process, software can check the CRL to determine whether the certificate is invalid. Certificates that are listed in CRLs are invalid and should not be trusted. When a revoked certificate expires, it is no longer published in the CRL. The CA uses its private key to digitally sign CRLs to prevent tampering with them.

Many public key infrastructures, including the Windows 2000 public key infrastructure, can be configured to periodically publish CRLs. CRLs can be distributed through directories, Web pages, public folders, and e-mail. The X.509 version 3 certificate format includes a field that lists the distribution points for the CRLs published by the issuing CA.

Certificate Renewal

When a certificate reaches its expiration date, the certificate is invalid and can no longer be used. However, certificates can be re-issued or renewed with new, valid dates. The renewal process is similar to the enrollment process.

Certificate Audit Trail

Each CA must maintain an audit trail of certificate requests and the certificates that are issued until they expire. The audit trail records all certificate transactions including failed requests and all of the information contained in each issued certificate. It also provides the information that is required to revoke a certificate and add it to the revocation list. CA administrators can query the audit trail to locate and view information about any certificate request or any certificate that has been issued by the CA.

An audit trail is necessary to meet the security obligations of the CA and the organization. The administrators of the CA must be able to provide records of all certificate transactions. For example, when an issued certificate was used for an illegal activity or for a fraudulent transaction, CA administrators might be asked to provide records to security or law enforcement personnel.

In addition, CA administrators need audit trail records to monitor the network for security breaches. For example, administrators can view the audit trail to detect failed certificate requests or to determine whether someone has improperly obtained certificates.

Key Management

Managing private keys is a crucial function of the public key infrastructure. Private keys must be securely generated, delivered to their owners, and stored. If attackers can obtain private keys, public key security is defeated. Anyone who can illicitly obtain a private key can impersonate the authorized owner of the private key during online public key cryptography operations. Depending on the security functions provided by the legitimate use of the private key, a compromised private key can be used to damage network resources, steal valuable information, and even damage reputations.

For example, if you control access to a secure Web site based on authentication certificates, someone with a copy of a private key can gain access to confidential information intended for the authorized user of the private key. If you grant the owners of private keys read/write permission on your Web site, an imposter with a copy of a private key can destroy or modify files on the Web site.

In addition, someone with an illicit copy of your private key for secure mail, can send counterfeit mail on your behalf. Likewise, someone with an illicit copy of your organization's private key for software publishing can sign a malicious program and present it on the Internet as your software to mislead others into trusting and running the malicious program.

Security for Private Keys

Only the owner of a private key must have possession of the key or use the key. Therefore, a public key infrastructure must store private keys in a protected place that is inaccessible to others, and no other copies of the key should normally exist. Public key infrastructures, including the Windows 2000 public key infrastructure, commonly manage private keys with secure interfaces and keep them in protected stores to ensure the ongoing confidentiality of private keys.

Key management technology can be software-based or hardware-based. In general, hardware-based key management provides greater security. For example, cryptosystems that store and manage keys on physical tokens such as smart cards usually provide higher security for keys than cryptosystems that manage and store keys with software.

Key Recovery

Sometimes a key recovery database or archive is used to securely store copies of nonsigning private keys that are used by security technologies such as S/MIME and EFS to encrypt persistent data. For S/MIME and EFS, a user's public key encrypts and protects the secret symmetric key that was used to encrypt bulk data. The user's private key must be used to decrypt the bulk encryption key, so that it can be used to decrypt the data. If the private key is lost or damaged and a key recovery database exists, a copy of the private key can be retrieved from a protected store for decrypting the symmetric key and recover encrypted data.

Important Private keys that are used for digital signing and authentication must never be stored in a key recovery database or archive. Someone other than the legitimate key owner might be able to gain access to the duplicate and impersonate the owner. The mere existence of a copy of a signing key compromises the authentication, integrity, or nonrepudiation provided by the key.

When recovery databases are properly used, only trusted administrators are allowed to use the database and restore keys to their owners. To prevent key compromise, the confidentiality of private keys must be maintained at all times during the key and data recovery processes.

In addition, some public key infrastructures or public key products might include a key escrow system that allows others besides the owners of private keys to access and use the keys. In a key escrow system, a third party holds copies of the private keys. Key escrow systems are sometimes required to obtain a cryptography export license. Key escrow systems are most commonly used when it is necessary for government officials to obtain the private keys that have been used to encrypt information.

Windows 2000 does not provide any key recovery or escrow services; however, some Microsoft or third-party products might support key recovery or key escrow services. For example, Microsoft Exchange Server provides key recovery services for secure mail so you can recover encrypted data if private keys are lost or damaged. Windows 2000, however, does support exporting nonsigning private keys that are used for technologies such as S/MIME and EFS, so you can manually archive keys and use them for data recovery. Such exported keys are stored in a password-protected encrypted format.

Certification Authorities in the Enterprise

Most digital certificates in use today for open network communication on the Internet are obtained from commercial CAs, which follow a number of standard practices and processes. However, an increasing number of organizations are beginning to deploy certificate services to implement CAs for issuing certificates on their intranets.

Windows 2000 includes Certificate Services, which you can deploy to create CAs in your enterprise. Various third-party vendors also provide certificate service products you can use to deploy CAs in your enterprise. For more information about Windows 2000 Certificate Services, see "Windows 2000 Certificate Services and Public Key Infrastructure" in this book. For more information about a third-party certificate service product, contact the specific vendor.

Services Provided by Certification Authorities

The role and function of CAs are basically the same, whether on an intranet or on the Internet. CAs perform the following basic services during the certificate lifecycle:

- Process certificate requests to verify the identity of the requester and then issue a certificate that is prepared according to the policy for that CA.
- Manage the certificate audit trail from the enrollment of the certificate to its expiration or revocation.
- Renew certificates before they expire.
- Revoke certificates as necessary.
- Maintain and publish certificate revocation lists (CRLs).

Certificate Policies and Certification Authority Practices

A *certificate policy* states your organization's requirements for certificates, such as public key lengths, certificate lifetimes, and uses for certificates. A *Certificate Practice Statement* (CPS) specifies the practices that the CA employs to issue and manage certificates to meet your certificate policies. A CPS also describes the CA's criteria and process for validating and approving certificate requests, revoking certificates, and publishing CRLs.

A commercial CA commonly publishes its CPS on its public Web site, so anyone can read the CPS to find out what practices the CA follows to issue various types of certificates. For example, a CPS might explain that the CA issues a basic type of certificate after it verifies the requester's e-mail address. For software publisher certificates, however, the CA conducts a thorough background check and requires certain collaborating evidence to verify the identity of requesters. Based on the CPS, you might choose to have low trust for basic certificates, but high trust for software publisher certificates issued by that CA.

Certificate policies can include the following types of information:

- Purposes for which the certificate can be used (user authentication, digital signing of software, encrypted e-mail, and so forth).
- Private key management requirements, such as requiring storage on smart cards or other hardware devices.
- Whether the private key can be exported.
- Requirements for users of the certificates, including what users must do in case their private key is lost or compromised.
- Requirements for certificate enrollment and renewal.
- Certificate lifetime.
- Cryptography algorithms to be used.
- Minimum length of the public key and private key set.

A CPS can include the following types of information:

- Positive identification of the CA (including CA name, server name, and DNS address).
- Certificate policies that are implemented by the CA and the certificate types that are issued.
- Policies, procedures, and processes for issuing and renewing certificates.
- Cryptography algorithms and key length used for the CA certificate.
- Lifetime of the CA certificate.
- Physical, network, and procedural security of the CA.
- The certificate lifetime of each certificate issued by the CA.
- Policies for revoking certificates, including conditions for certificate revocation such as employee termination and misuse of security privileges.
- Policies for certificate revocation lists (CRLs), including CRL distribution points and publishing intervals.
- Policies for certificate revocation lists (CRLs), including CRL distribution points and publishing intervals.
- Policy for renewing the CA's certificate before its expiration.

Security for Certificate Authorities

In general, it is important to provide high levels of security for CAs and their private keys. Each CA is certified with a CA certificate and uses its private key to sign all of the certificates and the certificate revocation lists it issues. If someone can steal or discover the CAs private key, they can impersonate the CA and issue counterfeit certificates. Likewise, someone who has the CA's private key, can publish counterfeit certificate revocation lists. Therefore, protecting the CA's private key is crucial to ensuring its integrity.

Ways to Trust Certificate Authorities

Many public key infrastructures, including the Windows 2000 public key infrastructure, support a hierarchical trust model where trust is placed in root CAs that are used to certify child CAs also called subordinate CAs. The root CA has a self-signed certificate and is the most trusted CA in an enterprise. Root CAs can issue subordinate CA certificates and these subordinate CAs can, in turn, issue subordinate CA certificates. The resulting CA trust chain or certification hierarchy can be many levels deep. You can choose to trust certificates for security functions based on trust for the root CA of the certification trust hierarchy for the issuing CA. The chain of trust for certificates is called the *certification path*. For more information about CA hierarchies and trust, see "Windows 2000 Certificate Services and Public Key Infrastructure" in this book.

Many public key infrastructures, including the Windows 2000 public key infrastructure, also include mechanisms for cross-certification trust so you can choose to trust certificates for CAs that are not in your organization's certification trust hierarchies. Windows 2000 certification provides a unique method for trusting third-party certificates and CAs that are called Certification Trust Lists (CTLs). Because certification hierarchies provides a very broad trust for all certificates issued by CAs in the chain, you can often use cross-certification trust to narrow the scope of your trust for certain certificates. For example, in Windows 2000 you can use CTLs to trust specific certificates issued by a business partner's CA to grant access to Web resources on an extranet. Even though the certificate might be valid for many purposes, you can use a CTL to restrict the authorized purposes of the certificates to Web authentication only.

Risk Factors for Cryptography Systems

There is no simple formula for determining how safe a specific cryptosystem is from attacks and potential security compromises. However, the following factors affect the risk of successful attacks on cryptosystems:

- Symmetric key length
- Public key length
- Key lifetimes
- Amount of plaintext known to attackers
- Strength of the security technology implementation
- Randomness of generated keys
- Strength of the security protocols
- Secure storage of private keys

The following sections are a summary of the complex topics you need to consider when you evaluate risks for cryptography-based security. For more detailed information about the following topics, see the cryptography literature referenced under "Additional Resources" at the end of this chapter.

For information about evaluating the risk of attacks on your security systems, see "Choosing Security Solutions That Use Public Key Technology" in this book.

Symmetric Key Length

Symmetric key encryption is subject to *key search attacks* (also called *brute force attacks*). In these attacks, the attacker tries each possible key until the right key is found to decrypt the message. Most attacks are successful before all possible keys are tried.

In general, you can minimize the risk of key search attacks by choosing shorter key lifetimes and longer key lengths. A shorter key lifetime means that each key encrypts less information, which reduces the potential damage if one of the keys is compromised.

Longer key lengths decrease the possibility of successful attacks by increasing the number of combinations that are possible. For example, for a 40-bit key, there are 2^{40} possible values. By using a personal computer that can try 1 million keys per second, an attacker can try all possible keys in about 13 days. However, an 128-bit key has 2^{128} possible values. If you could use a computer that would allow you to try 100 billion keys a second and you used 10 million of these computers, it would take about 10^{13} years to try every possible 128-bit key value. This is nearly 1000 times longer than the estimated age of the universe (15 billion to 20 billion years). Therefore, the longer the key, the more protection you have from attacks. For symmetric key encryption, increasing the key size by one bit doubles the effort that is required for an exhaustive key search attack. Doubling the key size squares the amount of effort required.

However, the time that is required to launch a successful key search attack (that is, the time that is required to discover a symmetric key) has diminished as the computing power that is available to attackers has increased. In addition, as the price of powerful computers has decreased, successful key search attacks have become more common. With the fastest off-the-shelf personal computers that are available today, an attacker now can break 40-bit encryption in hours rather than days. Some leading cryptography experts had previously estimated the cost of building a computer that could crack the widely implemented 56-bit Data Encryption Standard (DES) algorithm to be several million dollars or more. But in July 1998, Electronic Frontier Foundation in San Francisco, California, used a supercomputer that they built from off-the-shelf components at a cost of less than $250,000 to launch a key search attack on a message that had been encrypted with this algorithm. The computer searched 92 billion keys per second and decrypted the message in 56 hours after trying about 25 percent of the possible keys. In January 1999, the same organization used an improved version of its off-the-shelf supercomputer to break a 56-bit DES encrypted message in about 22 hours—less than half the time it took to break a 56-bit DES encrypted message only 6 months earlier.

Symmetric keys that are at least 64 bits long generally provide strong protection against brute force attacks. Today, symmetric keys that are 128 bits or longer are considered unbreakable by brute force attacks. However, the power of personal computers has historically doubled approximately every 18 months. In addition, attackers often develop new techniques and algorithms to improve the efficacy of key search attacks. Therefore, estimates of the time required for successful key search attacks must be revised downward as the computing power and resources available to attackers increases.

Public Key Length

Given a key of the same length, public key cryptography generally is more susceptible to attack than symmetric key cryptography, particularly to *factoring attacks*. In a factoring attack, the attacker tries all of the combinations of numbers that can be used with the algorithm to decrypt ciphertext. Factoring attacks are similar to key search attacks, but the number of possible factors varies with each algorithm and with the length of the public key and private key that are used. In general, for a given key length, a factoring attack on a public key requires fewer attempts to be successful than a key search attack on a symmetric key.

Although a 128-bit, symmetric key is generally considered unbreakable today, a 256-bit public key offers no protection from a knowledgeable attacker. As the size of public keys and private keys are increased, the effort required to compromise the keys by factoring attacks increases greatly—but at less than the exponential rate for symmetric keys. Therefore, the minimum length of public keys recommended for use today is 512 bits. However, to protect valuable information and highly confidential communications, it is recommended that you use public keys longer than 512 bits when feasible.

Note Using large public and private keys can have the adverse effect of placing a significantly higher performance load on computer processors. To ensure adequate performance of large keys, be sure to test proposed public keys in a controlled environment prior to deployment.

Public key cryptography is also more susceptible to attack because the public key encryption algorithms are more likely to produce patterns and clues in the ciphertext that help attackers decipher the plaintext with cryptanalysis. Furthermore, the public key is available for attackers to exploit in their attempts to crack the private key.

In addition, because public key algorithms are based on difficult-to-prove mathematical anomalies and number theory, an attacker might be able to discover an unforeseen shortcut for resolving the mathematical problem that provides the basis for a public key algorithm. If so, an attacker might break a public key cryptography scheme in less time than cryptography experts had predicted it would take and could keep the shortcut a secret.

Key Lifetimes

Key length is only one factor in the strength of both symmetric key and public key cryptography algorithms. The longer that a secret key or private key is used, the more susceptible it is to attack. The longer a key is used, the greater the amount of information that is encrypted with the key. In addition, a longer key lifetime also gives attackers more time to exploit weaknesses in the cryptography algorithm or its implementation.

In general, the more valuable the information that is to be protected by a key, the shorter the key lifetimes must be. The shorter lifetime not only minimizes the amount of ciphertext available for cryptanalysis, it also limits the damage that is caused if a key is compromised after a successful key attack.

Amount of Plaintext Known to Attackers

Key search or factoring attacks are seldom required to reveal the contents of encrypted information. Other types of cryptanalysis methods can be used to break encryption schemes, including known plaintext attacks and chosen plaintext attacks. Attackers can collect ciphertext to help them determine the encryption key. The more plaintext that is known to attackers, the greater the potential that an attacker can discover the encryption key used to produce ciphertext.

For a *known plaintext attack*, an attacker uses known information in encrypted files (such as standard e-mail headers) to break the encryption scheme for the rest of the ciphertext. For example, an early version of the Microsoft® Windows® 95 password file contained known encrypted plaintext, which enabled intruders to easily decipher user passwords that were stored in the file. When this security hole was discovered, a software patch was provided that removed the known plaintext from the encrypted portion of the Windows 95 password file.

For a *chosen plaintext attack*, an attacker chooses plaintext and submits it to be encrypted. Attackers can then analyze the ciphertext that corresponds to the chosen plaintext, identify subtle differences and patterns, and quickly break the encryption. Chosen plaintext attacks are relatively easy to launch. Attackers often can easily send chosen information through encrypted channels and then monitor the encrypted traffic with a network sniffer program to collect the chosen plaintext that has been encrypted.

Public key algorithms, in particular, are very susceptible to chosen plaintext attacks because attackers have the public key. Therefore, attackers can freely choose what plaintext to encrypt. All public key cryptosystems are vulnerable to chosen plaintext attacks and, thus, use a variety of techniques to protect against these attacks.

Many bulk encryption technologies, such as S/MIME secure mail and EFS, generate a new secret encryption key for each message or file that is encrypted. By using a new encryption key for each message or file, these technologies limit the amount of ciphertext available for cryptanalysis of that symmetric key. Likewise, secure online communications technologies, such as TLS and IPSec, normally use short lived session keys to limit the amount of ciphertext available for cryptanalysis. Some technologies, such as IPSec and the Kerberos authentication protocol, enable you to configure the lifetime of session keys to limit available ciphertext for each session key.

In general, you can reduce the risk of plaintext attacks by doing the following:

- Limit key lifetimes. This reduces the amount of ciphertext available for cryptanalysis for a particular key. The smaller the amount of ciphertext, the smaller the amount of material that is available for cryptanalysis, which reduces the risk of cryptanalysis attacks.
- Minimize the encryption of known plaintext. For example, if you encrypt known information such as system files on a hard disk, the known plaintext is available for cryptanalysis. You can reduce the risk of attack by not encrypting known files and sections of the hard disk.
- Minimize the amount of plaintext that is encrypted with the same session key. For example, during confidential IPSec communication, an attacker might be able to submit chosen plaintext for cryptanalysis. If the session key that is used to encrypt information is changed frequently, the amount of ciphertext produced by a single session key is limited, and thus reduces the risk of plaintext attacks.

Strength of the Security Technology Implementation

The strength of cryptography-based security depends on the strength of the encryption algorithm and the technology that implements the security. A weak algorithm or a poorly implemented security technology can be exploited to decrypt any ciphertext that it produces. For example, a weak algorithm can produce ciphertext that contains hints or patterns that greatly aid cryptanalysis. A poorly implemented security technology might also provide unintentional backdoors that attackers can discover and exploit. For example, a poorly implemented security technology might provide a way for attackers to obtain secret keys from memory caches.

For maximum protection, encryption keys need to be generated randomly; however, faulty implementations of random generators also can produce keys that are predictable. Furthermore, the implementations of security technologies can be attacked either at the system level or at the application level. An intruder might be able to start a computer from a floppy disk and use low-level disk tools to locate the encryption keys that are stored on the hard disk. If an operating system caches secret keys in memory, attackers might be able to force buffer overflows or core dumps to reveal keys. Attackers also might be able to plant an unauthorized application or virus to modify how an encryption implementation works (for example, to generate random-looking keys that are known to the attacker).

The best implementations of cryptography-based security are generally provided by security products that have been analyzed and tested over time and that have no known significant security flaws or weaknesses. However, no security software is perfect, so it is important to promptly fix significant security holes in products as they are discovered. Many vendors, including Microsoft Corporation, make timely security fixes available for their products when they are needed.

In general, you can reduce the risk from weaknesses in cryptography-based security products by doing the following:

- Use cryptography-based products that have been thoroughly analyzed and tested over time.
- Provide adequate system and network security measures, to reduce the potential for exploitation of weaknesses in your cryptography-based security systems. For example, you might protect servers that provide security by configuring the servers for high security and placing them behind firewalls.
- Update security applications and systems when security patches and fixes become available to correct problems as they are discovered.

Randomness of Generated Keys

To prevent key generation from being predictable, keys must be generated randomly. However, keys that are generated by computer software are never generated in a truly random manner. At best, software-key generators use pseudo-random processes to ensure that virtually no one can predict what keys are going to be generated. However, if an attacker can predict the major variables that are used in key generation, he or she also can predict what keys will be generated.

When properly implemented, software-based key generation provides ample security for a wide range of network and information security needs. But there is always a slight risk associated with software-generated keys, no matter how well the random-key generator is implemented. Therefore, to provide maximum protection of highly valuable information, consider deploying security solutions that provide truly random, hardware-generated keys.

Strength of the Security Protocols

Cryptography-based security technologies are implemented by using security protocols. For example, secure mail systems can be implemented by using the S/MIME protocol, and secure network communications can be implemented by using the IPSec suite of protocols. Likewise, secure Web communications can be implemented by using the TLS protocol.

Standards for security protocols, however, whether proprietary or open standards, often contain weaknesses or limitations that attackers can exploit (for example, to launch denial of service attacks). Even the best implementations of protocol standards contain the weaknesses and limitations that are inherent in the standards. Furthermore, protocol standards usually enable support for weaker cryptography by design. For example, the TLS protocol enables confidential communications to default to weak encryption to support government-imposed export restrictions that have been placed on cryptography.

In general, you can reduce the risk of weaknesses or limitations in security protocols by doing the following:

- Use protocols that have been thoroughly analyzed and tested over time and that have well understood limitations with acceptable security risks.
- Use the most recent versions of protocols, which offer stronger security or fix identified weaknesses in previous versions of the protocol. Protocols are revised periodically to improve the protocol and add new benefits and features.
- Use the strongest security options that are available with the protocol to protect valuable information. When it is feasible, require strong cryptography and do not allow systems to default to lower strength cryptography settings unless the value of the information to be protected is low.
- Prohibit the use of older and weaker versions of protocols when you want to protect valuable information. For example, require Secure Sockets Layer (SSL) version 3 or TLS for secure Web communications, and prohibit less secure SSL version 2 communications.

Secure Storage of Private Keys

The security of private keys is crucial for public key cryptosystems. Anyone who can obtain a private key can use it to impersonate the rightful owner during all communications and transactions on intranets or on the Internet. Therefore, private keys must be in the possession only of authorized users, and they must be protected from unauthorized use.

For software-based public key cryptography, cryptography operations occur in the computers operating system memory. Attackers might be able to force buffer overflows or memory dumps to obtain private keys. Even if a private key is protected by encryption while it is in memory, obtaining the protected key is the first step in a potential attack to discover what the key is. Hardware-based cryptography is inherently more secure than software-based cryptography.

In addition, many cryptosystems also store private keys on local hard disks. An attacker with access to a computer might use low-level disk utilities to locate encrypted private keys on the hard disk and perform cryptanalysis to decipher the key. In general, the risk of attacks on private keys is much lower when keys are stored on tamper resistant hardware devices such as smart cards.

In general, you can provide more security for private keys by doing the following:

- Provide physical and network security for computers and devices where private keys are generated and stored. For example, you can store servers used for CAs or secure Web communications in locked data centers and configure network and computer security features to minimize the risks of attacks.
- Use hardware-based cryptography devices to store private keys. Private keys are stored on tamper-resistant hardware rather than on the computer's hard disk drive. All cryptography takes place in the crypto-hardware, so private keys are never revealed to the operating system or cached in memory.

You generally provide the highest security for private keys where the compromise of the key would cause the most potential damage. For example, you might provide the highest security for your organization's CA keys and Internet software publishing (code signing) keys. You might also require smart cards for private keys that control access to valuable Web resources or that secure valuable e-mail communications.

Cryptography Export Restrictions

Cryptography is subject to export restrictions. Some governments, including the United States government, currently place export restrictions on encryption technology. Other governments also place import restrictions on encryption technology. The availability of the encryption technologies that are described in this guide and the actual strength of the encryption that you are allowed to use for security varies according to the export or import restrictions for a specific geographical area.

Windows 2000 and other security products you use might be export-controlled in geographic areas where your organization has offices. If so, security systems are going to be limited in cryptographic strength for those areas. Give cryptography export restrictions careful consideration when you are planning security systems.

In general, products and technology with exportable cryptography provide much less security than the nonexportable versions of the same products and technologies. Exportable security technology usually limits cryptography to much shorter symmetric encryption keys than the key lengths that are allowed for the nonexportable version of the same technology. For example, the nonexportable version of a secure mail product might use bulk encryption keys with a length of 128 bits. However, the bulk encryption keys for the exportable version of the secure mail product might be limited to a key length of 40 bits or 56 bits. Likewise, the nonexportable version of a secure Web browser might support 128-bit encryption for secure Web communications, whereas the exportable version might support only 40-bit or 56-bit secure communications. Shorter encryption keys are much more vulnerable to brute force attacks than are longer keys (keys that are at least 64 bits in length generally provide strong protection against brute force attacks). Therefore, consider using exportable technology only to protect information of relatively low value.

If you provide cryptography-based security between groups that use exportable technology and groups that use nonexportable technology, the cryptographic strength is limited to the lowest common denominator that is supported by both the exportable technology and the nonexportable technology. Security is generally limited by the exportable technology. For example, encrypted secure mail between groups that use exportable cryptography and groups that use nonexportable cryptography is limited to the longest key length supported by the exportable cryptography (for example, 40 bits or 56 bits). You cannot use the exportable technology to read secure mail that was encrypted with an 128-bit key.

Take the current limitations of exportable cryptography into account when developing your security plans. However, note that the cryptographic strength of the available technologies for exportable security products is subject to change when government policies on cryptography export change. The actual strength of the available cryptography technology might change before your security plans are implemented. For the latest information about the cryptography technology available for the products you intend to deploy, contact each applicable vendor.

For more information about the current cryptography restrictions and policies for Microsoft security products, see the Microsoft Security Advisor link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Additional Resources

For more information about cryptography, see the following publications:

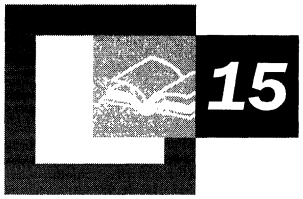
- *Applied Cryptography: Protocols, Algorithms, and Source Code in C* by B. Schneier, 1995, New York: John Wiley & Sons.
- *Internet Cryptography* by R. E. Smith, 1997, Reading: Addison-Wesley.

For more information about public key technology and security on intranets and the Internet, refer to the following publications:

- *Web Security & Commerce* by S. Garfinkel and G. Spafford, 1997, Sebastopol: O'Reilly & Associates.
- *Secure Electronic Commerce: Building the Infrastructure for Digital Signatures and Encryption* by W. Ford and M. S. Baum, 1997, Upper Saddle River: Prentice Hall PTR.
- *Understanding Digital Signatures: Establishing Trust over the Internet and Other Networks* by G. L. Grant, 1997, New York: McGraw-Hill.
- *Digital Certificates: Applied Internet Security* by Jalal Feghhi, Jalil Feghhi, and P. Williams, 1998, Reading: Addison-Wesley.
- See the Internet Engineering Task Force (IETF) link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

CHAPTER 15

Encrypting File System



Encrypting File System (EFS) is a new feature in Microsoft® Windows® 2000. EFS protects sensitive data in files that are stored on disk using the NTFS file system. It uses symmetric key encryption in conjunction with public key technology to provide confidentiality for files. It runs as an integrated system service, which makes EFS easy to manage, difficult to attack, and transparent to the file owner and to applications. Only the owner of a protected file can open the file and work with it, just as with a normal document. Others are denied access to the protected file. However, recovery administrators (whom you can designate) have the ability to recover protected files if that becomes necessary.

In This Chapter

Related Information in the Resource Kit

- For more information about the NTFS file system, see "File Systems" in the *Microsoft® Windows® 2000 Server Operations Guide*.
- For more information about Windows 2000 Certificate Services, see "Windows 2000 Certificate Services and Public Key Infrastructure" in this book.
- For more information about the basic concepts of cryptography, including symmetric key encryption and public key technology, see "Cryptography for Network and Information Security" in this book.

Introduction to EFS

You can use EFS to encrypt NTFS files to provide confidentiality for the file contents. EFS uses symmetric key encryption in conjunction with public key technology to protect the file and ensure that only the owner of the file can access the file. Users of EFS are issued a digital certificate with a public and private key pair that are used for EFS operations. To conduct EFS operations, EFS uses the key set for the user who is logged on to the local computer where the private key is stored. You can use EFS for file systems on remote computers only if the remote computers are trusted for delegation.

No administrative effort is needed to begin using EFS, and most operations are transparent. You also have the option to disable EFS by configuring EFS recovery policy.

Features of EFS

EFS provides its users with privacy, transparent operation, and a means of data recovery. In addition, it ensures that encryption is not inadvertently defeated by copying or moving files.

Privacy

EFS is designed to protect the privacy of sensitive data. Besides the user who encrypts a file, only designated recovery agent personnel can decrypt it. Other system accounts that have permissions for that file—even the Take Ownership permission—cannot open the file without the encryptor's private key.

EFS is especially useful for securing sensitive data on computers shared by several users and on portable computers. Both kinds of systems are susceptible to attack by techniques that circumvent the restrictions of access control lists (ACLs). In a shared system, access can be gained by starting up a different operating system. With a portable computer, a thief might take only a moment to steal it. The thief can then remove the hard disk drive, plug the hard disk drive into another computer, and read the files. EFS files, however, appear as unintelligible characters when the thief does not have the decryption key.

Transparent Operation

In EFS, file encryption does not require the file owner to decrypt and re-encrypt the file on each use. Decryption and encryption of the file take place transparently as it is read from and written to the disk.

In contrast, encryption services in most products are not transparent to the user. The user has to decrypt the file before every use and re-encrypt it when finished. If the user forgets to encrypt a file, the file is unprotected. And, because the user must go to the trouble of specifying that a file be encrypted and decrypted on each use, it discourages the use of encryption.

Integration with the File System

EFS is tightly integrated with NTFS. You set the encryption attribute for folders or files as you set other attributes, such as read-only, compressed, or hidden. When encryption is set for a folder, EFS automatically encrypts the following:

- All new files created in the folder
- All plaintext files copied to the folder
- Optionally, all existing files and subfolders in the folder

When EFS is implemented at the folder level, temporary copies of an encrypted file in the same folder (such as those created during editing) are also encrypted, as are backups created in the same folder. The encryption survives moves and renames, provided that all files are on Windows 2000 NTFS volumes.

Note EFS is available only on Windows 2000 NTFS volumes. Copying or moving the file or folder to another file system removes the encryption and returns the file to its normal format. The exception to this is files and folders that are stored by Windows 2000 Backup. Files and folders remain encrypted on the backup media.

Data Recovery System

Encrypting a file always raises a risk that it cannot be read again. The owner of the private key might leave the enterprise. If disgruntled, the owner might maliciously encrypt all of his or her files before leaving. Worse yet, he or she might encrypt critical shared files so that no one else can use them. For this reason, EFS is designed to be used only if the system is configured with one or more recovery agent administrators.

Designated user accounts, called *recovery agent accounts*, are issued recovery agent certificates with public keys and private keys that are used for EFS data recovery operations. Recovery agent accounts are designated by EFS recovery policy. By default, the recovery agent account is the highest-level Administrator account. On a stand-alone computer, this is the local Administrator. In a domain, the domain Administrator for the first domain controller installed in the domain is the default recovery agent account for all computers in the domain. Different recovery agent accounts can be assigned by changing EFS recovery policy, and different recovery policies can be configured for different parts of an enterprise. The private key for a recovery agent account must be located on the computer where recovery operations are to be conducted.

When a recovery agent certificate is issued, the certificate and private key are installed in the user profile for the user account that requested the certificate. You also have the option to export the recovery agent certificate and private keys to store them in archives or to transfer the certificate and private key to other user accounts and computers.

There can be more than one recovery agent account for an EFS file, each with a different private key. Data recovery discloses only the encrypted data, not the user's private key that was used to encrypt the bulk encryption key or any other private keys for recovery. This ensures that no other private information is revealed to the recovery agent administrator accidentally.

If you choose to configure an EFS recovery policy with no recovery agent certificates, EFS is disabled. Because of this feature, you cannot normally use EFS to encrypt data so it cannot be recovered—whether the encryption is done through accident or through malice. However, you could later destroy the private key for recovery to prevent data recovery.

Information to Remember About EFS

EFS only works on the Windows 2000 NTFS file system.

EFS does not run if there is no recovery agent certificate, but it does designate a recovery agent account by default and generates the necessary certificate if you do not.

You can use EFS to encrypt or decrypt data on a remote computer, but you cannot use it to encrypt data sent over the network.

You cannot encrypt system files or folders.

You cannot encrypt compressed files and folders until you decompress them.

Encrypting an entire folder ensures that the temporary copies of encrypted files that it contains are also encrypted.

Copying a file into an encrypted folder encrypts the file, but *moving* it into the folder leaves the file encrypted or unencrypted, just as it was before you copied the file.

Moving or copying EFS files to another file system removes the encryption, but backing them up preserves the encryption.

Other file permissions are unaffected. An administrator, for instance, can still delete a user's EFS file even though the user cannot open it.

What the User Sees

EFS might take a long time to run the first time you use it. In the first encryption operation, EFS must also check whether the user has a valid EFS user certificate, containing the user's public key. If the user does not have a valid EFS user certificate, EFS requests the certificate from an enterprise certification authority (CA) if one is available online or, if a CA is not available, it creates one for the user. After this is done, as long as the certificate is valid, the process does not need to be repeated for the same user and EFS then works much faster.

The following sections provide user scenarios that demonstrate how EFS works.

Finding and Accessing an Encrypted File

Encryption is an attribute of a file or folder, so to see whether a file or folder is already encrypted, examine its attributes. The Large Icons and Small Icons view in Windows Explorer display the attributes of a selected folder to the left of the file or folder icon. In Figure 15.1, the screen shows you that the file Floppyres.txt is encrypted.

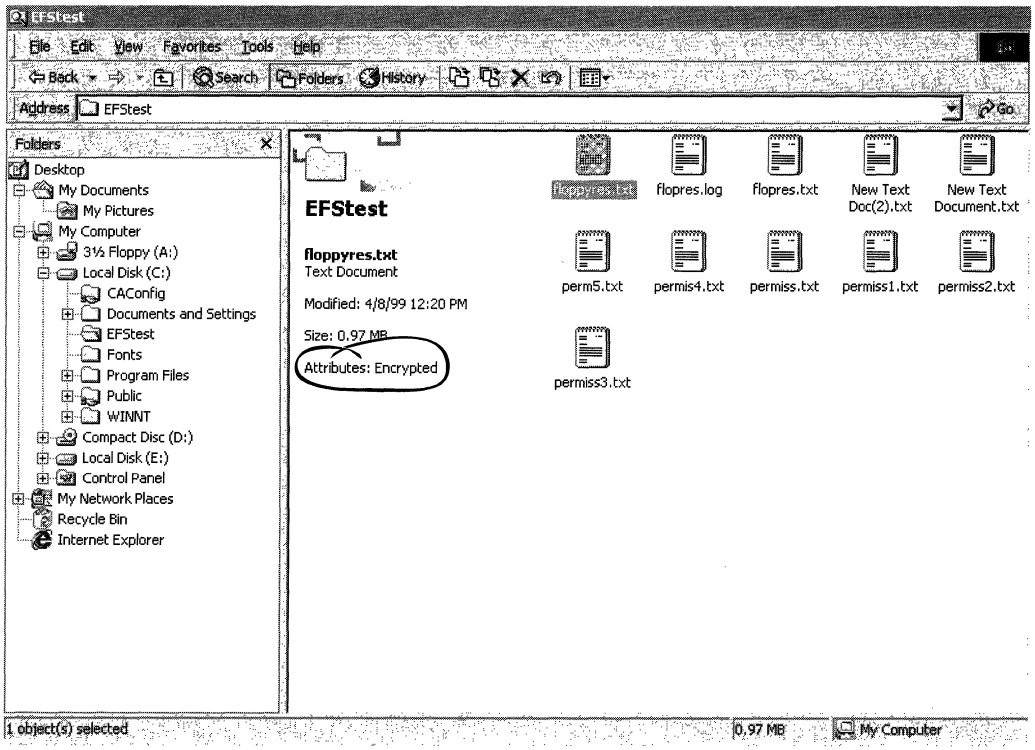


Figure 15.1 Large Icons View in Windows Explorer

The Details view in Windows Explorer can also show attributes; *E* in the attributes column means that the file is encrypted. You can configure Windows Explorer to display not only the *E* attribute, but also file ownership—so you know whether it is one of yours and whether you can open it. Figure 15.2 shows an example of this.

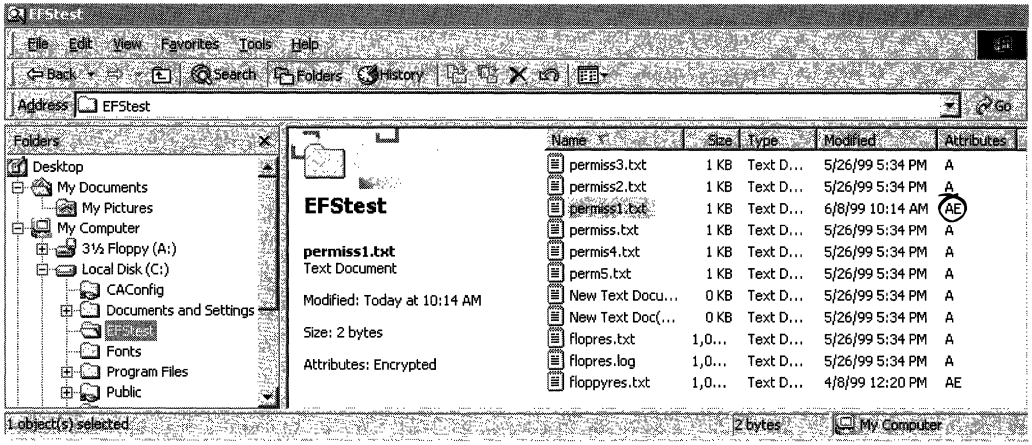


Figure 15.2 Details View in Windows Explorer

You can also change which fields are displayed in the Contents pane. To do this, right-click any of the column headings in the Contents pane. This displays a menu with the names of available fields. Fields that are currently displayed have a check mark by them. Click a field to display it or to remove it from the display.

In Figure 15.2, the **Attributes** column for the file Permiss1.txt displays *AE* to indicate that the file has the archive attribute on (*A*) and the encryption attribute on. Only encrypted files have *E* in the **Attributes** column.

Because EFS encryption is transparent, you can use an encrypted file in the same way that you did before it was encrypted. For example, you can still open a Microsoft® Word or Notepad document and modify it. Any other user who is trying to open this file gets an "Access denied" error message.

Note Windows 2000 identifies users by their account names, including the server name. For example, if Alice encrypts a file on her domain account, Wonderland\Alice, she can use the file only when she is logged on to the domain. If she tries to open the file while she is logged on to her local account, MyLaptop\Alice, access is denied.

Encrypting a File or Folder

Because encryption is an attribute, you must have write permission to encrypt a file or folder. But even if you have write permission, you cannot encrypt files or folders in the *systemroot* folder (for example, C:\WinNT\Notepad.exe or C:\WinNT\System32). You also cannot encrypt files or folders that have their system attribute set. If these types of files and folders could be encrypted, it might render the system useless. This is because many of these files are needed for the system to start up, and decryption keys are not available during the startup process to decrypt them. If you attempt to encrypt a file or folder in the *systemroot* folder or that has its system attributes set, the encryption attempt fails and an error message appears.

Encrypting at the folder level is recommended. There are two ways to accomplish this: from Windows Explorer, or from a command prompt, using the **cipher** command. You can also encrypt individual files by using **cipher** or Windows Explorer.

Encrypting a Folder by Using Windows Explorer

To encrypt a folder, select it in Windows Explorer, and then click **File**, click **Properties**, click the **General** tab, and then click **Advanced.** (For more information about encrypting a folder, see Microsoft® Windows® 2000 Server Help or Microsoft® Windows® 2000 Professional Help.) Figure 15.3 shows the dialog box that appears.

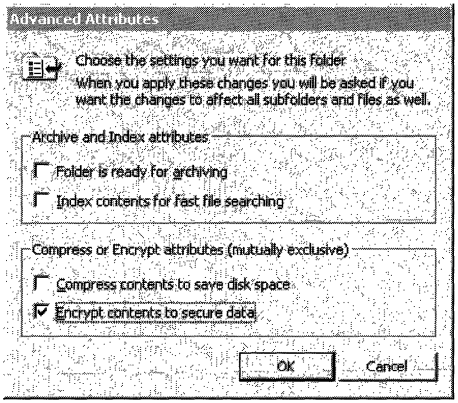


Figure 15.3 Advanced Attributes Dialog Box

Select the **Encrypt contents to secure data** check box, and then click **OK**. Figure 15.4 shows the dialog box that you use to choose which files and folders to encrypt.

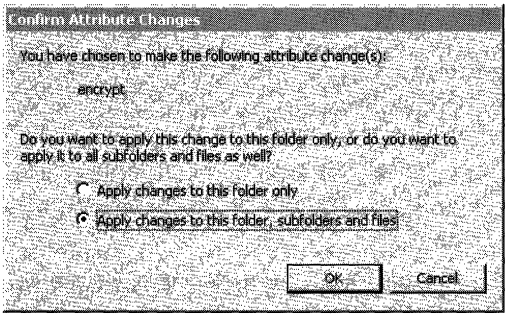


Figure 15.4 Confirm Attribute Changes Dialog Box

Table 15.1 shows the results of selecting the **Apply changes to this folder only** option.

Table 15.1 Results of Selecting the Apply Changes to This Folder Only Option

File description	Encryption status
Already stored in the folder and its subfolders.	Unchanged. They remain encrypted or unencrypted, whichever they were.
Created in or copied to the folder by you later.	Protected with your private key.
Created in or copied to the folder by another user (if the folder is shared) later.	Protected with the other user's private key.
Created in or copied to subfolders later.	Unchanged.
Moved to the folder or subfolders later.	Unchanged.

Table 15.2 shows the results of choosing the **Apply changes to this folder, subfolders, and files** option.

Table 15.2 Results of Selecting the Apply Changes to This Folder, Subfolders, and Files Option

File description	Encryption status
Already in the folder and its subfolders.	Protected with your private key if you have Write Attributes permission; otherwise, files are unchanged.
Later created in or copied to the folder or subfolders by you.	Protected with your private key.
Later created in or copied to the folder or subfolders by another user.	Protected with the other user's private key.
Later moved to the folder or subfolders.	Unchanged.

With either choice, the folder's list of files remains in plaintext and you can enumerate files as usual, provided you have access to the folder.

EFS lets you encrypt a file you do not own, provided that you have Write Attributes, Create Files/Write Data, and List Folder/Read Data permissions for the file. However, any file that you encrypt is protected with your private key, not the owner's private key. As a result, encrypting a file that you do not own locks out the owner unless you are the owner. For this reason, it is not a good idea to select **Apply changes to this folder, subfolders, and files** when the folder is shared by other users who own files or subfolders in the folder.

Encrypting a Single File from Windows Explorer

You are encouraged to turn on encryption at the folder level rather than the individual file level because it is easier; you do not have to encrypt each file individually when you save it. It also ensures that the data remains encrypted during and after editing, provided the application does its editing in the same folder. Editing creates temporary or backup files that some applications leave in plaintext. (Applications do not always use the same folder for temporary files or backup files while editing. Microsoft Word, for example, uses the folder where the encrypted file is located for temporary and backup files. Other applications often store temporary files to the Temp folder. Thus, it is recommended that you encrypt these types of folders to ensure that temporary files and backup files are encrypted.)

By using Windows Explorer, you can turn on EFS for an individual file in exactly the same way as you do for a folder. However, when you encrypt a single file, the user interface displays the warning shown in Figure 15.5.

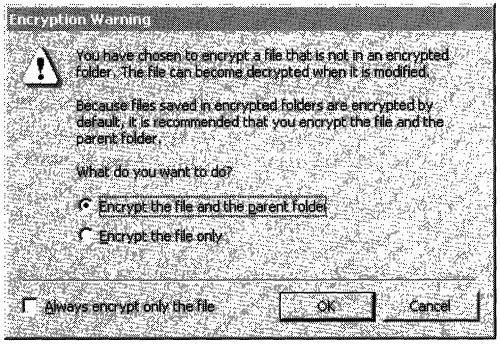


Figure 15.5 Warning Message

If you select the **Always encrypt only the file** check box, the warning no longer appears, and EFS encrypts only the file that you select.

Encrypting from the Command Prompt

You can also encrypt and decrypt folders or files by using the **cipher** command at the command prompt. In the example in Figure 15.6, the file called Flopres.txt is encrypted and then decrypted.

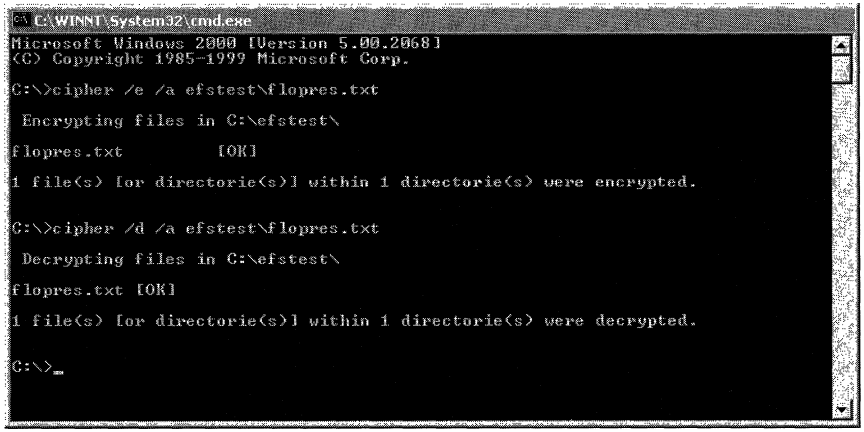


Figure 15.6 Use of Cipher Command

You can use **cipher** with a wildcard character to encrypt or decrypt files. For example, to encrypt all files with "cnfdl" in the name, you can type:

```
C:\>cipher /e /a *cnfdl*
```

You can use multiple directory names and wildcard characters. You must put spaces between multiple parameters.

Used without parameters, **cipher** displays the encryption state of the current folder and any files it contains. To see all the options available with **cipher**, type the following:

```
cipher /?
```

For more information about the cipher command, see Microsoft® Windows® 2000 Professional Help or Windows 2000 Server Help.

Decrypting a Folder or File

Decryption is the opposite of encryption—restoring encrypted data to its original plaintext form. To do this, in Windows Explorer, right-click the file or folder, click **Properties**, click **Advanced**, and then clear the **Encrypt contents to secure data** check box. This causes EFS to decrypt the selected folder and mark it as unencrypted. When you apply your choice, you have the option of decrypting all files and subfolders in the folder, in addition to resetting the folder as unencrypted.

You can also use the **cipher** command to decrypt files or folders. For example, to decrypt a file named "cnfdl.txt," you can type:

```
C:\>cipher /d /a cnfdl.txt
```

Users do not need to decrypt individual files or folders every time they access the data because EFS provides transparent encryption and decryption during data writes and reads.

Renaming, Copying, or Moving an Encrypted File or Folder

Encrypted files and folders can be renamed, copied, or moved. If this is done properly, they remain encrypted. For more information about renaming, copying, or moving encrypted files and folders, see Windows 2000 Professional Help or Windows 2000 Server Help. Table 15.3 describes the encryption status of files after managing encrypted files and folders.

Table 15.3 Encryption Status When Renaming Copying, or Moving EFS Files and Folders

Task	Encryption status
Changing the name of an encrypted file or folder	Encrypted.
Moving an encrypted file or folder	Encrypted if the target volume is Windows 2000 NTFS; otherwise, encryption is lost.
Copying an encrypted file or folder	Encrypted if the target volume is Windows 2000 NTFS; otherwise, encryption is lost.
Moving or restoring encrypted files or folders to a different computer	Encrypted if you use the Windows 2000 Backup tool, regardless of target volume. You cannot open the file on the other computer unless it has your private key.

EFS determines when an encrypted file is being moved or copied to another drive. EFS always decrypts and sends files as plaintext. The file is re-encrypted if the target drive supports EFS. If the target drive does not support EFS (for example, if it is a file allocation table (FAT) volume), the new copy of the file is stored as plaintext.

Note Most floppy disk drives are FAT volumes, and therefore, you lose your encryption unless you use the Windows 2000 Backup tool to copy to them.

If the target drive has EFS but is on a different computer, the file is re-encrypted with the sender's public key. The target computer must be trusted for delegation, or the transfer fails.

▶ **To tell whether a computer is trusted for delegation**

1. Select the computer in Active Directory.
2. See the **General** tab in the **Properties** dialog box. If the computer is trusted, the **Trusted for Delegation** check box is selected.

Domain administrators have permission to select and clear this check box. For more information about determining whether computers are trusted for delegation, see Windows 2000 Server Help.

The target computer finds the public key in the sender's profile or, failing that, creates a new certificate and key pair for the sender.

If you send an EFS file over the network (for example, by copying it to a folder on another computer), it is transmitted as plaintext. You can use a secure network protocol such as Secure Sockets Layer (SSL) or IPSec to encrypt the data while it is being transmitted.

Backing Up an Encrypted File or Folder

Backup copies of encrypted files that are made using Windows 2000 Backup or any other backup tool designed for Windows 2000 remain encrypted, regardless of destination. When you back up the encrypted file or folder to a floppy disk or other removable medium, it stays encrypted when transferred to a backup tape, a non-NTFS file system, or an e-mail attachment.

You can open the file at its new site if you have a roaming profile or if you use the Certificate Export wizard and the Certificate Import wizard to transfer your certificate and private key to your user profile on the other computer. For more information about moving or restoring encrypted files or folders to a different computer, see Windows 2000 Professional Help or Windows 2000 Server Help.

You can use Backup to send a file to a recovery agent administrator when the user's private key is lost. To do a simple move or copy, the user's key must be present to decrypt the file during the operation. Backup does not require the user's private key, however, because the file is not decrypted and re-encrypted; it keeps its original encryption. For more information about file recovery, see Windows 2000 Server Help.

Encrypting a File or Folder on Another Computer

To encrypt a file or folder on another computer, see Windows 2000 Professional Help or Windows 2000 Server Help. If the remote computer is trusted for delegation, EFS can use the public key from your profile to create a DDF and DRF for the file.

If the remote computer is not trusted for delegation, a domain administrator has to make it trusted for delegation. For more information about trusting a remote server for delegation, see Windows 2000 Server Help.

Setting Permissions for Shared Folders

It is not usually a good idea to encrypt folders that are shared by several users. Encryption is not designed to make information public; it is designed for keeping information private. However, there are situations when encrypting a shared folder might make sense. For example, on a computer that is shared by several users, it might be convenient to turn on encryption for one folder and let all users of the computer know that this folder is the place to store encrypted data. (You can do this by giving the folder a name that clearly signals its purpose—by calling it "Encrypted," for example). That way, everyone who uses the computer knows that they can encrypt a file by saving it in the encrypted folder.

If you must encrypt a shared folder, it is best to start with a new, empty folder. After you create the folder, enable encryption by following the procedure described in "Encrypting a Folder by Using Windows Explorer." Then set the permissions on the folder as shown in Table 15.4.

Table 15.4 Security Permissions for Shared Folders

Account/group	Permission	Applies to
Creator/Owner	Full Control	This folder, subfolders, and files.
Everyone	Traverse Folder/Execute File	This folder only.
Everyone	List Folder/Read Data	This folder only.
Everyone	Create Files/Write Data	This folder only.

How EFS Works

Public Key Technology

EFS uses public key encryption in conjunction with symmetric key encryption to provide confidentiality for files that resists all but the most sophisticated methods of attack. The file encryption key (FEK)—a symmetric bulk encryption key—is used to encrypt the file and is then itself encrypted by using the public key taken from the user's certificate, which is located in the user's profile. The encrypted FEK is stored with the encrypted file and is unique to it. To decrypt the FEK, EFS uses the encryptor's private key, which only the file encryptor has.

Public key encryption algorithms use asymmetric keys for encryption and decryption. "Asymmetric" means that different keys are used to encrypt and decrypt the same data. Public key encryption uses a private key (which is held only by its owner) and a public key (which is available to other entities on the network). A public key, for example, can be published in Active Directory so that it is accessible to users in the organization. The two keys are separate but complementary in function. Information that is encrypted with the public key can be decrypted only with the corresponding private key of the set. The two keys together are called a key pair or a key set.

One drawback of public key cryptography is the amount of processing time that is required for its mathematical operations. Symmetric key encryption, which uses the same key to both encrypt and decrypt, is commonly 100 to 1,000 times faster, so symmetric and asymmetric key encryption are often used together to provide a wide range of network and online information security solutions. Thus, EFS encrypts data symmetrically with the FEK and then encrypts and decrypts the FEK asymmetrically with the public key and the private key.

Structure of an Encrypted File

When you save a file to be encrypted, a random cryptographic generator supplies a unique file encryption key (FEK), which is a fast symmetric key designed for bulk encryption. The FEK encrypts the data in blocks. EFS adds a header to the file, where the FEK is stored. The header has at least two parts, as shown in Figure 15.7.

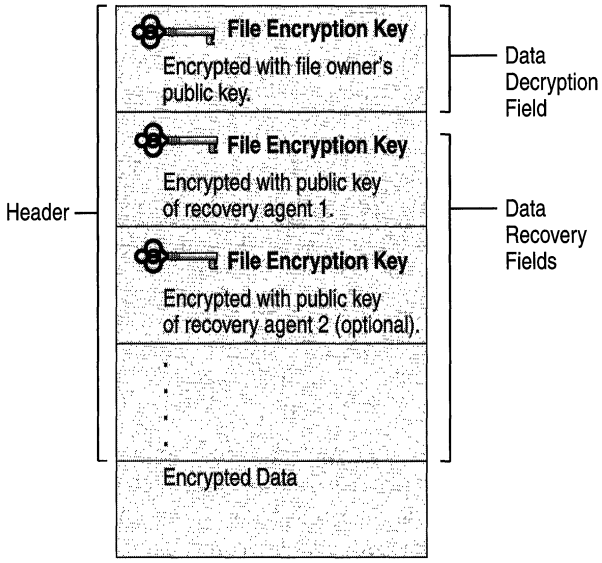


Figure 15.7 Structure of an Encrypted Data File

- The DDF contains the FEK, which is encrypted with the user's public key.
- The DRF contains the FEK, which is encrypted with each recovery agent certificates' public key. Because there can be more than one recovery agent certificate, multiple DRFs might exist.

Components of EFS

EFS consists primarily of the following components in the Windows 2000 operating system: the Microsoft Cryptographic Application Programming Interface (CryptoAPI), the EFS service, the EFS driver, and the EFS File System Run-Time Library (FSRTL). Figure 15.8 shows the EFS architecture.

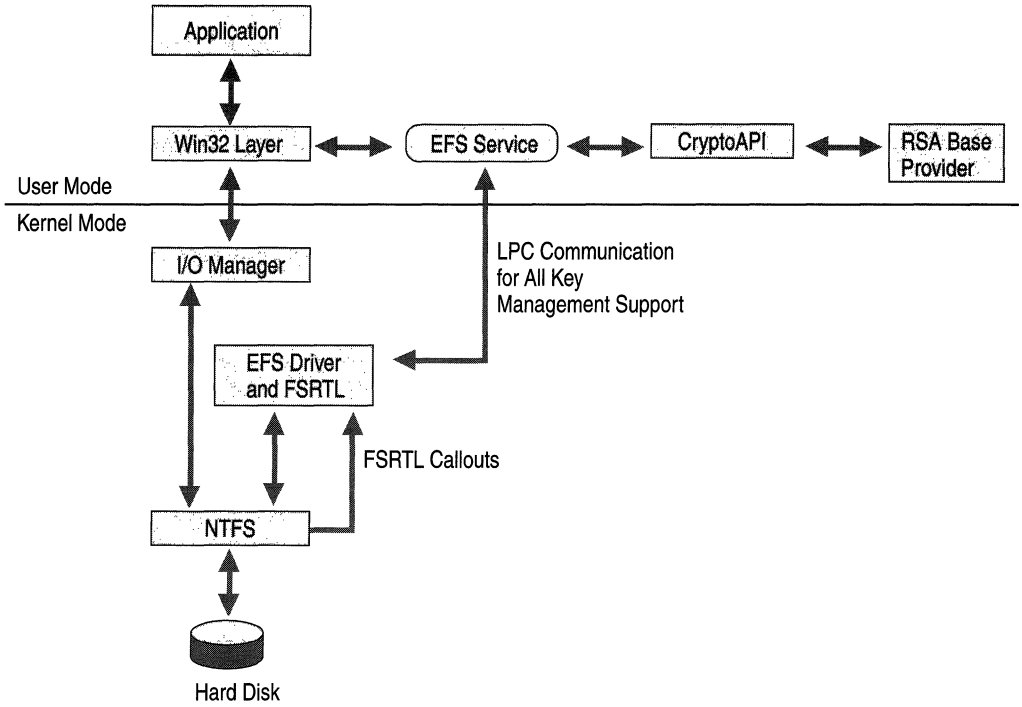


Figure 15.8 EFS Architecture

CryptoAPI

CryptoAPI is the interface for cryptographic operations in Windows 2000 and is used to generate FEKs as well as the public key pair that is used to encrypt and decrypt FEKs. EFS acquires FEKs from a statically linked library that cannot be replaced by a third-party product. EFS uses the DESX encryption algorithm for FEK generation and file encryption operations. DESX is a variation of the U.S. government's Data Encryption Standard (DES) algorithm. The public key pairs for EFS users and recovery agent accounts are obtained from the Microsoft Base cryptographic service provider (CSP). This CSP is included with Windows 2000 Professional and Windows 2000 Server and is approved for general export world wide. No other CSP can be used for EFS.

EFS Service

The EFS service calls CryptoAPI to acquire the file encryption key (FEK) for a data file and then to encode the FEK, thus producing the data decryption field (DDF) and data recovery field (DRF). The EFS service also returns the FEK, DRF, and DDF by way of the FSRTL to the EFS driver.

EFS Driver

The EFS driver communicates with the EFS service to request key management services from CryptoAPI such as the computation of FEKs, of DDFs, and of DRFs. The EFS driver passes FEKs, DDFs, and DRFs to the FSRTL to perform various file system operations (open, read, write, and append) transparently.

EFS FSRTL

When called by the EFS driver, the FSRTL reads, encrypts, and saves the file to disk. EFS avoids leakage of FEKs to paging files because the FSRTL resides in the Windows operating system kernel and uses the nonpaged pool to store the FEK. However, because the contents of paging files are not encrypted, the plaintext contents of encrypted file might be temporarily copied to paging files when encrypted files are opened and in use by applications. If the plaintext contents of encrypted files are copied to a paging file, the plaintext remains in the paging file until the contents are replaced by new data. Plaintext contents can remain in paging files for considerable amounts of time even after applications close the encrypted files.

A paging file is a system file, so it cannot be encrypted. (By default, Windows 2000 sets the name of the paging file as Pagefile.sys.) The file system security for paging files prevents any user from gaining access to and reading these files, and these security settings cannot be changed. However, someone other than the authorized user might start the computer under a different operating system to read a Windows 2000 paging file. To prevent others from reading the contents of paging files that might contain plaintext of encrypted files, you can configure security settings to clear the paging files every time the computer shuts down.

You can configure Group Policy to force all computers within the scope of the policy to clear paging files when the computers shut down. To configure Group Policy for this purpose, install the Group Policy MMC snap-in and open the Group Policy object you want to work with (domain, organizational unit, or local computer), and then follow the procedure described here.

▶ **To use Group Policy to force computers to clear their paging files when they shut down**

1. Click the Group Policy node you wish to work on, then click **Computer Configuration, Windows Settings, Security Settings, Public Key Policies, Local Policies,** and then click **Security Options**.
2. Double-click **Clear virtual memory page file when system shuts down.**

   The **Template Security Policy Setting** dialog box appears.
3. Select the **Define this policy setting in the template** check box and then click **Enabled**.

   The new settings take effect when Group Policy is refreshed at each computer within the scope of the policy.

How Files Are Encrypted

As shown in Figure 15.9, encryption with EFS requires the presence of at least two certificates in the certificate store on the local computer: one for the user (file owner) and one for a recovery agent account. If the certificates do not exist when the user logs on to the network, EFS builds them automatically.

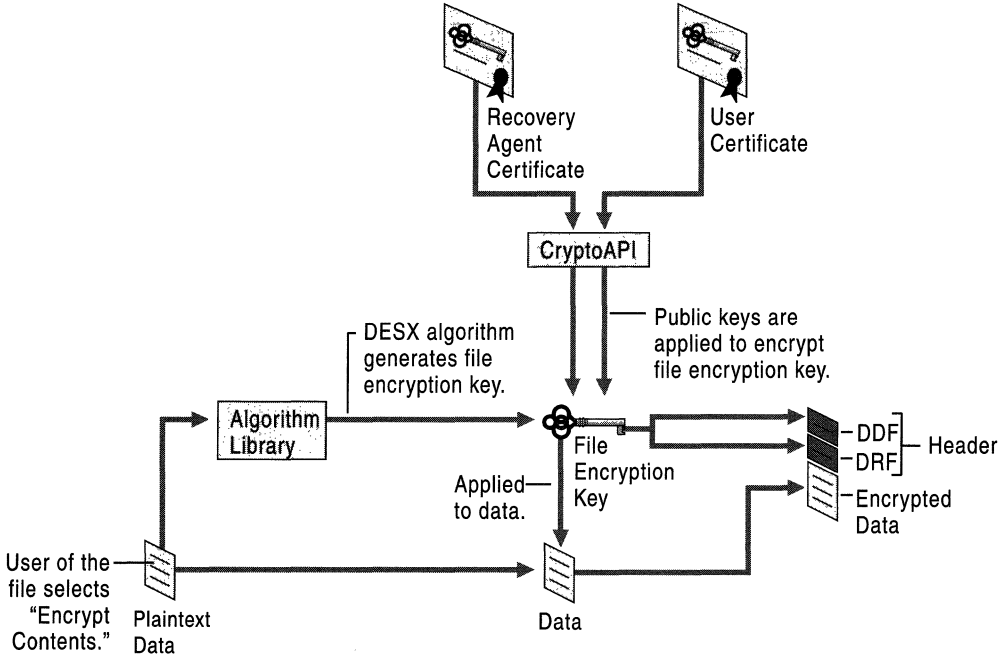


Figure 15.9 Generation of an Encrypted Data File

Creating a New File with Encryption Enabled

There are two ways to create a file that has encryption already enabled. One method is to use Windows Explorer to create a file inside an encrypted folder. For example, in Windows Explorer, you can click an encrypted folder and then, on the **File** menu, point to **New**, and then click **Rich Text Document**. A second method is available to applications when they call the CreateFile( ) API with the file encryption flag, FILE_ATTRIBUTE_ENCRYPTED. Depending on which way you do it, NTFS checks the encryption status of the parent folder or of the flag passed in. If NTFS finds that the file must be created as encrypted, it calls the EFS FSRTL, which requests the EFS service to generate the metadata (FEK, DDF, and DRF) for the file object. The EFS service acquires the metadata from CryptoAPI and returns it by means of the EFS FSRTL to the EFS driver.

When the EFS service returns the metadata, the EFS driver sets up an EFS context for the file object and creates a $EFS stream of data containing the metadata. As the user writes data in the file, NTFS calls the FSRTL to encrypt the data.

Encrypting an Existing Plaintext File

Changing a file from plaintext to encrypted (by selecting the **Encrypt contents** check box in the **Advanced Attributes** dialog box) calls EncryptFile( ). This API creates a backup copy of the file and requests the FEK, DRF, and DDF from the EFS service. The API then passes this metadata to the FSRTL by way of NTFS. The FSRTL sets up an EFS context for the file object and creates a $EFS stream with DDF, DRF, and so forth, for that object. EncryptFile( ) then reads the data from the backup and writes the data to the file. During the write, NTFS calls the FSRTL to encrypt the data before writing. When the writing is complete, the backup file is automatically deleted.

How Data Is Read and Written in Encrypted Files

Users do not need to decrypt files or folders during normal operations because EFS provides transparent encryption and decryption during data writes and reads. When you open the encrypted file, your private key (from your protected store in your user profile) is applied to the DDF to unlock the FEK, and the file contents appear on-screen in plaintext. EFS automatically detects an encrypted file and locates a user certificate and associated private key in the certificate and key stores.

Opening an encrypted file calls CreateFile( ). NTFS checks the file on the disk to determine its encryption status, then calls the EFS service to verify the user. The EFS service finds the user's certificate and private key, and requests CryptoAPI to use the private key to decrypt the FEK. The EFS service then returns the FEK by way of the FSRTL to the EFS driver. If the user password or recovery agent information has changed, the EFS service requests a new DDF or DRF from CryptoAPI, as required, and returns it with the FEK to replace the old metadata.

When the EFS service returns the metadata, the EFS driver sets up an EFS context for the file object and creates a $EFS stream containing the metadata. The FEK decrypts the file so that it appears in plaintext on the screen.

As the user writes data in the file, NTFS calls the EFS FSRTL to encrypt the data.

If another user tries to open the file, access is denied. No one else can decrypt the file without the encryptor's private key or a private key for recovery. This is true even if administrators change permissions or file attributes, or file ownership is shared.

How Files Are Decrypted

Decryption calls the DecryptFile() API. When it receives the API, the EFS driver looks up the EFS context; if the context does not exist, the file is already decrypted and nothing more need be done. Otherwise, the EFS driver opens the file and makes a backup copy in plaintext. The EFS driver then removes the encryption attribute and metadata, and writes the data back into the file.

For more information about decrypting a local folder or decrypting a folder on another computer, see Windows 2000 Professional Help or Windows 2000 Server Help.

How Files Are Recovered

If the owner's private key is unavailable (for example, because it is damaged), a recovery agent account can open the file by using the private key for recovery, which is applied to the DRF to unlock the FEK. The mechanism for file recovery works essentially the same way as decrypting a file, by using the user's private key.

A private key for recovery cannot decrypt the DDF. If there are multiple recovery agent accounts, each private key for recovery decrypts only its own DRF and no other. Thus, there is no danger that an unauthorized recovery agent account can access information from the file that enables access to other files.

As mentioned earlier, EFS assigns the local Administrator user account or the domain Administrator user account as recovery agent account by default, and you can use EFS indefinitely without ever changing this. However, it is good security practice to rotate recovery agent accounts. Because access to the file is denied if the agent designation changes, recovery certificates and private keys for recovery must be kept until all of the files that have been encrypted with them have been updated.

Note Do not encrypt files when you are logged on as the local Administrator unless you have changed the default recovery agent account. The effectiveness of EFS recovery is compromised if a file's creator is both the user and the recovery agent account.

For more information about how users can submit encrypted files for recovery or about how recovery administrators can recover files, see Windows 2000 Professional Help or Windows 2000 Server Help.

Changing EFS recovery policy must be done by a member of the Domain Admin security group. For more information about how to change EFS Recovery policy for the local computer or a domain, see Windows 2000 Server Help, and see "Configuring Recovery Agent Policy" later in this chapter.

How Certificates Are Stored

Windows 2000 stores user certificates that contain the public keys in the Personal certificate store for the certificate owner's user account. A certificate provides assurance that the public key is bound to the specific subject (an individual or other entity) that owns the private key. Certificates are stored in plaintext because they are public information and they are digitally signed by certification authorities to protect against tampering. However, the private keys must be kept confidential so only the authorized owner has access to the private key.

Certificates are issued by certification authorities (CAs), which verify the identity of entities before issuing the certificates. EFS issues its own certificates if no CA is available. However, you can deploy Certificate Services to issue EFS certificates and provide the following benefits:

- Central certificate management and the publication of certificate revocation lists
- The ability to issue alternate recovery agent certificates to designated user accounts

Each user has a personal certificate store that contains certificates that are issued to that user. User certificates reside in Documents and Settings\<*username*>\ApplicationData\Microsoft\SystemCertificates\My\Certificates for each user profile. These certificates in the user profile are written to the user's personal store in the system registry each time the user logs on to the computer. For roaming profiles, the user's certificates are located on the domain controller so the certificates follow users when they log on to different computers in the domain.

You can use the Certificates console, a snap-in to Microsoft Management Console (MMC), to view a user's personal certificate stores. Figure 15.10. shows an example of a user's personal store. The certificate for EFS displays Encrypting File System in the **Intended Purposes** column. Because users can have more than one certificate that supports EFS user operations, multiple certificates can appear with "Encrypting File System" in the **Intended Purposes** column.

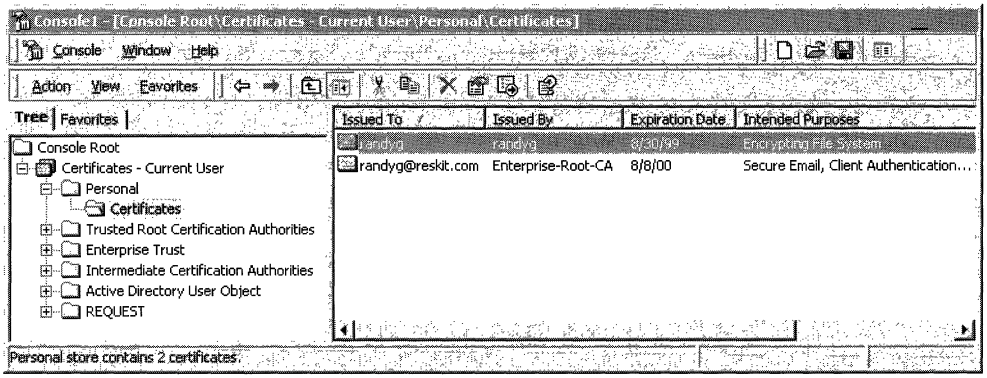


Figure 15.10 User Certificates in the Personal Certificate Store

Recovery agent certificates appear in the personal certificate store for the recovery agent account. Figure 15.11 shows an example of the personal certificate store for a recovery agent account.

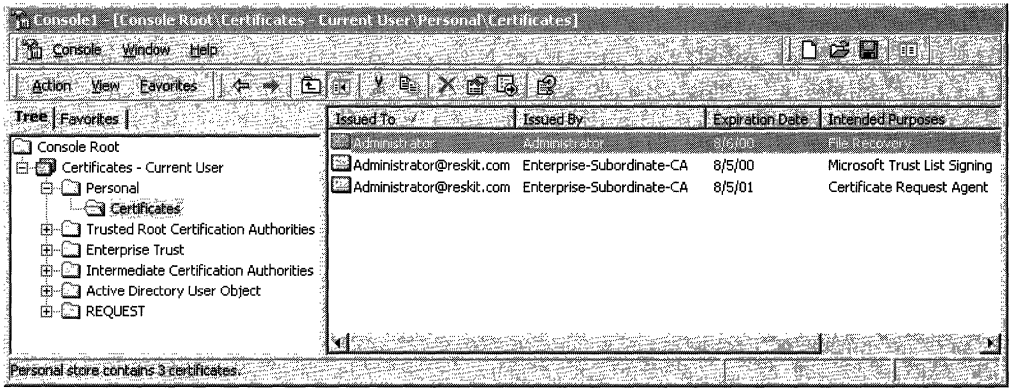


Figure 15.11 Recovery Agent Certificates in the Personal Certificate Store

For more information about certificate stores and the Certificates console, see "Windows 2000 Certificate Services and Public Key Infrastructure" in this book. The recovery certificate displays "File Recovery" in the **Intended Purposes** column.

Recovery certificates also appear in the details pane of the Group Policy console (a snap-in to MMC) for the **Encrypted Data Recovery Agents** container, as shown in Figure 15.12. Multiple certificates can appear.

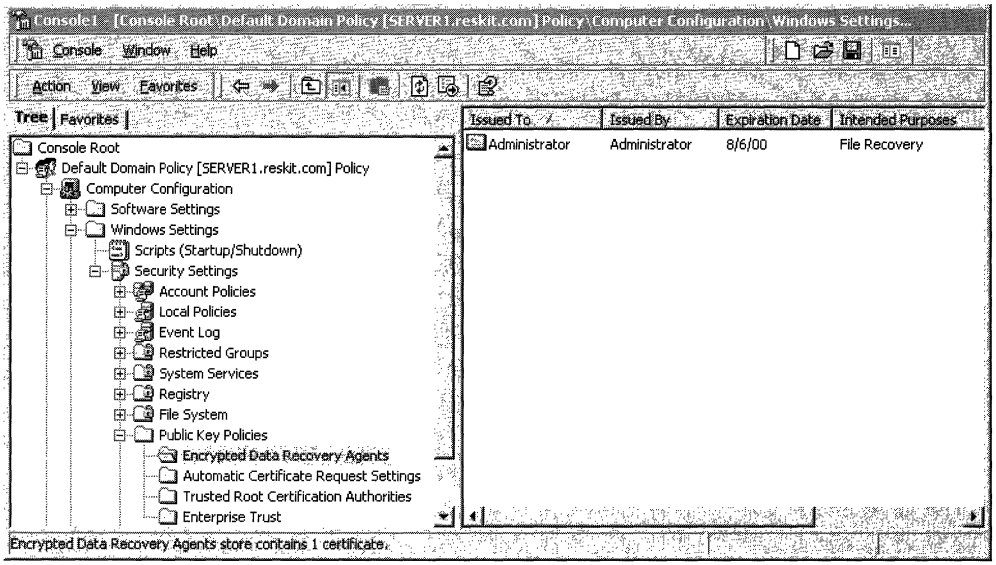


Figure 15.12 Recovery Agent Certificates in Recovery Policy

For more information about how to access the **Encrypted Data Recovery Agents** container and Group Policy, see "Windows 2000 Certificate Services and Public Key Infrastructure" in this book.

How Private Keys Are Stored

Private keys for the Microsoft RSA-based CSPs, including the Base CSP and the Enhanced CSP, reside in the user profile under *RootDirectory*\Documents and Settings\<*username*>\Application Data\Microsoft\Crypto\RSA. In the case of a roaming user profile, the private key resides in the RSA folder on the domain controller and is downloaded to the user's computer until the user logs off or the computer is restarted.

Unlike their corresponding public keys, private keys must be protected. Therefore, all files in the RSA folder are automatically encrypted with a random, symmetric key called the user's master key. The user's master key is generated by the RC4 algorithm in the Base or Enhanced CSP. RC4 generates a 128-bit key for computers with the Enhanced CSP (subject to cryptography export restrictions) and a 56-bit key for computers with only the Base CSP (available for all Windows 2000 computers). The master key is generated automatically and is renewed periodically. It encrypts each file in the RSA folder automatically as the file is created.

The RSA folder must never be renamed or moved because this is the only place the CSPs look for private keys. Therefore, it is advisable to provide additional security. The administrator can provide additional file system security for users' computers or use roaming profiles.

You should protect private keys for recovery, which is critical for backup, by exporting the certificate and private key to a floppy disk or other medium, storing the floppy disk or other medium securely, and then deleting the private key from the computer. This preserves the file from a system crash and makes it unavailable for cracking. To decrypt a data file, the recovery agent administrator inserts the floppy disk or other medium and imports the certificate and private key to the recovery agent account. For more information about how to secure recovery keys, see Windows 2000 Server Help.

Protect Folder

The user's master key is itself encrypted automatically by the Protected Storage service and stored in the user profile under *RootDirectory*\Documents and Settings\<*username*>\Application Data\Microsoft\Protect. For a domain user who has a roaming profile, the master key resides on the domain controller and is downloaded to the user's profile on the local computer until the computer is restarted.

The user's master key is encrypted twice, and each instance of encryption is stored in one of two parts of the Protect file. The first part, the password encryption key, is produced by the Hash-Based Message Authentication Code (HMAC) and SHA1 message digest function and is a hash of:

- A symmetric encryption of the user's master key produced by 160-bit RC4.
- The user's security identifier (SID).
- The user's logon password.

The second part is the backup/restore form of the master key. This is needed if the user's password is changed on one computer but the keys are in the user profile on another computer, or if the administrator resets the user's password. In either case, the Protected Storage service, which cannot detect password changes to update Part 1, uses Part 2 to recover the master key and regenerate Part 1.

To create the backup part of the file, the encrypted user's master key is sent on to the Protected Storage service on the domain controller. That service uses HMAC and SHA1 again to make a hash of the data it has received along with the domain controller's own backup/restore master key, and sends that back to the user's computer to store in the Protect file. These transmissions are authenticated (signed and encrypted) by way of remote procedure calls so that the user's master key never goes over the wire in plaintext.

The domain controller's backup/restore master key is stored on the system as a global Local Security Authority (LSA) secret in the HKEY_LOCAL_MACHINE/SAM key in the registry and is replicated over the network by means of Active Directory. (Global LSA secrets are objects provided by the LSA to enable system services to store private data securely.)

Caution Do not use a registry editor to edit the registry directly unless you have no alternative. The registry editors bypass the standard safeguards provided by administrative tools. These safeguards prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your system. Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. To configure or customize Windows 2000, use the programs in Control Panel or Microsoft Management Console (MMC) whenever possible.

The System Certificates, RSA, and Protect folders have their system attributes set. This prevents the files in them from being encrypted by EFS, which would make them inaccessible.

How Encryption Keys Are Protected

When an encrypted file is saved, Windows 2000 automatically provides five levels of encryption:

1. EFS provides the FEK, which encrypts the data in the file.
2. EFS uses the public key in the user's EFS certificate and the public keys in the recovery agents' certificates to encrypt the FEK. The public keys and certificates are stored by default in the certificate stores for each computer. The corresponding private keys, to be used for decrypting the FEK, are stored in an encrypted form in the user profiles for the corresponding user or recovery administrator account in the RSA folder.
3. The Protected Storage service generates the user's master key that is used to encrypt the user's private key.
4. The Protected Storage service generates a symmetric password encryption key, derived from a hash of the file creator's credentials, that encrypts the user's master key. It also generates a second such key, the backup/restore key, that incorporates an additional hash derived from the backup/restore master key on the domain controller.
5. The system key can be used to optionally protect all master keys as well as a variety of other secrets that are stored on computers. At system startup, Windows 2000 obtains the system key and uses it to decrypt all the private keys on the computer, including private keys that are used for EFS.

Planning EFS Implementation

When you are planning to install EFS, it is useful to analyze the threats to your system, how EFS handles these threats, and what certification authority (CA) you should deploy, if any.

Security Analysis

Following are the kinds of questions that administrators frequently ask when analyzing the value of EFS to a file security plan.

Is there a threat of users attempting to open other users' encrypted files?

EFS is designed to be transparent in normal operation. When a user attempts to open a file that has been encrypted by another user, EFS attempts to locate the private key that decrypts the FEK so the file can be opened. If the user who is calling does not possess the key, the FEK is not decrypted, and the attempt fails with an "Access denied" error message.

Is it possible to bypass recovery policy?

Not if the computer is joined to a domain and the policy is set at the domain level. Encrypted Data Recovery Agent policy is propagated from the domain as part of Group Policy and is enforced by EFS on the local computer. An attempt by a local administrator to define a different local EFS policy does not work because policy from the domain takes precedence.

The only option for a local administrator is to remove the computer from the domain, but this means that users cannot log on with domain credentials, and therefore cannot gain access to files that were encrypted by using domain recovery policy. Users might log on with other credentials and encrypt some new files, but if the computer is then reconnected to the domain, the new files are inaccessible using domain credentials.

Is it possible to destroy recovery policy?

A local administrator might attempt to locate the EFS policy storage and delete or replace it. Deletion is ineffective because that disables EFS. Replacement with another recovery policy does not work because domain policy has precedence and overwrites the other policy in each file the next time the file is opened.

Are the media physically accessible?

Someone with physical access to the computer might attempt a sophisticated attack directly on the file stored on the hard disk drive. This fails because the file contains no plaintext and no key that can convert it to plaintext by itself. It cannot be read without using the user's private key.

Another possible attack is to invalidate or delete the DRF. This does not work either because EFS refreshes the recovery information the next time the file is accessed.

Is there a danger of fatal failure during encryption or decryption operations?

EFS incorporates a crash recovery scheme to preserve data in the event of fatal errors such as system crash, full disk, or hardware failure. This is accomplished by creating a plaintext backup of the original file being encrypted or decrypted. After the original is encrypted or decrypted successfully, the backup is deleted.

Encryption and decryption are safe operations because if any one step fails, all previous steps are rolled back. If a fatal failure causes the computer to stop responding or to restart, the backup file is used to perform the rollback when the operating system is up again.

Windows 2000 Backup detects the encryption and retains the encryption attributes when it backs up encrypted files.

Note The plaintext copy can persist on the disk for some time, until another file uses those disk blocks. To avoid this problem, always start by creating an empty encrypted folder and then create files directly in that folder. The files are encrypted at their creation, and there is never any need to save them in plaintext anywhere on the disk.

Can recovery policy be changed from time to time?

A domain administrator might want to change the recovery policy for various reasons, such as the expiration of certificates or a change in recovery agent accounts. EFS does not check whether the recovery information on a particular encrypted file is changed until the next time that file is opened. If the information is not current, it is updated at that point.

This file-by-file approach is necessary because recovery information for a file cannot be updated without a decrypted FEK, which becomes available only when the file is opened.

This means that encrypted files that are not opened for long periods might have an old recovery policy. It is therefore important that recovery certificates and private keys be maintained for several years, even after the recovery policy has changed.

Can user certificates or keys be changed from time to time?

Just like changes in recovery policy, user certificate and key changes are handled when a particular file is opened. EFS determines whether the user's public key and certificate are current. If not, the DDF is updated with the user's current public key. To update the DDF, EFS must first find the old public key and certificate. For expired certificates, the certificates and private keys are archived automatically by the system, so they are available. Users, however, have the option to delete archived certificates and private keys. Users must retain their old keys until they are sure that all files that have been encrypted have been opened since the change was made. If a user's archived certificates and private keys are damaged or destroyed, you can use a recovery agent certificate and private key to recover the user's encrypted files. Thus, it is important to keep secure copies of recovery agent certificates and private keys in archives even after the certificates expire.

Is there a threat of not being able to start the system?

System data such as that stored in the registry, system DLLs, and other files that are required during system startup must never be encrypted because EFS does not become active until the operating system is up. If any of the files used by the operating system are encrypted, the system is rendered useless. EFS provides protection by disallowing encryption of files and folders with the system attribute.

Best Practices

Encryption is a sensitive operation. It is important that encrypted data not become decrypted inadvertently. To this end, it is recommended that users do the following:

- Encrypt the My Documents folder (*RootDirectory**UserProfile*\My Documents). This ensures that personal folders where most Microsoft® Office documents are saved are encrypted by default.
- Encrypt the Temp folder (*RootDirectory*\Temp). This ensures that any temporary files created by various applications are encrypted. This avoids any possible leaks.
- Encrypt folders rather than individual files. Applications work on files in various ways—for example, creating temporary files in the same folder during editing. These temporary files might or might not be encrypted, and some applications substitute them for the original when the edit is saved. Encrypting at the folder level ensures that files do not get decrypted transparently in this way.

- Export the private keys for recovery accounts, store them in a safe place on secure media, and remove the keys from computers. This prevents someone from using the recovery account on the computer to read files that are encrypted by others. This is especially important for stand-alone computers where the recovery account is the local Administrator or another local account. For example, someone might steal a portable computer that contains encrypted files. However, because the private key for recovery is not on the computer, the thief cannot log on as the recovery account and use it to recover files.
- The private keys associated with recovery certificates are extremely sensitive. Never leave them lying around. Export each such key into a .pfx file, protected under a strong password, and secure that file on a floppy disk. For more information about protecting recovery keys, see Windows 2000 Professional Help or Windows 2000 Server Help.
- Do not use the recovery agent account for any other purpose.
- Do not destroy recovery certificates and private keys when recovery agent policy changes. Keep them in archives until you are sure that all files that are protected by them have been updated with new recovery agent information.
- Never rename or move the RSA folder because this is the only place EFS looks for private keys.
- In a domain, change the default recovery agent account (the Administrator of the first domain controller installed for the domain) as soon as possible, and set a password for each recovery agent account. This adds an extra layer of protection in case the Administrator account is hijacked, and provides easy tracking of usage of the recovery account.
- Designate two or more recovery agent accounts per organizational unit (a subgroup of computers, or even a single computer, within a domain), depending on the size of the organizational unit. Designate one computer for each designated recovery agent account, and give permission to appropriate administrators to use the recovery agent accounts.
- Implement a recovery agent archive program to ensure that EFS files can be recovered using obsolete recovery keys. Recovery certificates and private keys must be exported and stored in a controlled and secure manner. It is recommended that you store archives in a controlled-access vault, and that you have a master archive and a backup archive. The master archive is located onsite; the backup archive is located in a secure offsite location.
- When you are printing, avoid making a spool file. If you must, ensure that it is created in an encrypted folder.
- Configure system key for stand-alone computers that are not members of a domain to provide system key protection for the EFS users' private keys. For more information, see "Using the System Key" later in this chapter.

Recovery Policy

EFS recovery policy specifies the data recovery agent accounts that are used within the scope of the policy. EFS requires an Encrypted Data Recovery Agent policy before it can be used, and uses a default recovery agent account (the Administrator) if none has been chosen. In a domain, only members of the Domain Admins group can designate another account as the recovery agent account. In a small business or home environment where there are no domains, the computer's local Administrator account is the default recovery agent account. Only the Administrator account can change local recovery policy for a computer.

A recovery agent account is used to restore data for all computers covered by the policy. If a user's private key is lost, a file protected by that key can be backed up, and the backup sent by means of secure e-mail to a recovery agent administrator. The administrator restores the backup copy, opens it to read the file, copies the file in plaintext, and returns the plaintext file to the user using secure e-mail again.

As an alternative, the administrator can go to the computer that has the encrypted file, import his or her recovery agent certificate and private key, and perform the recovery there. However, this might not be safe and is not recommended because of the sensitivity of the recovery key—the administrator cannot afford to leave the recovery key on another computer.

Policy Implementation

Encrypted Data Recovery Agent policy is part of Group Policy, administered using the Encrypted Data Recovery Agents container in the Group Policy console. To configure EFS Recovery policy, see "EFS Recovery Agents" in "Windows 2000 Certificate Services and Public Key Infrastructure" in this book.

Like other components of Group Policy, the scope of a recovery policy can be from broadest to narrowest:

- The site
- The domain
- An organizational unit—a subgroup of computers, or even a single computer, within a domain.
- An organizational unit nested within a larger organizational unit.
- The local stand-alone computer.

The default scope of Group Policy is the domain. Every Windows 2000–based computer within a scope is governed by the policy set for that scope, although the effect can be filtered by using ACLs. For more information about Group Policy and how it works, see "Group Policy" in this book.

Note Policy is applied per computer, not per user, because the encrypted data is stored on computers regardless of who encrypted it.

The Windows 2000 security subsystem takes care of enforcing, replicating, caching, and updating the recovery policy. Therefore, users are able to use file encryption on a temporarily offline system, such as a portable computer, much like logging on to their domain account using cached credentials.

Only the public portions of the recovery key pairs are needed for encryption of the FEK in the DRF. These public recovery keys must be present at all times on an EFS system for normal file system operations. They are present in Encrypted Data Recovery Agent policy as recovery agent certificates.

Security of the recovery key is crucial. You should remove the private keys of recovery agent accounts from the recovery computer between sessions and store them safely on floppy disks or other secure storage devices. (For more information about how to safely remove private keys from your computer, see Windows 2000 Professional Help or Windows 2000 Server Help.)

Policy Enforcement

EFS enforces policy each time an encrypted file is opened. The existing recovery information is checked to ensure that it is based on current policy. If it is not, new recovery information is generated for the file. This keeps the recovery information up to date on all files that are being used. To perform recovery, you can view EFS recovery policy to see which recovery agent administrators you can send the file to for recovery. If the file has not been opened for a long time, the recovery agent information might be obsolete. You can use the **efsinfo** command to display the recovery agent information contained in encrypted files. For more information about **efsinfo**, see "Viewing Recovery Agent Information" later in this chapter.

You can assign one or more recovery agent accounts to a group of computers within a domain by using Group Policy to designate that group as an organizational unit. Any of the designated recovery accounts can be used to recover users' files for that organizational unit.

A policy with no recovery certificates—called an empty policy—disables EFS on all computers in its scope. An empty policy is distinct from *no* policy, where the recovery agent certificates are deleted from EFS recovery policy. "No policy" implies "don't care" and, therefore, each computer can use its locally defined policy.

By EFS rules, a recovery policy with any invalid certificate is invalid as a whole, and EFS is turned off for any new encryptions. Existing encrypted files can still be decrypted.

By default, Group Policy is inherited and cumulative, and is applied from the broadest scope to the narrowest. When a user logs on to the domain controller, the site policy is applied first, then the domain policy, and finally the organizational unit policy. If the computer has a local policy, it applies only in stand-alone mode and is superseded by the organizational unit policy when the computer is joined to a domain. The order of policy importance is the reverse: If there are contradictions in policy between the site, domain, or organizational unit, the policy that takes precedence is the one applied at the scope closest to the user.

For an example, let's say the site recovery agent account is RECOV1 and neither the domain nor the organizational unit that you are joining has a policy.

- You log on to the network and join the domain. RECOV1 is applied, then "no policy" and another "no policy." There is no conflict here, so RECOV1 automatically becomes your recovery agent account.
- Now let's say the domain has a designated recovery agent account, RECOV2. RECOV2 takes precedence over RECOV1 and becomes your recovery agent account.
- Let's say that, instead of "no policy," the organizational unit has an empty policy. This takes precedence and disables EFS, so you cannot use it.
- Suppose LOCRECOV is the stand-alone recovery agent account for the computer. It is ignored because the organizational unit policy takes effect when you join the domain. If you leave the domain by removing your account from the domain controller, LOCRECOV becomes the agent again.

When you configure Group Policy, you have the following options:

- Block policy inheritance, so lower-level organizational units do not inherit the policy from the parent site, domain, or organizational units.
- Prevent a lower-level policy from overriding or superceding policy from a parent site, domain, or organizational unit.

For more information about Group Policy, see Group Policy Help and "Group Policy" in this book.

You can associate multiple Group Policy objects with a single site, domain, or organizational unit; and you can prioritize how these Group Policy objects affect the directory object to which they are applied. Conversely, multiple sites, domains, and organizational units can use a single Group Policy object. Any site, domain, or organizational unit can be associated with any Group Policy object, even across domains in the same forest.

Administrators can exempt certain computers from inheriting Group Policy from the domain to which they belong. In such cases, with regard to Group Policy, the computer functions as if it were a stand-alone computer.

Active Directory can accommodate a wide range of implementation scenarios. A computer can exist in multiple nested organizational units and have more than one Group Policy object applied to it. Because the directory is stored on all domain controllers, it scales to organizations with many thousands of users.

Storage

EFS stores the recovery agent information for EFS recovery policy in Active Directory as part of Group Policy. All computers within the scope of a Group Policy are required to enforce that policy. For stand-alone computers, EFS recovery policy information is stored in Local Group Policy. This means that, in a domain, only domain administrators can control the recovery keys and in stand-alone mode, local administrators can.

The EFS service initializes during system startup as part of the Local Security Authority Subsystem (LSASS). LSASS is responsible for getting the EFS policy information in memory, either from the domain or locally.

For stand-alone computers, EFS recovery policy information is stored in local policy. The key is created by the EFS service, and the security on the key allows only System Full Control. This partially ensures that no one other than local administrators can gain access to this information directly. To guard against accidental corruption of data, the administrators are not given direct access. This is also the place where EFS recovery policy information from the domain is cached to handle situations caused by network problems.

Note When creating EFS recovery policy, as with any Group Policy object, it is a good idea to make a note of the date and action. Group Policy is not automatically backed up.

Certificates

For EFS to work, the file's encryptor must have a valid EFS user certificate, and the current EFS recovery policy must specify at least one valid recovery agent certificate. If available, EFS requests certificates from a Windows 2000 enterprise CA, but EFS does not require a CA to issue certificates. If an enterprise CA is not available, EFS automatically generates its own certificates to users and to default recovery agent accounts.

Note Certificates that EFS generates are self-signed rather than signed by a CA. Therefore, the certification path is the same as for root CA certificates, which are also self-signed. EFS certificates that are self-signed are identified by Windows 2000 as "not trusted" because the certifying authority does not have a certificate in the Trusted Root Certification Authorities store. Nevertheless, self-signed EFS certificates are valid for use by EFS. For more information about the certification path, see "Windows 2000 Certificate Services and Public Key Infrastructure" in this book.

For more information about certificates, see "Cryptography for Network and Information Security" in this book. For more information about Windows 2000 Certificate Services and CAs, see "Windows 2000 Certificate Services and Public Key Infrastructure" in this book.

User Certificates

Certificates with an object identifier of 1.3.6.1.4.1.311.10.3.4 in the **Enhanced Key Usage** field of the certificates are valid for EFS user operations. If an enterprise CA is available, EFS automatically requests an EFS certificate for users the first time that they encrypt a file or a folder. When the request is approved, the certificate is issued and placed in the user's personal certificate store. If no CA is available, EFS generates an EFS certificate and places it in the user's personal certificate store. EFS operations require users to have a valid EFS certificate in their personal certificate store. If the EFS user certificate has expired, EFS ensures that a new certificate is issued for the user with a new public key pair the next time an EFS operation is performed for that user.

You can deploy Certificate Services to issue and manage certificates for EFS users. Certificates that are issued by enterprise CAs are based on certificate templates. Certificate templates are stored in Active Directory, and define the attributes of certificate types to be issued to users and computers. There are many certificate templates, but only the following three certificate types support EFS user operations:

- User
- Administrator
- Basic EFS

When a user has a valid certificate of one of these types, EFS uses it for EFS operations. Administrator and user certificates combine a number of certificate uses in one certificate, including EFS. A basic EFS certificate can be used for EFS operations only.

Enterprise CAs use ACLs for certificate templates to determine whether to approve certificate requests. By default, members of the Domain Admins and Domain Users security groups have Enroll permission for basic EFS certificates and user certificates. By default, members of the Domain Admins and Enterprise Admins security groups have Enroll permission for administrator certificates.

For more information about certificate templates, see "Windows 2000 Certificate Services and Public Key Infrastructure" in this book.

Recovery Agent Certificates

Certificates with an object identifier of 1.3.6.1.4.1.311.10.3.4.1 in the **Enhanced Key Usage** field of the certificates are valid for EFS recovery agent operations. EFS automatically generates its own certificates for the default recovery agent accounts: the domain Administrator account for the first domain controller installed in the domain and the local Administrator account for stand-alone computers. The default recovery agent certificates are placed in the personal certificate store for the Administrator account. To recover data, a valid recovery agent certificate and private key must be installed on the computer where the recovery takes place. EFS recovery policy is valid only if all recovery certificates are valid.

If you want to designate alternate recovery agent accounts (other than the default recovery accounts), the alternate recovery accounts must have valid EFS recovery agent certificates. You can deploy Certificate Services to issue and manage EFS recovery agent certificates.

Windows 2000 includes the EFS Recovery Agent certificate template for use by enterprise CAs. EFS recovery agent certificates can be used for recovery agent operations only. By default, members of the Domain Admins and Enterprise Admins security groups have Enroll permission for EFS recovery agent certificates. However, if you want to change the accounts that are authorized to receive EFS recovery agent certificates, you can change the default ACLs for the certificate template. For more information about how to modify ACLs for certificate templates, see "Windows 2000 Certificate Services and Public Key Infrastructure" in this book.

Stand-alone CAs do not use certificate templates, but can also issue EFS certificates. You must use the Advanced Certificate page Web form to request certificates from a stand-alone CA. For more information about how to request EFS certificates from a stand-alone CA, see "Windows 2000 Certificate Services and Public Key Infrastructure" in this book.

Administrative Procedures

You can perform the following administrative procedures to set up and secure EFS for your organization:

- Secure the default recovery key.
- Assigning alternate recovery agent accounts.
- Configuring recovery agent policy.
- Viewing recovery agent information.
- Recovering an encrypted file or folder.
- Disabling EFS for a set of computers.
- Disabling EFS for a specific folder.

Securing the Recovery Key

As mentioned earlier in this chapter, it is recommended that you remove private keys for recovery agent accounts from the computers by exporting the keys to removable media and then putting the keys in locked storage. This should be done with the default recovery keys before any changes are made to recovery policy.

The Certificate Export wizard accomplishes this purpose. This wizard is available through the Certificates console (a Microsoft Management Console snap-in). For more information about using the Certificates console and the Certificate Export wizard, see "Windows 2000 Certificate Services and Public Key Infrastructure" in this book.

If you are securing the recovery key for a stand-alone computer, log on as Administrator. The EFS recovery agent certificate is contained in the personal certificate store for the Administrator account.

If you are securing the recovery key for a domain, log on as Administrator on the *initial* domain controller created for the domain. The EFS recovery agent certificate is contained in the personal certificate store for the Administrator account of the first domain controller installed for the domain.

Use the Certificate Export wizard to export the certificate and private key to a removable medium. For information about how to export a certificate and its private key, see Certificates Help, and see "Windows 2000 Certificate Services and Public Key Infrastructure" in this book.

To delete the private key from the computer, you must select the **Delete the private key if the export is successful** check box on the Export File Format page of the wizard. When you have completed the wizard, the private key is deleted from the computer and the recovery agent certificate and private key resides in a .pfx file in the folder or drive that you have specified. Now you need to protect the .pfx file by putting it into secure storage.

▶ **To protect a .pfx file**

1. If you created the .pfx file on a floppy disk, the file is right where it should be —on a medium that can be physically removed and locked away in another location. If you did not create the .pfx file on a floppy disk, copy it to a floppy disk and delete it from your hard disk drive.
2. Remove the floppy disk and make a backup copy of the .pfx file on another floppy disk. Store both floppy disks in safes or in a secure place. One floppy disk should be stored in a secure offsite location.

You then can use the Certificates console to import the .pfx file to a recovery computer and perform recovery operations. After recovering encrypted files, secure the private key again.

An alternative to securing the private key on removable storage media is to use physically secured stand-alone computers for recovery operations and leave the private key for recovery on the computer. You then log on to the recovery agent account and use the secure computer for data recovery only. It is important, however, that you keep a backup of the certificate and private key so you can restore them to the recovery computer if necessary. You can use the Certificate Export wizard to create a backup of the recovery agent certificate and private key, but make sure the **Delete the private key if the export is successful** check box is cleared before completing the wizard. You cannot use the computer for recovery if the private key is deleted.

You can also store the recovery agent certificate and private key on a smart card. You must map the smart card certificate to the designated recovery account by using the certificate mapping feature of the Active Directory Users and Computers console (a Microsoft Management Console snap-in). You then can perform recovery operations with the certificate and private key that are stored on the smart card. You can perform recovery operations from any Windows 2000–based computer in the domain that has a smart card reader. It is a good idea to maintain a secure backup of the recovery agent certificate and private key in case the smart card is damaged or ceases to work properly. As long as you have the recovery certificate and private key in archives, you can import the certificate and key to a computer and use it for recovery operations. For more information about storing certificates on smart cards, see "Windows 2000 Certificate Services and Public Key Infrastructure" and "Choosing Security Solutions That Use Public Key Technology," in this book. For more information about certificate mapping, see "Windows 2000 Certificate Services and Public Key Infrastructure" in this book.

Assigning Recovery Agent Accounts

Rather than attempt to manage EFS recovery on a domainwide basis, consider assigning dedicated recovery computers to manage recovery for subsets of computers in your domain, or even for single computers. Domain administrators can do this by using the Active Directory Users and Computers console to group computers into organizational units, and then configuring a separate EFS recovery policy for each organizational unit. You might want to appoint several administrators to use one recovery account to recover users' files as necessary for that organizational unit.

Although recovery policy can be set to apply to an organizational unit, it must be set at the domain level. Subdomain administrators can view recovery agent policy, but cannot set or modify the policy.

To use Group Policy for this purpose, install the Group Policy MMC snap-in and open the Group Policy object you want to work with (domain, organizational unit, or local computer), and then follow the procedure described here.

▶ To use Group Policy to delegate recovery

1. Expand the Group Policy node by clicking **Computer Configuration** and then **Windows Settings**, **Security Settings**, and **Public Key Policies**.
2. Right-click **Encrypted Data Recovery Agents**, and then click **Add**.

   This opens the Add Recovery Agent wizard. Figure 15.13 shows the opening screen of the wizard.

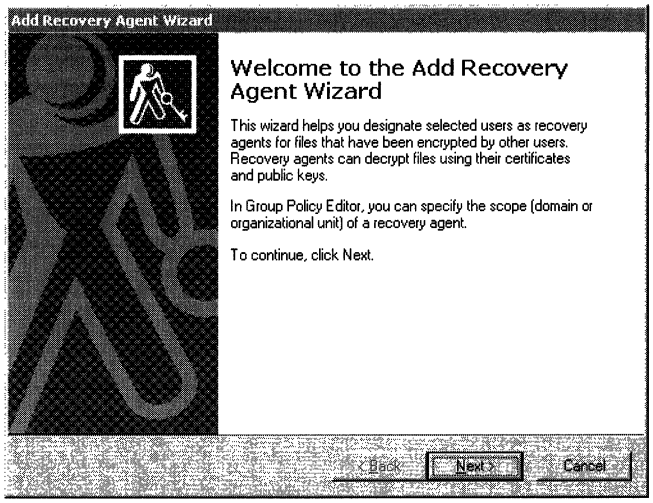


Figure 15.13 Welcome Screen in Add Recovery Agent Wizard

3. Click **Next**.
4. Use the second screen to add recovery agent certificates. If the recovery certificates are published in Active Directory, use the **Browse Directory** option. Otherwise, you can use the **Browse Folders** option.

5. Repeat this step to add as many recovery agent certificates as required. Figure 15.14 shows a typical display of the second screen after certificates are selected.

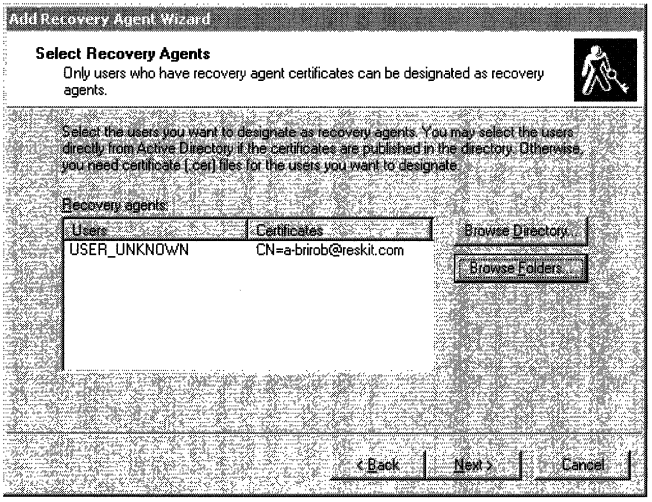


Figure 15.14 Second Screen in Add Recovery Agent Wizard

You can add recovery agent certificates that are published in Active Directory. The recovery agent user account information associated with the published certificates appears in the **Users** column.

You can also add a recovery agent certificate from a file. If so, the **Users** column displays "USER_UNKNOWN." This is because adding the certificate from a file does not provide any security identifier (SID) information about the owner of the private key.

Note The Add Recovery Agent wizard accepts a recovery agent certificate file only if it has a .cer extension. You can import certificates to the local computer using the Certificate Request wizard, as described in Certificates Help.

---

6. When you have added all of the recovery agent certificates, click **Next**, and then click **Finish**.

Configuring Recovery Agent Policy

For more flexible EFS recovery management, consider issuing EFS recovery certificates to designated recovery agent accounts, besides the default Administrator account. Also, legal or corporate policy might require that the recovery agent account be different from the domain Administrator account. You can also configure EFS recovery policy for portable computers to use the same recovery agent certificates, whether the computers are connected to domains or are operated as stand-alone computers.

For more information about how to configure local recovery agent policy on a stand-alone computer or at the local computer level in a domain, see Windows 2000 Professional Help or Windows 2000 Server Help. For more information about how to configure recovery agent policy at the domain controller level, see Windows 2000 Server Help.

You can request recovery agent certificates from either an enterprise CA or a stand-alone CA. You must be logged on as a member of the Domain Admin security group to request the EFS recovery agent certificate from enterprise CAs. You can use the Certificate Request wizard or the Web Enrollment Support pages to request certificates from an enterprise CA. You can use the Web Enrollment Support pages to request certificates from a stand-alone CA. (Certificate requests to stand-alone CAs are held as "pending" until approved by the CA administrator.) For more information about requesting certificates, see "Windows 2000 Certificate Services and Public Key Infrastructure" in this book.

When the recovery agent certificate is installed in the personal certificate store of the requesting account, you can export the certificate and its private key (as described earlier in this chapter), and then import them for the designated recovery agent accounts. You also then add the certificates to the appropriate EFS recovery policies.

Viewing Recovery Agent Information

The Windows 2000 Resource Kit includes the Efsinfo.exe tool, which you can use to view information about EFS files, including information about the EFS user account and the recovery agent accounts.

To view user information for an encrypted file, type the following at the command prompt:

efsinfo /u <*filename*>

The user name and e-mail address of the file's encryptor are then displayed.

To view recovery agent information for an encrypted file, type the following at the command prompt:

efsinfo /r <*filename*>

The user names and e-mail addresses of the file's recovery agent accounts are then displayed.

The information that is displayed by Efsinfo.exe is from the user's EFS certificate or the recovery agent certificates. When certificates are created by an enterprise CA, the CA obtains the user's information from the certificate requestor's user account in Active Directory. For stand-alone CAs, the user information is not obtained from Active Directory; you must instead enter the user name and the e-mail information in the Advanced Certificate Request page Web form when you submit the certificate request.

You can use Efsinfo.exe to verify who the file encryptor is or to verify which recovery accounts are authorized for recovering the file. This is especially important for files that have not been opened for a long time and thus do not have current user and recovery agent information.

For more information about how to use Efsinfo.exe, see Tools Help.

Recovering a File or Folder

The process for recovering an encrypted file or folder when users have lost their private keys includes the following:

1. The user can use a secure protocol, such as Secure/Multipurpose Internet Mail Extensions (S/MIME) encrypted mail, to send the file or folder to the recovery administrator. Or the user can use Windows 2000 Backup to back up the encrypted file and send the backup file as a regular e-mail attachment.

2. The recovery administrator decrypts the file or folder by running **cipher** from the command line. (To use this, the recovery agent certificate and private key must be installed on the recovery computer and the administrator must be logged on as the recovery agent account.)
3. The administrator makes a backup of the plaintext file and sends it back to the user with a secure protocol, such as S/MIME encrypted mail.

If the administrator has followed the procedure described earlier under "Securing the Recovery Key," the recovery agent account's certificate and public key are offline and securely stored in a .pfx file. To use the certificate on a recovery computer, you must import the certificate into the personal certificate store for the designated recovery account. For more information about importing certificates, see Certificates Help.

After you are done using the certificate for file recovery, delete it from the hard disk. There is no need to export it again because it remains on the removable medium.

Note that in this process the private key for recovery always stays on a designated recovery computer. The recovery agent administrator could bring his or her private key to the owner's computer, but it is not a good security practice to copy a private key on another computer.

Disabling EFS for a Specific Set of Computers

A particular computer must have at least one valid recovery agent certificate to enable EFS. EFS does not allow encryption of data if no recovery agent certificate is specified by recovery agent policy. Therefore, you can disable EFS by setting either *no* recovery policy (where the policy is removed from the computer) or an *empty* recovery policy (where the policy remains, but the recovery agent certificates are deleted). These are applied as follows:

- Both *no policy* and *empty policy* disable EFS on a stand-alone computer.
- Both *no policy* and *empty policy* are ineffective in disabling EFS on the local computer in a domain if there is a policy at a higher level, such as a domain or organizational unit.
- Applying *no policy* at a higher level disables EFS at that level only. The lower-level computers use their own local policies.
- Applying *empty policy* at a higher level disables EFS at that level and all lower levels.

▶ **To set no recovery policy**

1. On a stand-alone computer, open the MMC and add the Group Policy snap-in for the local computer.
2. In the Group Policy console, right-click Encrypted Data Recovery Agents, and then click Delete Policy.
3. Answer **Yes** when the system prompts you with the question **Are you sure...?** The details pane of the window displays the message "There is no policy defined."

If a domain administrator wants to disable EFS for all the computers in a domain or an organizational unit, the best way is to set an empty recovery policy. This is because the effective policy is an accumulation of Group Policy objects that are defined at various levels in the directory tree. The absence of a recovery policy at the domain or organizational unit level allows policies at a lower level to take effect. However, an empty recovery policy at these higher levels disables EFS by providing no effective recovery certificates and blocking the individual computers from using lower-level policies.

▶ **To set an empty policy at the domain or organizational unit level**

1. Log on as Administrator of the initial domain controller created in the domain and display the certificate listings in the details pane of the window.
2. Right-click **Administrator** and any other certificate that might be listed in the details pane, and then click **Delete**.
3. Answer **Yes** to the question **Permanently delete the selected certificate?**

▶ **To re-enable EFS on the local computer**

1. Restore recovery policy by right-clicking **Encrypted Data Recovery Agents** and then clicking **Initialize Empty Policy**.
2. After you have an empty policy, to re-enable EFS, you must add a policy by right-clicking **Encrypted Data Recovery Agents** and clicking **Add**. This starts the Add Recovery Agent wizard. The Add Recovery Agent wizard accepts a recovery agent certificate file only if it has a .cer extension.

▶ **To re-enable EFS on the domain or organizational unit**

- Add one or more valid recovery agent certificates to EFS recovery policy by following the procedure in "Designate Assigning Recovery Agent Accounts" earlier in this chapter.

Disabling EFS for a Specific Folder

You can disable EFS on any folder by using Windows Explorer to set the System attribute to "on." This takes advantage of the safety precaution that prevents the encryption of files that are needed for the system to start. There are some folders, however, that cannot be marked as system folders. An example is the Profiles folder, which contains Ntuser.dat files.

▶ **To disable EFS on folders that cannot be marked as system folders**

- Use the Winadvapi API, which has the following format:

```
WINADVAPI
BOOL
EncryptionDisable(
    IN LPCWSTR DirPath,
    IN BOOL Disable
    )
/*++
```

This routine disables and enables EFS in the directory DirPath.

Arguments are:

```
    DirPath - Directory path.
    Disable - TRUE to disable (FALSE to enable)
```

The return value is:

```
TRUE for SUCCESS
--*/
```

The header file for this is Winefs.h. This API disables or enables EFS in the folder that is passed in. It creates a Desktop.ini file with these lines:

```
[Encryption]
Disable=1;  (or Disable=0)
```

If Desktop.ini already exists, the API adds these lines to the file. Manually adding the lines to a Desktop.ini file has the same effect.

Using the System Key

You can provide another level of protection for master keys and various other secrets through use of the system key. The system key protects the following sensitive information:

- Master keys that are used to protect private keys
- Protection keys for user account passwords stored in Active Directory
- Protection keys for passwords stored in the registry in the local Security Accounts Manager (SAM) registry key
- Protection keys for LSA secrets
- The protection key for the administrator account password that is used for system recovery startup in safe mode

For all computers in a domain, the secret key is enabled by default and all master keys and protection keys stored on a computer are encrypted with the unique 128-bit symmetric random system key. The system key must be in volatile memory on the operating system during system startup to unlock the password protection key. There are three ways to configure the system key for computers:

- Use a computer-generated random key as the system key and store it on the local system by using a complex obfuscation algorithm that scatters the system key throughout the registry. This option allows you to restart the computer without having to enter the system key. This is the default configuration for the system key.
- Use a computer-generated random key, but store it on a floppy disk. The system key is not stored anywhere on the local computer, and the floppy disk must be inserted for the system to start. It is inserted when prompted after Windows 2000 begins the startup sequence, but before it is available for users to log on to the system.
- Use a password chosen by the administrator to derive the system key. The password is not stored anywhere on the computer. Windows 2000 prompts the administrator for the password when the system is in the initial startup sequence, but before the system is available for users to log on.

The system key configuration options are available from the system key dialog boxes that appear when you run **syskey**. For computers in a domain, you must be a member of the Domain Admin group to run **syskey**. For stand-alone computers, you must be logged on as the local Administrator to run **syskey**. You can configure the system key differently for each computer in the domain.

System key protection is enabled by default in each domain, but you might want to change the default system key option for various computers in a domain. You also might need to enable system key protection for stand-alone computers.

▶ To configure system key protection

1. Type **syskey** at the command prompt. This brings up the dialog box shown in Figure 15.15.

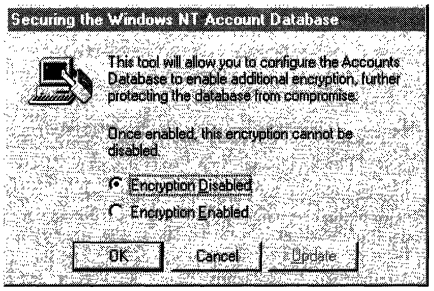


Figure 15.15 System Key Dialog Box

After system key protection is enabled, it cannot be disabled.

2. If it is not already selected, click **Encryption Enabled**, and then click **OK**. After a reminder that you should create an updated emergency repair disk, you are presented with options for the Account Database Key as shown in Figure 15.16. The default option is a system-generated password that is stored locally.

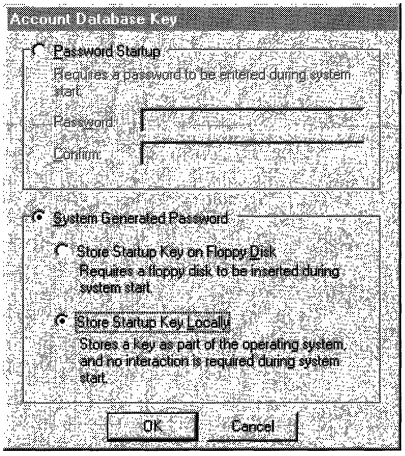


Figure 15.16 Account Database Key Dialog Box

3. Select the system key option that you want, and then click **OK**.
4. Restart the computer.

When the system restarts, you might be prompted to enter the system key, depending on the key option you chose. Windows 2000 detects the first use of the system key and generates a new random password encryption key. The password encryption key is protected with the system key, and then all account password information is strongly encrypted.

At subsequent startups:

1. Windows 2000 obtains the system key, either from the locally stored key, the password entry, or insertion of a floppy disk, depending on the option you chose.
2. Windows 2000 uses the system key to decrypt the master protection key.
3. Windows 2000 uses the master protection key to derive the per-user account password encryption key that is then used to decrypt the password information in Active Directory or the local SAM registry key.

The **syskey** command can be used again later to change the system key storage option or to change the password.

▶ **To change the system key option or password**

1. Type **syskey** at a command prompt to bring up the initial system key dialog box, as shown in Figure 15.15.
2. Click **Update**.
3. In the **Account Database Key** dialog box (Figure 15.16), select a key option or change the password, and then click **OK**.
4. Restart the computer.

Changing the system key requires knowledge of, or possession of, the current system key. If the password-derived system key option is used, **syskey** does not enforce a minimum password length; however, passwords longer than 12 characters are recommended. The maximum length is 128 characters.

Warning If the system key password is forgotten or the floppy disk that contains the system key is lost, it might not be possible to start the system. Protect and store the system key safely. If it is on a floppy disk, make backup copies and store them in a different location. The only way to recover the system if the system key is lost is by using a repair disk to restore the registry to a state prior to enabling system key protection. This means that you would lose any information or changes which have accrued since then.

System key options can be configured independently on all computers in a domain. When configured for the system key, each computer has a unique password encryption key and a unique system key. For example, the first domain controller might be configured to use a computer-generated system key stored on a disk, and secondary domain controllers might each use a different computer-generated system key stored on the local system. A computer-generated system key stored locally on a primary domain controller is not replicated.

Before enabling the system key when you have a single domain controller, you might want to ensure that a second, complete, updated domain controller is available as a backup system until changes to the first domain controller are complete and verified. Before you change the system key options on a computer, it is recommended that you make a fresh copy of the emergency repair disk for that computer. For more information about making an emergency repair disk, see Microsoft Windows 2000 Server Help or Microsoft Windows 2000 Professional Help.

Printing EFS Files

An EFS file is as transparent to a printer or other output device as it is to the monitor. If you can read an encrypted file on screen, it prints in plaintext. If you cannot read it on screen, you cannot print it.

This transparency requires that the same physical controls imposed on the computer also be imposed on the printer. The printer itself and the cabling to it must be secured so that an attacker cannot tap into them. If you print with a print server, the server must also be secured.

During printing, Windows 2000 copies the print job onto a spool (.spl) file that resides on the local print provider. In local printing, the local print provider on the local computer is used. In client/server printing, this is bypassed, and the .spl file resides on the local print provider of the server.

By default, .spl files are stored in the *SystemRoot*\System32\Spool\Printers folder. If that folder is unencrypted (as it generally is), the encryption that was in the original file is lost. You can avoid this by encrypting the folder, but this would slow processing by causing every .spl file to be encrypted. A better way is to create a special printer for encrypted files. This printer might use the same print hardware device, but with different print instructions. It should be local and unshared, and it should bypass the default folder by using one of the two following techniques:

- Select the **Print directly to the printer** check box on the **Advanced** page of the printer's **Properties** dialog box. The print job is not spooled, and no .spl file is created.

  Note Unspooled print jobs cannot be scheduled or prioritized.

- Create an encrypted folder and specify that .spl files are to be routed to it. The procedure is described in "Network Printing" in the *Microsoft® Windows® 2000 Server Resource Kit Server Operations Guide.*

By default, when the print job is complete, the .spl file is deleted. You can override the default by selecting the **Keep printed documents** check box on the **Advanced** page. If you select this option, you can resubmit a document to the printer from the printer queue instead of from the program. This is not recommended because the security risk does not outweigh the benefit. Even though the .spl files are encrypted, it is not a good practice to leave multiple copies of sensitive data in different folders.

Troubleshooting EFS

Typical EFS problem situations and approaches to solving them are discussed in this section.

When I try to encrypt my files, it doesn't work.

Make sure that the following conditions are true:

- A recovery agent policy has been defined.
- The file volume is NTFS.
- The file is not compressed.
- You have write access to the file.

Sometimes users think that the file is not encrypted because they can open it and read the file. Remind them to verify that the file is encrypted by checking the file's attribute.

Sometimes a user tries to encrypt a folder that has the compression attribute set or is on a compressed drive. First, you have to remove the compression attribute, and then you can encrypt.

I can't open files I have encrypted.

Make sure you have the correct EFS certificate and private key for the file. If it is an old file, the public key and private key set might no longer be available. Expired certificates and private keys are archived. However, users can delete archived certificates and private keys, or they might be damaged. If so, recover the file as described earlier in this chapter.

If the computer previously operated in stand-alone mode and is now a member of a domain, this can make a difference. The file might have been encrypted by using a local self-signed certificate issued by the computer, whereas the CA designated at the domain level is now the issuing authority.

Are there warnings to a user that a file goes from an encrypted state to an unencrypted state when copying or moving?

There is no warning. Always check the properties of the resulting file to ensure that it is still encrypted.

I can't open an EFS file after upgrading from a previous build of Windows 2000. A message that read "Access denied" appeared, but I can still encrypt and open new EFS files.

It is possible that the previous build is a domestic, nonexportable build with support for strong cryptography, and the new build is an international, exportable build with weaker cryptography. The weaker cryptography technology cannot handle files that have been encrypted using the stronger cryptography.

If you qualify to use and deploy nonexportable cryptography, you can obtain the Encryption Pack CD from Microsoft and use it to convert Windows 2000 to support nonexportable, strong cryptography technology. This CD is not exportable. The Microsoft Enhanced Cryptographic Provider for Windows 2000 is available on this CD. Instructions on how to use the CD are provided with the CD.

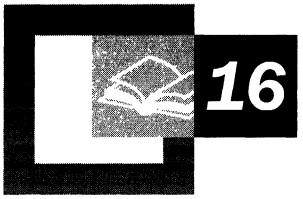
For more information about the availability of the Encryption Pack CD and current cryptography export policies for Microsoft products, see the Microsoft Security Advisor Web site at http://www.microsoft.com/security.

When my virus check program runs, it cannot check all the files on my hard disk and I get "Access Denied" error messages.

Your virus check program can only read files that have been encrypted by you. If other users have encrypted files on your hard disk, access to these files is denied to the virus check program. To perform a virus check for files that have been encrypted by other users, the other users must log on and run the virus check program.

CHAPTER 16

Windows 2000 Certificate Services and Public Key Infrastructure



Microsoft® Windows® 2000 includes both Certificate Services, which is easily administered through the Certification Authority console, a snap-in for Microsoft Management Console (MMC), and a comprehensive public key infrastructure. Understanding the components of Windows 2000 Certificate Services and the public key infrastructure is the starting point for designing, deploying, and maintaining a public key infrastructure that meets all of your public-key security needs.

In This Chapter

Related Information in the Resource Kit

- For more information about the concepts of public key infrastructure and public key technology, see "Cryptography for Network and Information Security" in this book.
- For more information about security solutions that use public key technology, see "Choosing Security Solutions That Use Public Key Technology" in this book.
- For more information about planning and deploying your public key infrastructure, see "Planning Your Public Key Infrastructure" in the *Microsoft® Windows® 2000 Server Resource Kit Deployment Planning Guide*.

Benefits of the Public Key Infrastructure

The Windows 2000 public key infrastructure (also known as a PKI) provides the framework of services, technology, protocols, and standards that enable you to deploy and manage a strong information security system that is based on public key technology. You can deploy your public key infrastructure to support a wide range of network and information security needs.

The Windows 2000 public key infrastructure includes Certificate Services for issuing and managing digital certificates and Microsoft CryptoAPI version 2 for secure cryptographic operations and private key management. The public key infrastructure is fully integrated with the Active Directory™ directory service in Windows 2000, and with distributed security services.

The discussion in this chapter focuses on the individual components and features of the Windows 2000 public key infrastructure. For more information about public key infrastructure and public key technology, see "Cryptography for Network and Information Security" in this book.

Strong Security with Public Key Technology

The Windows 2000 public key infrastructure enables you to deploy strong security solutions that use digital certificates and public key technology. Security solutions can include the following:

- Secure mail, which uses certificates and the Secure/Multipurpose Internet Mail Extensions (S/MIME) protocol to ensure the integrity, origin, and confidentiality of e-mail messages.
- Secure Web sites, which use certificates and certificate mapping to map certificates to network user accounts for controlling user rights and permissions for Web resources.
- Secure Web communications, which use certificates and the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols to authenticate servers, to optionally authenticate clients, and to provide confidential communications between servers and clients.
- Software code signing, which uses certificates and digital signing technology (such as Microsoft® Authenticode®) to ensure the integrity and authorship of software that is developed for distribution on an intranet or on the Internet.
- Smart card logon process, which uses certificates and private keys stored on smart cards to authenticate local and remote access network users.

- Internet Protocol security (IPSec) client authentication, which has the option to use certificates to authenticate clients for IPSec communications.
- Encrypting File System (EFS), which uses certificates for both EFS user and EFS recovery agent operations.
- Custom security solutions, which use certificates to provide confidentiality, integrity, authentication, or nonrepudiation.

For more information about security solutions that use public key technology, see "Choosing Security Solutions That Use Public Key Technology" in this book.

Integration with Active Directory and Distributed Security Services

Windows 2000 Certificate Services form the core of the Windows 2000 public key infrastructure. Enterprise certificate services are integrated with Active Directory and distributed security services, as shown in Figure 16.1.

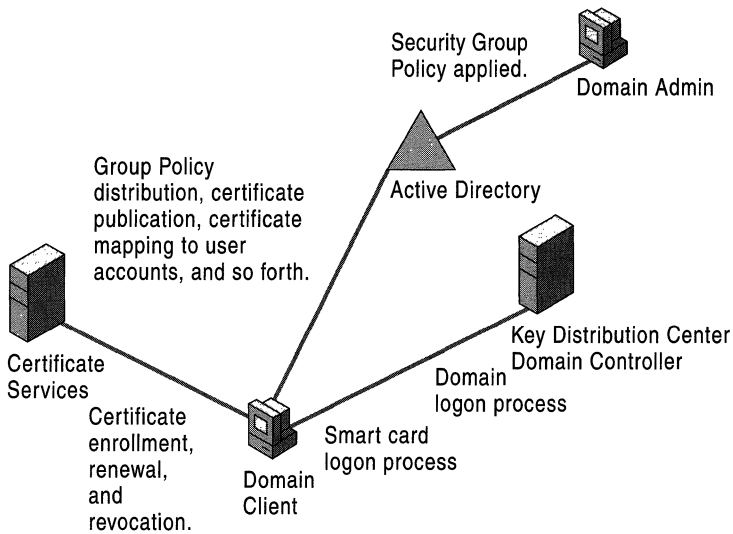


Figure 16.1 Certificate Services in Windows 2000

You can install Windows 2000 Certificate Services to create enterprise certification authorities (CAs) for issuing and managing digital certificates. Active Directory contains information that enterprise CAs require, such as user account names, security group memberships, and certificate templates. Active Directory also contains information about each enterprise CA that is installed in the domain. Certificate requests are usually sent to enterprise CAs that process the requests to either deny or approve them. Issued certificates are distributed to Active Directory and to the requestor's computers. CAs also publish certificate revocation lists to Active Directory.

In addition, Active Directory stores Public Key Group Policy for distribution to all computers that are within the scope of the policy. Public Key Group Policy enables you to control which CAs are to be trusted in the enterprise, to specify alternative EFS recovery agents, and to configure automatic enrollment and renewal of certificates for Windows 2000based computers—all from a central administration point.

Active Directory also supports mapping certificates to network user accounts for authenticating clients and controlling access to network resources. Using smart cards for the user logon process is a special case of certificate mapping that extends the Kerberos v5 authentication protocol to include authentication of users on the basis of certificates and private keys that are stored on smart cards. Using smart cards for the user logon process provides enhanced security for user authentication and a single set of user credentials for logging on locally or remotely over a network.

Major Components of the Public Key Infrastructure

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

- Windows 2000 Certificate Services for issuing and managing digital certificates.
- Microsoft CryptoAPI and cryptographic service providers (CSPs) for providing cryptographic operations and private key management.
- Certificate stores for storing and managing certificates in the enterprise.

Windows 2000 Certificate Services

Figure 16.2 shows a functional block diagram of Windows 2000 Certificate Services.

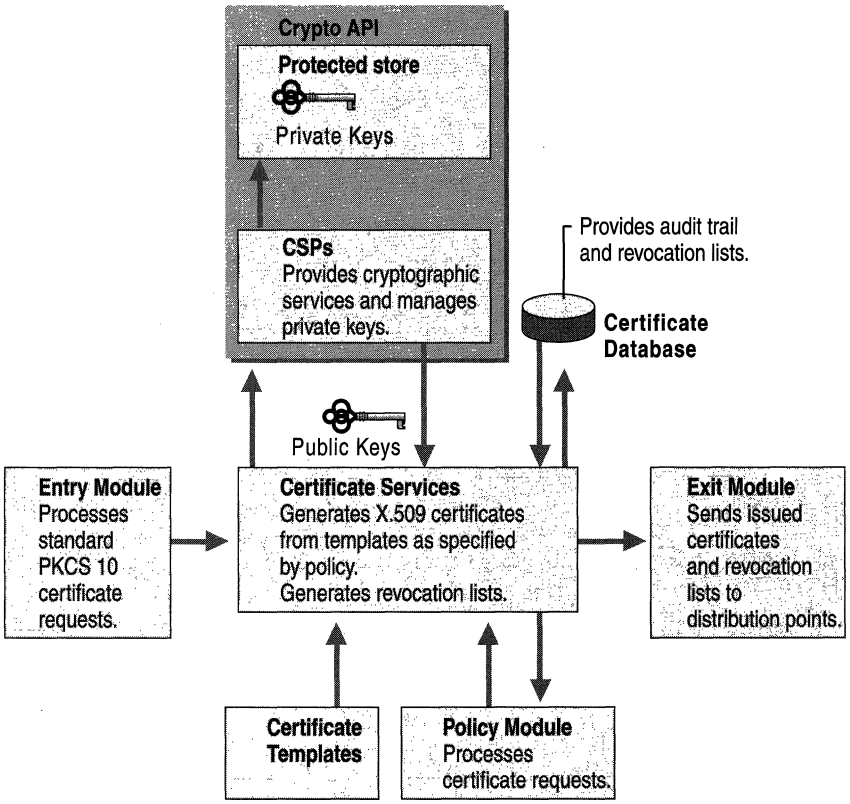


Figure 16.2 Certificate Services Functional Diagram

The components of Windows 2000 Certificate Services work in conjunction with Microsoft CryptoAPI and cryptographic service providers (CSPs) to perform a variety of tasks, including the following:

- Process certificate requests (entry module).
- Verify whether requestors are qualified to receive certificates (policy module).
- Create and issue certificates for qualified requestors (Certificate Services engine).
- Generate a private key and distribute it to the requestor's protected store (CSPs and Microsoft CryptoAPI).
- Manage the private key for all cryptography operations (CSPs and Microsoft CryptoAPI).
- Distribute the certificates that are issued to qualified requestors and, optionally, publish certificates to Web pages, public folders, or Active Directory (exit module).
- Publish periodic certificate revocation lists to Active Directory and, optionally, to Web pages or public folders (exit module).
- Store all certificate transactions for the audit trail (certificate database).

Note In Windows 2000, all cryptographic functions and private key management are performed by Microsoft CryptoAPI in conjunction with CSPs. Any system service or application can request cryptographic services by using Microsoft CryptoAPI.

Entry Module

The default entry module processes standard PKCS (Public Key Cryptography Standards) 10 certificate requests made through remote procedure calls (RPCs) or the Hypertext Transfer Protocol (HTTP). The entry module is a dynamic-link library (DLL) that cannot be customized. Windows 2000 services usually use RPCs to submit certificate requests to enterprise CAs. However, the Web Enrollment Support pages use Hypertext Transfer Protocol (HTTP) to submit certificate requests to CAs.

Certificate requests to Certificate Services are placed in a pending queue until they are approved or denied by the policy module.

Note You can develop custom certificate enrollment applications that submit RPC or HTTP requests to Certificate Services. For more information about developing custom applications for Windows 2000 Certificate Services and about the required certificate request format, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Policy Modules

The policy module determines whether a certificate request must be approved, denied, or queued (left pending) for a later decision by the administrator about whether or not to issue the certificate. Windows 2000 Certificate Services includes a default policy module that incorporates CA policy for both enterprise and stand-alone CAs. You can also build custom policy modules for special needs.

Enterprise CA Policy Enterprise CA policy always issues a certificate or denies a request immediately. Enterprise CA policy uses Active Directory to determine the identity of the requester, and then automatically determines whether the requester has security permissions to receive a certificate of the type that is being requested.

Stand-alone CA Policy By default, stand-alone CA policy sends certificate requests to a pending queue so that an administrator can approve or deny them. You have the option of setting stand-alone CA policy to automatically approve all certificate requests. However, because a stand-alone CA does not verify the identity of requesters who are using Active Directory, there is no way to verify the identity and validity of the certificate requester automatically. Therefore, setting a stand-alone CA to approve certificate requests automatically can pose a significant security risk.

Custom Policy Modules The policy module is a fully customizable DLL. For more information about how to customize policy modules, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. You can change the installed policy module by using the Certification Authority console. You can also develop your own policy modules or acquire a third-party policy module when one is available.

Note It is recommended that you use custom policy modules with stand-alone CAs only. Enterprise CAs require the enterprise policy module to ensure proper integration with Active Directory. Using a custom policy module with an enterprise CA can produce both problems and unpredictable results.

When Certificate Services determines whether to grant certificate requests, the policy module can check information in the request against various sources for verification, such as a directory service, an external legacy database, or credit information from an outside authority. The policy module also can send alerts to the appropriate administrator if manual (offline) approval of the request is required.

The policy module can insert additional certificate attributes or extensions that might be required by a client application. For example, information such as a job title and a signing limit into certificates can be inserted and used by an online purchasing form to determine whether the user can sign for the amount requested.

The policy module can use additional information included in the certificate request to incorporate requested attributes in the issued certificate. For example, certificate requests to stand-alone CAs must include all information about the requested certificate; so the policy module incorporates this information into each certificate that is issued. However, enterprise CAs use certificate templates to specify certificate attributes; so certificate requests to enterprise CAs require less information.

Certificate Templates

For enterprise CAs, certificate templates define the attributes for certificate types. You can configure enterprise CAs to issue specific certificate types to authorized users and computers. When a CA creates a certificate, the certificate template is used to specify its attributes, such as the authorized uses for the certificate, the cryptographic algorithms that are to be used with it, the public key length, and the certificate lifetime. Certificate templates are stored in Active Directory and provide information for each of the certificate types that are listed in Table 16.1.

Table 16.1 Certificate Types for Enterprise CAs

Certificate Type	Purpose of the Issued Certificate
Administrator	Used for authenticating clients and for EFS, secure mail, certificate trust list (CTL) signing, and code signing.
Authenticated Session	Used for authenticating clients.
Basic EFS	Used for EFS operations.
CEP Encryption (offline request)	Used to enroll Cisco Systems, Inc. routers for IPSec authentication certificates from a Windows 2000 CA.
Code Signing	Used for code signing operations.
Computer	Used for authenticating clients and servers.

(continued)

Table 16.1 Certificate Types for Enterprise CAs *(continued)*

Certificate Type	Purpose of the Issued Certificate
Domain Controller	Used for authenticating domain controllers. When an enterprise CA is installed, this certificate type is installed automatically on domain controllers to support the public key operations that are required when domain controllers are supporting Certificate Services.
EFS Recovery Agent	Used for EFS encrypted-data recovery operations.
Enrollment Agent	Used for authenticating administrators that request certificates on behalf of smart card users.
Enrollment Agent (computer)	Used for authenticating services that request certificates on behalf of other computers.
Exchange Enrollment Agent (offline request)	Used for authenticating Microsoft® Exchange Server administrators that request certificates on behalf of secure mail users.
Exchange Signature Only (offline request)	Used by Exchange Server for client authentication and secure mail (used for signing only).
Exchange User (offline request)	Used by Exchange Server for client authentication and secure mail (used for both signing and confidentiality of mail).
IPSec	Used for IPSec authentication.
IPSec (offline request)	Used for IPSec authentication.
Root Certification Authority	Used for root CA installation operations. (This certificate template cannot be issued from a CA and is used only when installing root CAs.)
Router (offline request)	Used for authentication of routers.
Smart Card Logon	Used for client authentication and logging on with a smart card.
Smart Card User	Used for client authentication, secure mail, and logging on with a smart card.
Subordinate Certification Authority (offline request)	Used to issue certificates for subordinate CAs.
Trust List Signing	Used to sign CTLs.
User	Used for client authentication, EFS, and secure mail (used for both signing and confidentiality of mail).
User Signature Only	Used for client authentication and secure mail (used for signing only).
Web Server (offline request)	Used for Web server authentication.

Many certificate templates are provided for online requests from enterprise CAs. Online certificate templates are used to issue certificates to requestors that have Windows 2000 accounts and that support obtaining certificates directly from an enterprise CA. Certificate templates for offline requests are used to issue certificates to requestors that do not have Windows 2000 accounts or that do not support obtaining certificates directly from an enterprise CA. When a certificate is issued for online requests, identification information about the requestor is obtained from the requestor's Windows 2000 user account for inclusion in the certificates that are issued. Offline requests must include the requestor's identification information in the certificate request when the request is submitted. When you use the Certificate Services Web Enrollment Support pages to request offline certificates from an enterprise CA, enter the identification information (name, e-mail address, department, and so forth), in the Web form before you submit the request to the CA.

For example, you might use the Web Enrollment Support pages to obtain a Web Server certificate for a third-party Web server, and then install the certificate on the appropriate server computer. Likewise, you might obtain an offline IPSec certificate, and then manually install the certificate on a non-Windows 2000 IPSec client. The Subordinate Certification Authority certificate template is an offline template because the identification information for the subordinate CA is entered during the installation process.

An enterprise CA only issues the certificate types that are specified by its certificate issuing policy. By default, Windows 2000 enterprise CAs are installed so that they are ready to issue several types of certificates. You can modify the default configuration by using the Certification Authority console in MMC to specify the types of certificates that are to be issued by each CA.

Stand-alone CAs do not use certificate templates. Therefore, certificate requests to them must include all of the information that is necessary to define the type of certificate that is to be issued. When Windows 2000 services submit certificate requests to stand-alone CAs, the requests include the information that is necessary to define the type of certificate that is being requested. You can use the Web Enrollment Support pages for stand-alone CAs to submit certificate requests to stand-alone CAs for a variety of types of certificates.

Certificate Database

The certificate database records all certificate transactions. It tracks all certificate requests and records whether they were granted or denied. It records information for the issued certificate, such as the serial number and expiration date. It provides a complete audit trail for each certificate from request to expiration. It also flags and tracks certificates that are revoked by CA administrators. You can use the Certification Authority console to manage the audit trail.

Because the certificate database is a transaction database, it includes certificate log files, which record all certificate transactions. By default, the certificate database and the certificate log files are installed at the following location:

```
<Drive:>\WINNT\System32\CertLog
```

where *<Drive:>* is the letter of the disk drive where the CA is installed.

At the time you install the CA, you have the option of choosing another location to install either the database or the logs, including storing the database and log files separately on different drives.

Exit Modules

The exit module packages the issued certificate in the appropriate transport mechanism or protocol and distributes it to the location specified in the request. Certificate requests can specify that the certificate be distributed to Lightweight Directory Access Protocol (LDAP) directory services, file systems, or URLs. An exit module also delivers certificate revocation lists (CRLs) to CRL distribution points.

The default enterprise exit module publishes certificates and CRLs to Active Directory, and the default stand-alone exit module publishes certificates and CRLs to the local file system. However, Windows 2000 Certificate Services supports multiple exit modules and you can use the Certification Authority console to install them for a CA. For example, you can install exit modules that send certificates and CRLs in e-mail messages or send them to public folders on the network. You can also install exit modules that post certificates to legacy open database connectivity (ODBC) databases or to third-party Lightweight Directory Access Protocol (LDAP) directory services.

Like the policy module, the exit module is a DLL and is fully customizable. For more information about customizing exit modules, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. You can change the installed exit module by using the Certification Authority console. You also can develop your own exit modules or acquire third-party exit modules.

Certification Authority Console

The Certification Authority console is an MMC snap-in that you can use to manage multiple CAs, performing a variety of administrative tasks that include the following:

- Starting and stopping the CA.
- Backing up and restoring the CA.
- Changing exit and policy modules.
- Viewing the CA certificate.
- Installing or reinstalling a CA certificate for the CA.
- Setting security permissions and delegating administrative control for the CA.
- Revoking certificates.
- Viewing or modifying certificate revocation list (CRL) distribution points.
- Scheduling and publishing CRLs.
- Configuring the types of certificates that are to be issued by the CA.
- Viewing information about certificates that have been issued.
- Viewing information about certificates that have been revoked.
- Viewing pending certificate requests.
- Approving or denying pending certificate requests.
- Viewing failed certificate requests.
- Renewing the CA's certificate.

For more information about how to use the Certification Authority console to manage a CA and perform specific administration tasks, see Certificate Services Help.

▶ **To add a Certification Authority console to MMC**

1. Open MMC.
2. Click **Console**, and then click **Add/Remove Snap-in** or press CTRL+M.

   The **Add/Remove Snap-in** dialog box appears.
3. Click **Add**.

   The **Add Standalone Snap-in** dialog box appears.
4. Select **Certification Authority** from the list of snap-ins, and then click **Add**.

   The **Certification Authority** dialog box appears.

5. Choose one of the following:
   - To manage the CA that is running on the local computer, select the **Local computer** check box, and then click **Finish**.
   - To manage the CA that is running on another computer, select the **Another computer** check box, and then type the domain name of the computer that runs the CA or click **Browse** to select the computer from a list. Then click **Finish**.

   You can click **Add** in the **Add Standalone Snap-in** dialog box again to add more Certification Authority consoles.

   The **Add/Remove Snap-in** dialog box displays the snap-ins that you have added and that are to be installed in MMC.
6. When you have finished adding snap-ins, in the **Add Standalone Snap-in** dialog box, click **Close.**
7. In the **Add/Remove Snap-in** dialog box, click **Close**.

Figure 16.3 shows an example of a Certification Authority console that has been added to MMC. This console manages the CA on the local computer.

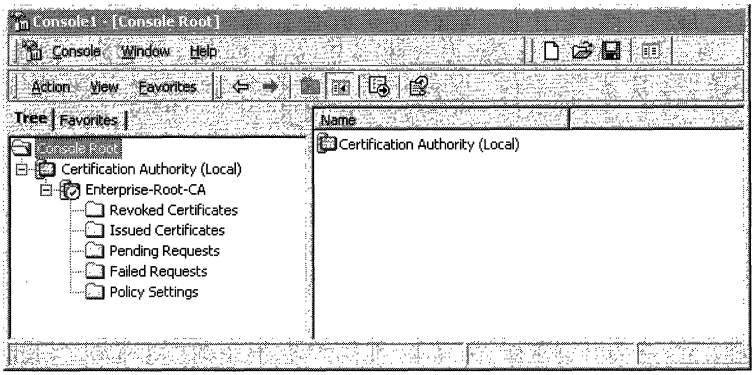


Figure 16.3 Certification Authority Console

The Certification Authority (Local) console node has been expanded to show all of the containers for an enterprise CA named Enterprise-Root-CA. These containers are used as follows:

Revoked Certificates Click this container to show information about all revoked certificates for this CA. To manually publish CRLs, right-click the Local node. Click **All Tasks**, and then click **Publish**. To change the CRL publication schedule, right-click the Local node, and then click **Properties**. To view a certificate, double-click the certificate. Use the dialog boxes that appear to publish the CRL, change the CRL publication schedule, or view the certificate.

Issued Certificates Click this container to show information about all certificates that have been issued by this CA. To revoke a certificate, right-click the certificate, and then click **All Tasks**. Then click **Revoke Certificate**. To view a certificate, double-click the certificate. Use the dialog boxes that appear to revoke or view certificates.

Pending Requests Click this container to show information about all certificates that are pending for this CA. To approve a pending certificate request, click this container, and then right-click the certificate request. Click **All Tasks**, and then click **Issue**. To deny a pending certificate request, click this container and then right-click the certificate request. Click **All Tasks**, and then click **Deny**. Use the dialog box that appears to deny the certificate request.

Failed Requests Click this container to show the information about all certificate requests that have failed. The information in the Request Disposition Message column explains why the request failed.

Policy Settings (Enterprise CAs Only) Select this container to show the types of certificates that the enterprise CA can issue. To remove one of the types of certificates, select the type you want to delete, and then press DELETE. To add another type of certificate, right-click the container. Click **New**, and then click **Certificate to Issue**. Use the dialog box that appears to add the types of certificates that you want to issue.

When you click a container (such as the Failed Requests container), by default, many of the columns that can be displayed in the details pane of the console are hidden.

▶ **To change the columns that are displayed in the details pane for a container**

1. Right-click the container, click **View,** and then click **Choose Columns**.

   The **Modify Columns** dialog box appears.

2. Use the **Modify Columns** dialog box to add, remove, or change the order in which columns appear, and then click **OK**.

   For more information about how to use the **Modify Columns** dialog box, see Certificate Services Help.

Microsoft CryptoAPI and Cryptographic Service Providers

Microsoft CryptoAPI provides a secure interface for the cryptographic functionality that is supplied by the installable cryptographic service provider (CSP) modules. CSPs perform all cryptographic operations and manage private keys. CSPs can be implemented in software as well as in hardware. Windows 2000 Certificate Services uses CryptoAPI and CSPs to perform all cryptographic and private key management operations. CryptoAPI and CSP services are available to all services and applications that require cryptographic services. For more information about CryptoAPI and CSPs, see the Microsoft Security Advisor link and the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Hardware and Software Cryptographic Service Providers

CSPs can be software-based, hardware-based, or a combination of both. Hardware-based cryptography and key management is more secure than software-based cryptography and key management because cryptographic operations and private keys are isolated from the operating system. However, hardware-based CSPs (such as smart card CSPs) often store only a limited number of private keys and can take a long time to generate keys.

Software CSPs usually provide more flexibility than hardware CSPs, but at the cost of somewhat less security. Nevertheless, software-based CSPs can provide ample security to meet a wide range of needs. You usually use hardware-based CSPs only for special security applications, such as for logging on with smart cards or for secure Web communications with FORTEZZA Crypto Cards.

Vendors can develop hardware or software CSPs that support a wide range of cryptographic operations and technologies. However, Microsoft must certify and digitally sign all CSPs. CSPs do not work in Windows 2000 unless they have been digitally signed by Microsoft.

Microsoft Cryptographic Service Providers

Windows 2000 includes the following Microsoft CSPs.

Microsoft Base Cryptographic Provider Provides a broad set of basic cryptographic functionality. It is not subject to United States government cryptography export restrictions and can be exported to other countries (subject to general United States export restrictions, as well as the import restrictions of other countries). The Base CSP uses RSA technology, which is licensed from RSA Data Security, Inc.

Microsoft Enhanced Cryptographic Provider Provides the same capabilities as the Microsoft Base Cryptographic Provider, but in addition, provides stronger security by supporting longer key lengths and additional cryptographic algorithms. This CSP is subject to government-imposed cryptography export restrictions and might not be available in your locality. The enhanced CSP also uses RSA technology.

Microsoft DSS Cryptographic Provider Provides data signing and signature verification capability by using the Secure Hash Algorithm (SHA) and Digital Signature Algorithm (DSA). It is not subject to United States government cryptography export restrictions and can be exported to other countries (subject to general United States export restrictions, as well as the import restrictions of other countries).

Microsoft Base DSS and Diffie-Hellman Cryptographic Provider Provides a superset of the DSS Cryptographic Provider and also supports Diffie-Hellman key exchange, hashing (message digests), data signing, and signature verification by using the SHA and DSA algorithms. This CSP is subject to government-imposed export restrictions on cryptography and might not be available in your locality.

Schannel Cryptographic Providers The Microsoft RSA/Schannel Cryptographic Provider, the Microsoft DSS Cryptographic Provider, and the Diffie-Hellman/Schannel Cryptographic Provider offer various cryptographic services that are required for data integrity, session key exchange, and authentication during secure Web communications with the SSL and TLS protocols. These CSPs are not subject to United States government cryptography export restrictions and can be exported to other countries (subject to general United States export restrictions, as well as the import restrictions of other countries).

FIPS 140-1 Level 1 Certification

The Windows 2000 Microsoft CSPs have received the Federal Information Processing Standard (FIPS) 140-1 Level 1 certification by the National Institute of Standards and Technology (NIST). The requirements for FIPS 140-1 Level 1 certification are contained in the FIPS 140-1 publication, which is published by NIST. For more information about how to obtain the FIPS140-1 publication, contact NIST. For more information about FIPS 140-1, see "Choosing Security Solutions That Use Public Key Technology" in this book.

Base vs. Enhanced Cryptographic Service Providers

The Microsoft Base Cryptographic Provider (Base CSP) is provided for export in compliance with United States government export restrictions on cryptography. The Microsoft Enhanced Cryptographic Provider (Enhanced CSP), however, is subject to United States government export restrictions on cryptography and is available only for localities where the export of strong cryptography is permitted. For more information about restrictions on cryptography, see "Cryptography for Network and Information Security" in this book, and see the Microsoft Security Advisor link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Table 16.2 highlights differences between the Base CSP and the Enhanced CSP. The public key lengths shown in the table are the default key lengths.

Table 16.2 Comparison of Microsoft Base CSP and Microsoft Enhanced CSP

Algorithm	Base CSP	Enhanced CSP
RSA public key signature algorithm	Key length: 512 bits.	Key length: 1,024 bits.
RSA public key exchange algorithm	Key length: 512 bits.	Key length: 1,024 bits.
RC2 block encryption algorithm	Key length: 40 bits.	Key length: 128 bits. Salt length: Settable.
RC4 stream encryption algorithm	Key length: 40 bits.	Key length: 128 bits. Salt length: Settable.
DES	Not supported.	Key length: 56 bits.
Triple DES (2-key)	Not supported.	Key length: 112 bits.
Triple DES (3-key)	Not supported.	Key length: 168 bits.

For both the Base CSP and the Enhanced CSP, public keys that are used for digital signatures can be up to 16,384 bits long. However, public keys that are used for key encryption and key exchange (to protect secret keys) are limited to a maximum of 1,024 bits for the Base CSP and 16,384 bits for the Enhanced CSP. In addition, the symmetric keys for the encryption algorithms in the Base CSP are limited to shorter key lengths, resulting in significantly weaker cryptographic security. Overall, the key lengths and the encryption algorithms in the Enhanced CSP provide far stronger cryptographic security.

For both the Base CSP and the Enhanced CSP, public keys used for signing or key exchange can be a minimum of 384 bits long. However, the use of 384-bit public keys is not recommended. The minimum recommended length of public keys is 512 bits; however, public keys of at least 1,024 bits are recommended whenever this is feasible. Signing keys that exceed 1,024 bits in length can produce strong digital signatures. However, because they also can increase the computational load significantly and require large amounts of time to sign data, they also can adversely affect computer performance and, thus, might not be feasible. The default public-key length of the Base CSP is 512 bits, and the default public key length of the Enhanced CSP is 1,024 bits. Windows 2000 Certificate Services usually uses the default public-key lengths of the CSP, unless you choose another key length that is supported by the CSP in advanced options.

The Enhanced CSP is compatible with the Base CSP, except that the CSPs can generate only RC2 or RC4 keys of the default key length. The default symmetric key length for RC2 and RC4 in the Base CSP is 40 bits. The default symmetric length for RC2 and RC4 in the Enhanced CSP is 128 bits. Therefore, the Enhanced CSP cannot create keys with Base CSP–compatible key lengths. However, the Enhanced CSP can import RC2 and RC4 keys of up to 128 bits. Therefore, the Enhanced CSP can import and use 40-bit keys that were generated by using the Base CSP.

Smart Card Cryptographic Service Providers

Windows 2000 includes smart card CSPs from two vendors: Gemplus SCA and Schlumberger Limited. The Gemplus GemSAFE Card CSP and the Schlumberger CSP support cryptographic operations for the Gemplus and Schlumberger PC/SC-compliant smart cards, respectively. Additional smart card CSPs might be developed and certified for use with Windows 2000. For current information about smart card CSPs that are currently available, see the Microsoft Security Advisor link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Cryptography Export Restrictions

CSPs are subject to cryptography export restrictions. Some governments, including the United States government, currently place export restrictions on encryption technology. Other governments also place import restrictions on encryption technology. The availability of CSPs varies according to the export or import restrictions for a specific geographical area.

All Windows 2000 products support a maximum of 40-bit or 56-bit symmetric key encryption and are exportable to most localities worldwide. If you qualify to use and deploy nonexportable cryptography, you can obtain the Encryption Pack compact disc (CD) from Microsoft and use it to convert exportable Windows 2000 products into nonexportable, strong cryptography products. The Microsoft Enhanced Cryptographic Provider for Windows 2000 is available on this CD, which is not exportable.

For more information about the availability of the Encryption Pack CD and current cryptography export policies for Microsoft products, see the Microsoft Security Advisor link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Certificate Stores

In Windows 2000, public-key objects such as certificates, CRLs, and CTLs are stored in certificate stores for use by users, services, and computers. The Windows 2000 certificate stores include physical stores and logical stores.

The *physical certificate stores* are where public-key objects such as certificates, CRLs, and CTLs are physically stored either locally in the system registry of the computer or remotely in Active Directory. Many of the public-key objects in the physical stores are shared among users, services, and computers through the use of logical certificate stores.

Logical certificate stores group certificates together in logical, functional categories for users, computers, and services. Logical certificate stores contain pointers to the physical certificate stores. Use the Certificates console (an MMC snap-in) to manage certificates in certificate stores. Changes to the logical certificate stores are made to the appropriate physical stores that are located in either the system registry or Active Directory. Because you use only the logical certificate store for a user, service, or computer, you neither have to keep track of where the certificates are actually stored, nor do you have to edit the system registry to manage the certificate stores.

The use of logical certificate stores eliminates the necessity of storing duplicates of common public key objects, such as trusted root certificates, CTLs, and CRLs for users, computers, and services. Users and services share many public key policy objects in common with the local computer. The common public-key objects are stored in sections of the registry of the local computer. However, some certificates, CTLs, or CRLs, are issued for use only by an individual service, user, or local computer. Therefore, users, computers, and services also have individual stores that provide a place to store certificates, CTLs, or CRLs that are not shared in common. For example, a user can request and obtain a certificate or a CRL, which appears in the individual's logical store and is physically stored in the user's unique certificate store in the registry. Such individual user certificates and CRLs are not shared with local computers or with services.

In addition, some public-key objects, such as trusted root certificates and CTLs, can be distributed through Public Key Group Policy. Public key objects that are distributed through Group Policy are stored in special areas of the system registry and appear in the logical stores for users, computers, and services. When you use Group Policy, separate CTLs can be created for users and computers. The CTLs for users are not shared with services or the computer. However, the CTLs for computers are shared with users and services.

The logical certificate stores include the following categories for users, computers, and services:

Personal. Contains individual certificates for the user, service, or computer. For example, when an enterprise CA issues you a User certificate, the certificate is installed in the Personal store for your user account.

Trusted Root Certification Authorities. Contains certificates for root CAs. Certificates with a certification path to a root CA certificate are trusted by the computer for all valid purposes of the certificate.

Enterprise Trust. Contains CTLs. Certificates with a certification path to a CTL are trusted by the computer for purposes specified in the CTL.

Intermediate Certification Authorities. Contains certificates for CAs that are not trusted root certificates (for example, certificates of subordinate CAs), but that are required to validate certification paths. This store also contains CRLs for use by the user, service, or computer.

Active Directory User Object. Contains certificates that are published in Active Directory for the user. This store appears in the Certificates console for users only, not for computers or services.

Request. Contains pending or rejected certificate requests. This store appears only in the Certificates console after a certificate request has been made for the user, computer, or service.

SPC. Contains certificates for software publishers that are trusted by the computer. Software that has been digitally signed by publishers with certificates in this store is downloaded without prompting the user. By default, this store is empty. When Microsoft® Internet Explorer downloads software that has been signed by a software publisher for the first time, users are prompted to choose whether they want to trust all software that is signed by this publisher. If a user chooses to trust all software signed by the publisher, the publisher's software publisher certificate (SPC) is added to the SPC store. This store appears in the Certificates console for the local computer only, not for users or services.

Features of the Public Key Infrastructure

The Windows 2000 public key infrastructure and Windows 2000 Certificate Services include the following key features:

- Certificates console (an MMC snap-in)
- Certification authority trust model
- Windows 2000 enterprise (and stand-alone) certificate authorities
- Certificate life cycle
- Certificate enrollment and renewal methods
- Public Key Group Policy
- Certificate revocation lists
- Preinstalled trusted root CA certificates
- Smart card support
- Certificate mapping
- Roaming profile support

Certificates Console

The Certificates console is an MMC snap-in, which you can use to manage the certificate stores for users, computers, and services.

You can use the Certificates console to perform the following tasks:

- View information about certificates, such as certificate contents and the certification path.
- Import certificates into a certificate store.
- Move certificates between certificate stores.
- Export certificates and, optionally, export private keys (if key export is enabled).
- Delete certificates from certificate stores.
- Request certificates from an enterprise CA for the Personal certificate store.

For more information about how to use the Certificates console to do these tasks, see Certificate Manager Help.

▶ To add a Certificates console to MMC

1. Open MMC.
2. Click **Console**, and then click **Add/Remove Snap-in**.

   –Or–

   Press CTRL+M.

   The **Add/Remove Snap-in** dialog box appears.
3. Click **Add**.

   The **Add Standalone Snap-in** dialog box appears.
4. Select **Certificates** from the list of snap-ins, and then click **Add**.

   The **Certificates Snap-in** dialog box appears.
5. Select one of the following accounts:
   - **My user account**
   - **Service account**
   - **Computer account**

   The Certificates console manages the certificate stores for this account.
6. Click **Next**.

   If you selected **My user account**, the **Add Standalone Snap-in** dialog box appears. You can click **Add** to add another snap-in.

   If you selected **Service account** or **Computer account**, the **Select Computer** dialog box appears. To manage the local computer, click **Next**. To manage another computer, either type the domain name of the computer in **Another computer**, or click **Browse** to select the computer from a list. Then click **Next**.

   If you selected **Computer account**, the **Add Standalone Snap-in** dialog box appears. You can click **Add** to add another snap-in.

   If you selected **Service account**, the **Certificates Snap-in** dialog box appears. Select a service from the **Services account** list, and click **Finish**. When the **Add Standalone Snap-in** dialog box appears, you can click **Add** to add another snap-in.
7. When you are finished adding snap-ins, in the **Add Standalone Snap-in** dialog box, click **Close**.

   The **Add/Remove Snap-in** dialog box appears and displays the snap-ins that you are installing in MMC.
8. In the **Add/Remove Snap-in** dialog box, click **Close**.

Figure 16.4 shows an example of three Certificates console nodes that have been added to MMC. The first Certificates console node manages certificates for the logged on user. The second Certificates console node manages certificates for the World Wide Web Publishing service for the local computer. The third Certificates console node manages certificates for the local computer itself.

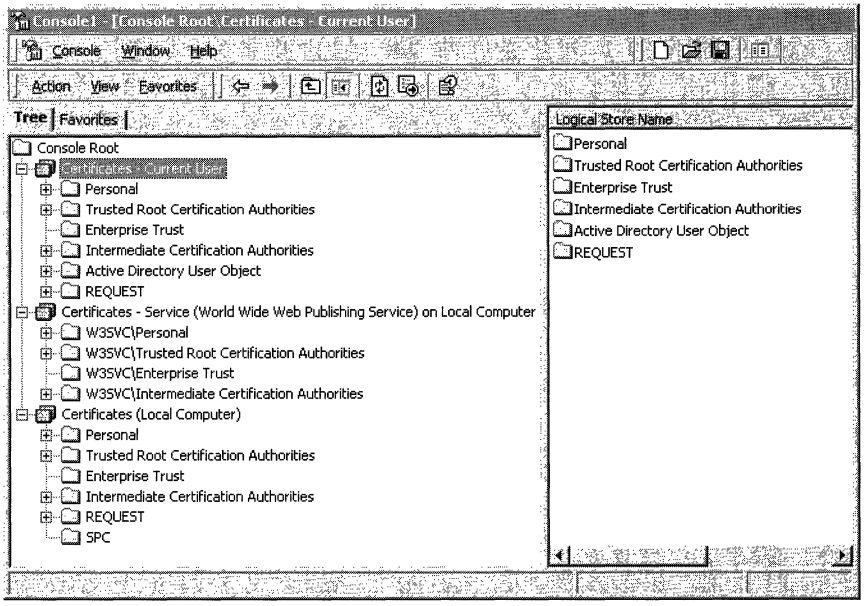


Figure 16.4 Certificates Console

The Certificates console nodes in Figure 16.4 have been expanded to show the logical certificate stores. This is called the Logical display mode. You also have the option of viewing certificates by their physical stores or by their purpose.

To change the display mode, select the Certificates console (such as the Certificates - Current User console). Click **View** and then click **Options**. When the **View Options** dialog box appears, you can choose from the display mode options that are described in Table 16.3.

Table 16.3 View Options Dialog Box

Option	Description
Certificate purpose	Select this option to view certificates in the Purposes display mode, in which certificates are grouped by the intended purpose of the certificates, such as Encrypting File System, File Recovery, and Code Signing.
Logical certificate stores	Select this option to view certificates in the Logical display mode, in which certificates are grouped by the logical store where they are located. This is the default display mode.
Physical certificate stores	Select this option to view the physical stores in addition to the logical stores. This option is available for the Logical display mode only.
Archived certificates	Select this option to view archived certificates. When certificates expire or are renewed, Windows 2000 maintains archives of the certificates and their private keys. Retaining archived certificates is recommended because you might need to use the certificate and its private key later. For example, you might have to verify digital signatures for old documents that were signed with a key for a currently expired or renewed certificate.

Certification Authority Trust Model

The Windows 2000 public key infrastructure supports a hierarchical CA trust model and CTLs. To control what certificates are trusted in the enterprise, you can deploy Windows 2000 Certificate Services to create CA trust hierarchies and you can create CTLs.

Certification Authority Hierarchies

The Windows 2000 public key infrastructure supports a hierarchical CA trust model, called the *certification hierarchy,* to provide scalability, ease of administration, and compatibility with a growing number of commercial third-party CA services and public key-aware products. In its simplest form, a certification hierarchy consists of a single CA. However, the hierarchy usually contains multiple CAs that have clearly defined parent-child relationships. Figure 16.5 shows some possible CA hierarchies.

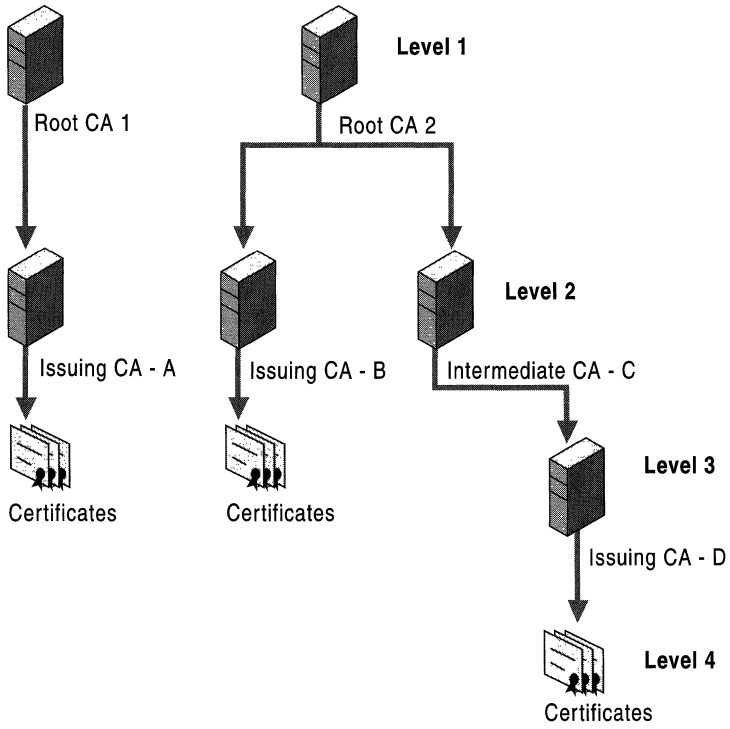


Figure 16.5 Certification Hierarchies

You can deploy multiple CA hierarchies to meet your needs. The CA at the top of the hierarchy is called a *root CA*. Root CAs are self-certified by using a self-signed CA certificate. Root CAs are the most trusted CAs in the organization and it is recommended that they have the highest security of all. There is no requirement that all CAs in an enterprise share a common top-level CA parent or root. Although trust for CAs depends on each domain's CA trust policy, each CA in the hierarchy can be in a different domain.

Child CAs are called *subordinate CAs*. Subordinate CAs are certified by the parent CAs. A parent CA certifies the subordinate CA by issuing and signing the subordinate CA certificate. A subordinate CA can be either an intermediate or an issuing CA. An *intermediate CA* issues certificates only to subordinate CAs. An *issuing CA* issues certificates to users, computers, or services.

There is no restriction with regard to how deep the certification hierarchy can be. However, for many organizations, a three-level certification hierarchy (root CA, intermediate CA, and issuing CA) meets most needs.

Certification Path

A certification hierarchy forms a trust chain, called the *certification path*, from the certificate back to the root CA. Figure 16.6 illustrates a certification path for a four-level path that corresponds to the three-level CA hierarchy in Figure 16.5.

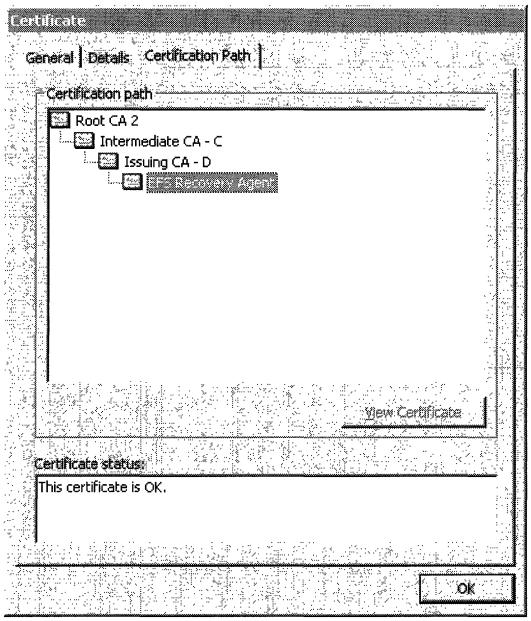


Figure 16.6 Trusted Certification Path

In the example, an EFS Recovery Agent certificate that was issued by Issuing CA - D has a certification path to Root CA 2 at the top of the path. The EFS Recovery Agent certificate is trusted because the certificate for Root CA 2 is contained in the Trusted Root Certification Authorities store.

The certification path links each certificate in the chain back to the root CA. Certificates that have a valid certification path to a root certificate that is in the Trusted Root Certification Authorities store are trusted for all purposes listed in the certificate. If the root CA's certificate for a certification path is not in the Trusted Root Certification Authorities store, the certification path is not trusted until the certificate of the root CA is added to the Trusted Root Certification Authorities store.

Before it trusts a certificate, Microsoft CryptoAPI validates the certification path from the certificate to the certificate of the root CA by checking each certificate in the path. Each certificate contains information about the parent CA that issued the certificate. CryptoAPI retrieves the certificate of each parent CA in the path from either the Intermediate Certification Authorities store or the Trusted Root Certification Authorities stores (if the certificates are present in the stores), or from an online location (such as an HTTP or LDAP address) that is specified in the certificate. If CryptoAPI discovers a problem with one of the certificates in the path, or if it cannot find a certificate, it does not trust the certification path.

When CryptoAPI retrieves a subordinate CA certificate for certificate path validation and the certificate is not located in the Intermediate Certification Authorities store, the API stores the certificate in the Intermediate Certification Authorities store for future reference. However, for computers that operate offline, such as portable computers that are used by mobile users, you might have to import subordinate CA certificates into the Intermediate Certification Authorities store to ensure that nonroot CA certificates are available to validate certification paths.

Figure 16.7 shows an example of a nontrusted certification path where the root certificate is not in the Trusted Root Certification Authorities store.

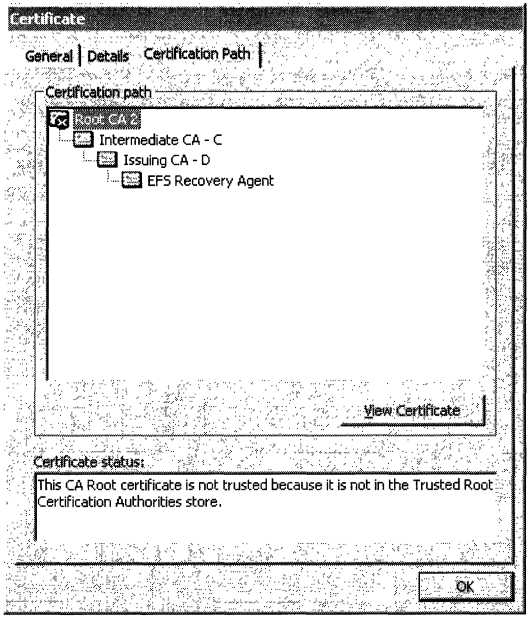


Figure 16.7 Nontrusted Certification Path

By default, certificates that are issued by trusted CAs are trusted for all of the intended purposes that are listed in the certificate. You can use the **Certificate Details** dialog box to restrict the purposes for which local certificates can be used. You can also use CTLs to establish trust for certificates and restrict the purposes for which certificates are trusted.

Certificate Trust Lists

You can use the Certificate Trust List wizard that is available from the Public Key Policy section of the Group Policy console (an MMC snap-in) to create CTLs. By using CTLs, you can choose to trust certificates that have certification paths to root CAs that are listed in the CTL. You can create CTLs for computers and users. CTLs for computers apply to all computers, users, and services within the scope of the Group Policy. However, CTLs for users apply only to users within the scope of the Group Policy. Figure 16.8 shows an example of a certification path with a CTL.

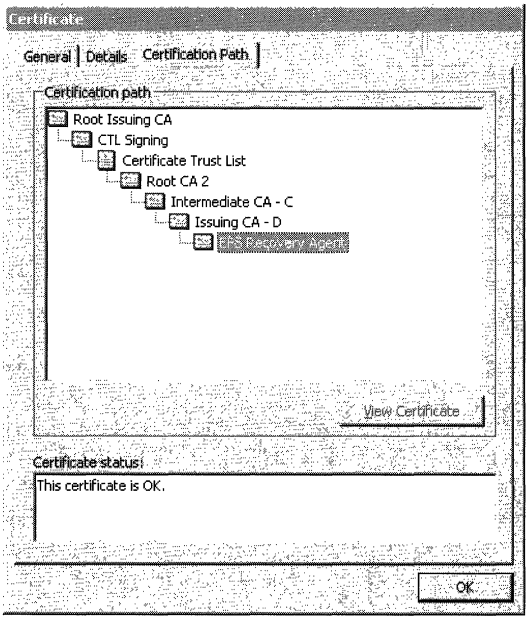


Figure 16.8 Trusted Certification Path with a CTL

In the example, the certification path from EFS Recovery Agent to Root CA 2 is identical to the certification path shown in Figure 16.6, but the certificate for Root CA 2 is not in the Trusted Root Certification Authorities store. The certification path also includes the CTL, the trust list signing certificate ("CTL Signing" in the example), and the root CA certificate that issued the signing certificate ("Root Issuing CA" in the example). The EFS Recovery Agent certificate is trusted because the certificate for Root Issuing CA (which issued the CTL Signing certificate) is contained in the Trusted Root Certification Authorities store.

A CTL must be signed by an administrator who has a valid certificate for trust list signing, such as the Administrator and Trust List Signing certificates that can be issued by enterprise CAs. By default, CTLs are valid until the trust list signing certificate expires and the CTL becomes invalid. However, to limit the time that certificates are trusted, you have the option of specifying a shorter lifetime for the CTL.

By default, members of the Domain Admins and Enterprise Admins security groups are granted permissions to enroll for Administrator and Trust List Signing certificates. To change the default certificate enrollment settings, modify the ACLs for the Administrator and Trust List Signing certificate templates.

For the CTL to be valid, the trust list signing certificate must have a certification path to a root CA in the Trusted Root Certification Authorities store. Figure 16.9 shows an example of a CTL that is invalid because the trust list signing certificate is invalid. This might be the situation because either the certification path for the trust list signing certificate does not validate to a trusted root certificate or the trust list signing certificate has expired.

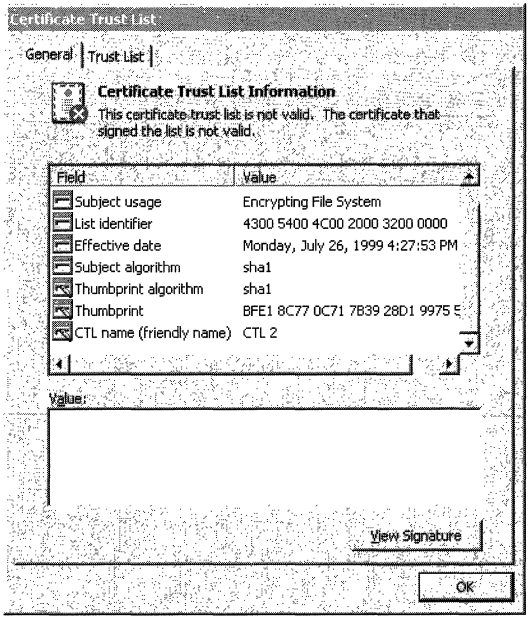


Figure 16.9 Invalid CTL

CTLs are stored in the Enterprise Trust store and you can use the Certificates console to view them.

In addition, you can use CTLs to restrict the purposes for which certificates can be used. For example, even though a certificate permits the purposes of software code signing, secure mail, and client authentication, you can use a CTL to restrict certificate use to client authentication only. CTLs are frequently used to restrict trust for certificates that are issued and managed by other organizations. For example, you might configure a CTL to trust a business partner's CA for only code signing and client authentication on an extranet that you manage.

Internet Information Services (IIS) also supports CTLs for secure Web sites. For more information about CTLs with IIS, see "Choosing Security Solutions That Use Public Key Technology" in this book.

Certificate Validation Process

Before it trusts certificates, Windows 2000 performs a validation check to ensure that certificates are valid and that they have a valid certification path. Figure 16.10 shows the basic certificate validation process.

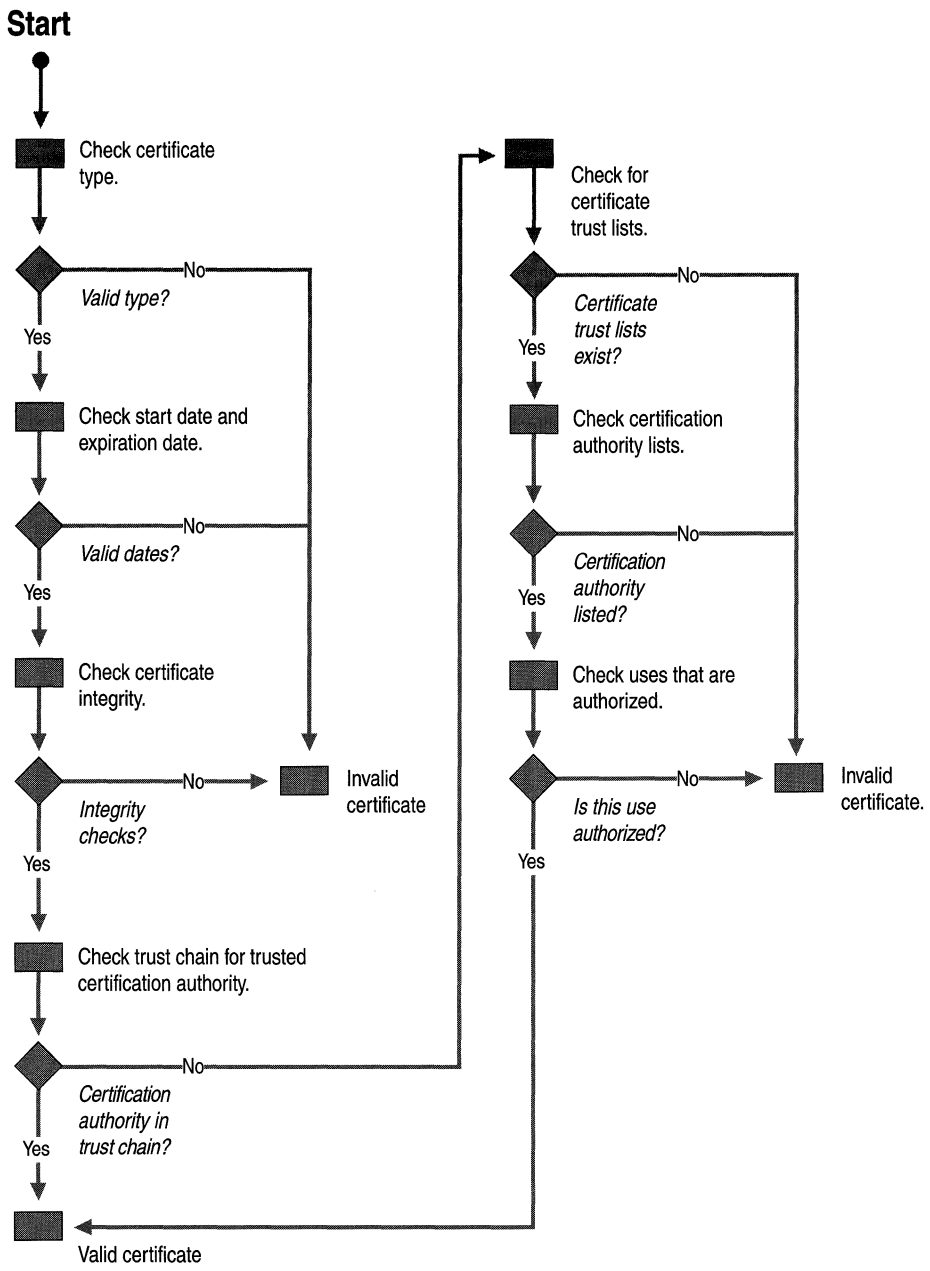


Figure 16.10 Basic Certificate Validation Process

Certificates can be invalid or are not trusted for a variety of reasons, including the following:

- The start and expiration dates are improper or expired.
- The certificate format is improper (does not conform to the X.509 version 3 standard for digital certificates).
- The information in certificate fields is improper or incomplete.
- The certificate's digital thumbprint and signature fail the integrity check, indicating that the certificate has been tampered with or corrupted.
- The certificate is listed as revoked in a published certificate revocation list.
- The issuing CA is not in either a trusted certification hierarchy or a CTL.
- The root CA for the certification path is not in the Trusted Root Certification Authorities store.
- The certificate is not permitted for the intended use as specified in a CTL.

An expired CA certificate in the certification path does not invalidate the path. In the Windows 2000 public key infrastructure, a certification path can be valid as long as the CA certificate was valid at the time the certificate was issued. For example, a third-party CA might issue a certificate with a lifetime that extends past the CA certificate's expiration date. After the CA's certificate expires, the certification path for the certificate is still valid and the certificate is trusted as long as all other validation criteria are met.

Benefits of Multiple-Level Certification Hierarchies

Consider deploying multiple-level certification hierarchies that include root CAs, intermediate CAs, and issuing CAs. Multiple-level trust hierarchies provide many benefits.

General Benefits

Deploying multiple-level certification hierarchies provides the following general benefits:

- They require trust in a relatively small number of root CAs that you can centrally control and maintain to ensure high security and integrity for root CAs.
- They reduce the cost and impact of a failed or compromised CA.
- They provide flexibility so business units can deploy and manage intermediate CAs to meet their public-key security needs.
- They provide flexibility so business units can deploy and manage issuing CAs to distribute certificate load and provide duplication of certificate services.

Administrative Benefits

Deploying multiple-level certification hierarchies provides the following administrative benefits:

- It enables flexible configuration of the CA security environment (key strength, physical protection, protection against network attacks, and so forth). You can tailor the CA environment to provide a balance between security and usability. For example, for a root CA, you might choose to use special purpose cryptographic hardware, maintain it in a locked vault, and operate it in offline mode. However, for an issuing CA, crypto-hardware, locked vaults, and offline operations are costly, make the CA difficult to use, and reduce the performance and effectiveness of the CA.
- It enables relatively frequent renewals of keys and certificates for those intermediate and issuing CAs that are at high risk for compromise, without requiring a change to established root trust relationships.
- It enables you to "turn off" a subsection of the CA hierarchy without affecting established root trust relationships or the rest of the hierarchy. For example, you can easily shut down an issuing CA that services one site, without affecting other certificate services for that site and without affecting certificate services for other sites.

Benefits of Multiple Issuing Certification Authorities

Deploying multiple issuing CAs provides several benefits, including the following:

- You can specify separate certificate policies for different groups of users or computers. You can deploy separate issuing CAs to administer separate certificate policies for each group of users and computers.
- You can specify separate certificate policies based on organizational divisions, such as a user's or computer's role in the organization. You can deploy issuing CAs to administer separate certificate policies based on such organizational divisions.
- You can specify separate certificate policies based on geographic divisions, such as the locations of users and computers at multiple physical sites.
- You can distribute certificate load and provide redundant services by deploying multiple issuing CAs to distribute the certificate load, meeting site, network, and server connectivity and load requirements. For example, slow or noncontinuous network links between sites might require issuing CAs at each site for acceptable certificate services performance and usability requirements. You can also deploy multiple issuing CAs to provide duplicate services so that if one CA fails, another issuing CA is available to provide uninterrupted service.

Windows 2000 Certification Authorities

Windows 2000 Server and Certificate Services support two types of CAs: enterprise CAs and stand-alone CAs. A root CA or a subordinate CA can be installed as either an enterprise CA or a stand-alone CA.

Enterprise Certification Authorities

Enterprise CAs are integrated with Active Directory. Enterprise CAs publish certificates and CRLs to Active Directory. Enterprise CAs use certificate template information, user account information, and security group information that are stored in Active Directory to approve or deny certificate requests. For a certificate request to be approved, the requestor must have Enroll permissions granted by the security ACLs of the certificate template for the certificate type that was requested. When a certificate is issued, the enterprise CA uses information in the certificate template to generate a certificate with the appropriate attributes for that certificate type.

It is recommended that you install most issuing CAs as enterprise CAs to gain the benefits of integration with Active Directory, including automated certificate approval and automatic computer certificate enrollment. Furthermore, only enterprise CAs can issue certificates for logging on with smart cards because this process requires that smart card certificates be mapped automatically to the user accounts in Active Directory and because it uses certificate templates.

Stand-alone Certification Authorities

Stand-alone CAs do not require Active Directory and do not use certificate templates. For stand-alone CAs, all information about the requested certificate type must be included in the certificate request. The Web Enrollment Support pages that are installed for stand-alone CAs, support requests for a variety of certificate types.

By default, all certificate requests submitted to stand-alone CAs are held in the Pending Queue until the CA administrator approves them. You can configure stand-alone CAs to issue certificates automatically upon request, but this adds a significant security risk and usually is not recommended.

If you want to automate certificate requests for stand-alone CAs, consider developing custom policy modules that securely approve or deny certificate requests. For example, you might develop a custom policy module that automatically grants certificates to authenticated requestors based on security information about the requestor that is contained in a legacy database or a third-party directory service. Stand-alone CAs cannot issue certificates for the smart card logon process, but they can issue other types of certificates for smart cards. For example, you can use the Web Enrollment Support pages for a stand-alone CA to issue secure mail and secure Web browser certificates to requestor's smart cards.

By default, stand-alone CAs publish CRLs to the following location:

```
<Drive:>\WINNT\System32\Certsrv\Certenroll
```

where *<Drive:>* is the letter of the disk drive where the CA is installed.

The use of stand-alone CAs for high-volume issuing usually incurs a high administrative cost because administrators must manually review and approve or deny each certificate request. Therefore, stand-alone issuing CAs are intended primarily for use with public key security applications on extranets and the Internet, when users do not have Windows 2000 accounts and the volume of certificates to be issued and managed is relatively low.

You must, however, install stand-alone CAs to issue certificates when you are using a third-party directory service or when Active Directory is not available. Furthermore, stand-alone CAs can provide more flexibility for planning and managing the certificate life cycle by using root CA and intermediate CAs.

Certificate Life Cycle

The certificate life cycle includes the following events:

- CAs are installed and their certificates are issued.
- Certificates are issued by CAs.
- Certificates are revoked (as necessary).
- Certificates are either renewed or allowed to expire.
- The CAs' certificates are renewed before they expire.
- The CA is revoked or retired.

Issued certificates expire at the end of their lifetime, but they can be renewed as necessary. You also can renew CAs before the CA's certificate expires to ensure continuous certificate services in your enterprise.

Windows 2000 CAs require nested validity dates for the certificate life cycle. A Windows 2000 CA cannot issue certificates with a lifetime that extends beyond the end date for the CA's certificate validity. If the lifetime specified for a requested certificate type exceeds the expiration date of the CA's certificate, the CA truncates the lifetime of the issued certificate to match the validity end date for the CA's certificate. Therefore, nested validity dates are an important consideration when you are planning the certificate life cycle for Windows 2000 Certificate Services CAs. Third-party CAs might not require nested lifetimes for the certificate life cycle.

The certificate lifetimes of certificates that are issued by enterprise CAs are determined differently than the lifetime of certificates that are issued by stand-alone CAs. An enterprise CA issues certificates with lifetimes that are based on the certificate template for the requested certificate type. A stand-alone CA issues certificates with a lifetime that is determined by system registry settings for the CA. Furthermore, the lifetime of CA certificates is affected by several other factors. In addition, take into account how long private keys can be safely used so that you do not exceed the maximum safe lifetime of the keys.

Nested Validity Dates

Windows 2000 enterprise CAs and stand-alone CAs require nested validity dates for all CA certificates and all issued certificates. For example, if a Windows 2000 root CA's certificate end date is January 2, 2010, no Windows 2000 child CA in the chain below the root can issue a certificate with a date that is past January 2, 2010. If a Windows 2000 intermediate CA has a certificate end date of January 2, 2006, no Windows 2000 child CA can issue certificates with an end date that is past January 2, 2006. If a Windows 2000 issuing CA has a certificate end date of January 2, 2002, no certificate the CA issues can have an end date that is past January 2, 2002.

If a Windows 2000 CA's certificate has an end date of January 2, 2002, and it receives a request to issue a one-year certificate on August 1, 2000, the CA issues the one-year certificate with an end date of July 31, 2001. However, if the CA receives a request to issue a one-year certificate on August 1, 2001, the CA issues the certificate with an end date of January 2, 2002.

A Windows 2000 CA with a certificate life of five years ending on January 2, 2005, can issue one-year certificates until January 2, 2004, or two-year certificates until January 2, 2003. After January 2, 2003, the CA does not issue two-year certificates; it truncates the validity end date to January 2, 2005. Likewise, after January 2, 2004, the CA truncates the validity end date of both one-year and two-year certificates to January 2, 2005.

You usually renew Windows 2000 CAs with new CA certificates before they are constrained by nested validity dates. To avoid the constraints of nested validity dates, deep certification hierarchies with Windows 2000 Certificate Services might require frequent renewals for issuing CAs.

Certificates Issued by Stand-alone Certification Authorities

For stand-alone CAs, the lifetime of issued certificates is determined by the following registry entries:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\CertSvc
\Configuration\Stand-aloneCA\ValidityPeriod
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\CertSvc
\Configuration\Stand-aloneCA\ValidityPeriodUnits
```

Caution Do not use a registry editor to edit the registry directly unless you have no alternative. The registry editors bypass the standard safeguards provided by administrative tools. These safeguards prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your system. Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. To configure or customize Windows 2000, use the programs in Control Panel or Microsoft Management Console (MMC) whenever possible.

Where *Stand-aloneCA* is the name of the installed CA, the value of **ValidityPeriod** is either "Days," "Weeks," "Months," or "Years," and **ValidityPeriodUnits** is the number of days, weeks, months or years that constitute the lifetime of certificates issued by the CA. For example, when the value of **ValidityPeriod** is "Years" and the decimal value of **ValidityPeriodUnits** is "2," the CA issues certificates with a lifetime of two years.

By default, stand-alone CAs issue certificates with lifetimes of one year. (The default settings are: **ValidityPeriod** = Years and **ValidityPeriodUnits** = 1,) To specify another lifetime for certificates that are issued by a stand-alone CA, edit the registry for the stand-alone CA, and enter the appropriate values for **ValidityPeriod** and **ValidityPeriodUnits**.

All certificates that the stand-alone CA issues have the lifetime specified by the values of the **ValidityPeriod** and **ValidityPeriodUnits** registry entries. Therefore, if you want to issue certificates with different lifetimes, you must deploy either enterprise CAs, multiple stand-alone CAs, or third-party CAs.

Certificates Issued by Enterprise Certification Authorities

For enterprise CAs, the maximum lifetime of certificates that are issued is determined by the settings of **ValidityPeriod** and **ValidityPeriodUnits** in the registry. The default settings are: **ValidityPeriod** = Years and **ValidityPeriodUnits** = 2. Therefore, the maximum lifetime of certificates that are issued by an enterprise CA is two years unless you modify the registry settings.

In addition, the lifetime of each certificate type is determined by its certificate template. The lifetime for many certificate types is one year. However, the following certificate templates specify a lifetime of two years:

- CEP Encryption (offline request)
- Enrollment Agent
- Enrollment Agent (computer)
- Enrollment Agent (offline request)
- IPSec
- IPSec (offline request)
- Router (offline request)
- Web Server

The following certificate templates specify a lifetime of five years:

- Domain Controller
- Subordinate Certification Authority

These certificates are usually issued for two years (the maximum default lifetime of certificates issued by enterprise CAs). To enable an enterprise CA to issue certificates for five years, you must change the settings of **ValidityPeriod** and **ValidityPeriodUnits** for the CA to five years or more.

In addition, you can modify **ValidityPeriod** and **ValidityPeriodUnits** of a CA to reduce the maximum lifetime of certificates that it issues. For example, to reduce the maximum lifetime of all certificates issued by a CA to six months, you can change **ValidityPeriod** to "Month" and **ValidityPeriodUnits** to "6". You can also deploy custom certificate services to meet special certificate lifetime needs for your organization.

Certification Authorities' Certificates

For enterprise root CAs and enterprise stand-alone root CAs, the CA certificates are installed with a default lifetime of two years. However, during CA installation, you can specify a different lifetime for the CA. You can specify the root CA's lifetime in days, weeks, months, or years. For example, you might specify a root CA lifetime of 20 years because you use a large private key and provide high security for the CA. You might also want to specify short lifetimes of days or weeks when you are testing the deployment of Certificate Services.

During the installation of subordinate CAs, the system enables you to request a subordinate CA certificate from an active parent CA, or you have the option of creating a certificate request file and then submitting the request offline to a parent CA. For online requests from an active CA, when the request is approved, the subordinate CA is issued a subordinate CA certificate automatically by the parent enterprise CA. For offline requests, you must use the Web Enrollment Support pages to submit the certificate request file to the parent CA. After the subordinate CA certificate is issued, you must use the Certification Authority console to install the certification path file to certify and start the CA.

The lifetime of a subordinate CA certificate is determined by the parent CA that approves the certificate request and issues it. If the parent CA is an enterprise CA, the default lifetime of the subordinate CA's certificate is two years, unless the **ValidityPeriod** and **ValidityPeriodUnits** values are changed in the registry for the parent CA. You can change the registry to specify a shorter or longer lifetime for certificates that are issued by the parent CA, but the maximum lifetime for subordinate CA certificates is five years as specified by the Subordinate Certification Authority certificate template. If the parent CA is a stand-alone, the lifetime of the subordinate CA's certificate is determined by the values of the **ValidityPeriod** and **ValidityPeriodUnits** entries in the registry of the parent CA.

Consider using stand-alone CAs for root and intermediate CAs to provide the most flexibility for defining certificate life cycles. If you specify long lifetimes for CAs and later discover that they are at greater risk than originally anticipated, it is easy to renew CAs in the certification hierarchy with shorter lifetimes as necessary to reduce risk.

Using stand-alone CAs for root and intermediate CAs can provide other benefits as well. If you operate stand-alone CAs offline (not connected to the network) and maintain them in secure physical environments, the risk of attacks is reduced. You also can regulate the installation process to carefully control the CAs that are installed and trusted in the enterprise.

Administering offline certificate requests for both stand-alone root and intermediate CAs is usually cost effective because the CAs are used infrequently to process relatively few certificate requests. You might, however, occasionally connect to the network only as long as necessary to publish CRLs or to process infrequent online certificate requests for subordinate CA certificates.

Example of a Certificate Life Cycle

Table 16.4 describes an example of a certificate life cycle that an organization might plan for Windows 2000 CAs and standard Microsoft CSPs.

Table 16.4 Windows 2000 Certificate Life Cycle

Purpose of Certificate	Certificate Life	Private Key Life
Stand-alone root CA. (4,096-bit key)	20 years	Renew at least every 10 years to ensure that intermediate CA certificates can be issued with lifetimes of 10 years. Renew by using a new key at least every 20 years.
Stand-alone intermediate CA for all certificates except smart card certificates. (3,072-bit key)	10 years	Renew at least every 5 years to ensure that child issuing CAs can be issued for a full 5 years. Renew by using a new key at least every 10 years.
Enterprise issuing CA for all certificates except smart card certificates. (2,048-bit key)	5 years	Renew at least every 3 years to ensure that Web server certificates can be issued for a full 2 years. Renew by using a new key at least every 5 years.
Enterprise issuing CA 2 for smart card certificates. (2,048-bit key)	5 years	Renew at least every four years to ensure that certificates can be issued for a full year. Renew by using a new key at least every 5 years.
Enterprise issuing CA 3 for all other certificates besides smart cards, secure mail, and secure browser certificates. (2,048-bit key)	5 years	Renew at least every 4 years to ensure that certificates can be issued for a full year. Renew by using a new key at least every 5 years.
Secure mail and secure browser certificates.	1 year	Renew by using a new key at least every 2 years.
Smart card certificates. (1,024-bit key)	1 year	Renew by using a new key at least every 2 years.
Administrator certificates. (1,024-bit key)	1 year	Renew by using a new key at least every 2 years.
Secure Web server certificates. (1,024-bit key)	2 years	Renew by using a new key at least every 2 years.
Business partners' users certificates for an extranet. (512-bit key)	6 months	Renew by using a new key at least every year.

Note The certificate life cycle described in Table 16.4 is provided only as an example and is not intended to be a recommendation. Your certificate life cycle can differ from the example in many ways, including the length of certificate lifetimes, key lengths, and key lifetimes.

In Table 16.4, all certificates are issued by Windows 2000 CAs except for the certificates for the business partners' users (for the extranet), which are issued by the CA of the business partner. The certificates of the business partner are trusted in the extranet domain by using CTLs. Stand-alone CAs are used to provide flexible lifetimes for CAs where this is appropriate. Renewing certificates with new keys limits the time that keys are in use and reduces the risk of key compromise.

Because of the constraints of nested validity dates, when you allow CAs to issue certificates with truncated lifetimes, the certificates that are issued must be renewed more frequently as the end validity date of the CA's certificate is approached. Therefore, CAs are usually renewed before the certificates that are issued by the CA have truncated lifetimes. You also renew certificates with new keys before their maximum safe lifetime are exceeded. To reduce risks for private keys, you might also renew certificates with new keys each time the certificate is renewed if it is feasible to do so.

The deeper the certification hierarchy, the shorter the certificate lifetimes become. Plan your certificate life cycles to avoid excessively short certificate lifetimes and certificate renewal cycles.

General Considerations for Key Lifetimes

There is no simple formula for determining maximum private key lifetimes. The lifetimes you choose depend on various risk factors, such as the following:

- The length of private keys for certificates. In general, longer keys support longer key lifetimes.
- Security provided for private keys by the CSPs. In general, hardware-based CSPs provide more security and can support longer private key lifetimes than software-based CSPs.
- Security provided for CAs and their private keys. In general, the more secure the CA and its private key, the longer the safe CA lifetime. For example, you might improve the security for CAs by operating them offline and storing them in locked vaults or data centers.

- The strength of the cryptographic technology used for cryptographic operations. Some cryptographic technologies provide stronger security, as well as support for stronger cryptographic algorithms. For example, you might use smart cards for logging on by users or FORTEZZA Crypto Cards for secure mail and secure Web browsers. In general, stronger cryptographic technology supports longer key lifetimes.
- The risk of attack on the CA certification chain. These risks depend primarily on how secure your enterprise is, how valuable the network resources protected by your public key security applications are, and how much launching attacks would cost the attackers. In general, high risks of attack require longer CA private keys and shorter key lifetimes.

To further reduce the risk of a compromised private key, the private key and public key sets for certificates might be renewed each time the certificates are renewed, instead of waiting for the maximum key lifetime. However, for some hardware-based CSPs, renewing certificates with new key sets is not feasible either because of key storage limits or because key generation takes a long time.

When you install a Windows 2000 CA, you can select the **Advanced options** check box on the first page of the Windows Components wizard, with which you can specify the key length that is used with the CA's certificate. You can select CA key lengths from 384 bits to 16,384 bits. In general, the longer the key, the longer the safe key lifetime. The use of keys that are at least 1,024 bits long is recommended for CAs.

Consider using the largest keys that are practical to use for CAs to provide the maximum protection feasible without degrading CA performance. Keep in mind that very large keys can place a high load on computer processors and might require excessive amounts of time for signing operations. Test proposed CA key lengths in the lab and pilot programs before you deploy CAs to your production environment.

For more information about the risks associated with private keys, see "Cryptography for Network and Information Security" in this book.

When you renew certificates by using the Microsoft CSPs, you also can renew the certificate's private key and public key set. In general, the longer the key set is in use, the higher the risk that the key might become compromised. Establish maximum allowable key lifetimes, and renew certificates with new key sets before these limits are exceeded.

Certificate Enrollment and Renewal Methods

Windows 2000 Certificate Services supports the following certificate enrollment and renewal methods:

- Manual certificate requests that use the Certificate Request wizard (only for Windows 2000 users and computers).
- Automatic certificate requests, which use the Automatic Certificate Request Setup wizard (only for Windows 2000 computer certificates).
- Manual certificate requests that use the Web Enrollment Support pages (for Web browser users).
- Smart card enrollment, which uses the Smart Card Enrollment Station available in the Web Enrollment Support pages.
- Custom certificate enrollment and renewal applications.

The enrollment methods and types of certificates that are supported by third-party certificate services depend on the features and functions of each third-party product. For more information, contact the vendor for the certificate service.

Manual Certificate Requests for Windows 2000–based Clients

You can request or renew certificates for Windows 2000 users and computers by using the Certificate Request wizard that is available in the Certificates console. The Certificate Request wizard does not function unless an enterprise CA is online to process and issue certificate requests. The ACLs for the certificate templates determine which user accounts or computer accounts can enroll for the various types of certificates.

You can also use the Certificate Renewal wizard that is available in the Certificates console to renew certificates either before or after they expire. The Certificate Renewal wizard does not function unless an enterprise CA is online to process and issue certificate requests. You have the option of renewing certificates with the same private key and public key set. You must not renew certificates with the same private and public key sets if the maximum safe key lifetime would be exceeded.

Automatic Computer Certificate Enrollment and Renewal

You can use the Automatic Certificate Request Setup wizard (available from the Public Key section of the Group Policy console) to configure autoenrollment for computer certificates. Autoenrollment is not available for user certificates and does not function unless an enterprise CA is online to process certificate requests. You can configure autoenrollment for Computer, Domain Controller, and IPSec certificates.

When autoenrollment is configured, the specified certificate types are issued automatically to all computers that are within the scope of the Public Key Group Policy and to all computers that have Enroll permissions for that certificate type. Autoenrollment certificates are issued the next time the computer logs on to the network.

For example, if you configure autoenrollment for Computer certificates, the certificates are issued to all computers in the Domain Computers security group that are within the scope of the Public Key Group Policy. By default, all Windows 2000 computers are members of the Domain Computers security group, except for domain controllers, Routing and Remote Access servers, and Internet Authentication Services (IAS) servers. You can control which computers receive the Computer certificates by modifying the ACLs for the Computer certificate templates, for example, to grant Enroll permissions to a special security group composed of computers that you designate. Computers within the scope of the Public Key Group Policy that are members of the special security group are then issued Computer certificates the next time they log on to the network.

In addition, you also can use organizational units (OUs) and Public Key Group Policy for those OUs to restrict autoenrollment to certain groups of computers. For example, you might create an IPSec Authentication OU that contains the Windows 2000 clients that you designate for IPSec authentication with certificates. To limit the scope of autoenrollment for IPSec certificates, configure Public Key Group Policy and autoenrollment for the IPSec Authentication OU.

When autoenrollment is configured, the Computer certificates that are issued by autoenrollment also are automatically renewed from the enterprise issuing CA. You can also renew Computer certificates manually with the Certificate Renewal wizard or through the Certificate Services Web Enrollment Support pages.

Web Enrollment Support Pages

The Windows 2000 Certificate Services Web Enrollment Support pages are composed of Active Server Pages and ActiveX® controls that provide a Web-based user interface to a CA. By default, the Web Enrollment Support pages are automatically installed on the computer where the CA is installed, but you also have the option of installing the Web Enrollment Support pages on another Windows 2000 Server computer.

You can use the Web Enrollment Support pages to perform the following tasks:

- Request and obtain a basic user certificate.
- Request and obtain other types of certificates by using advanced options.
- Request a certificate by using a certificate request file.
- Renew certificates by using a certificate renewal request file.
- Save a certificate request to a file.
- Save the issued certificate to a file.
- Check on pending certificate requests.
- Retrieve the CA's certificate.
- Retrieve the latest certificate revocation list from the CA.
- Enroll for smart card certificates on behalf of other users (for use by trusted administrators).

The Web Enrollment Support pages that are installed for stand-alone CAs are similar to the pages that are installed for enterprise CAs, but they differ in the respect that stand-alone CAs do not use certificate templates. For stand-alone CAs, all information about the certificate, including information about the requestor, must be specified in the certificate request. The Web Enrollment Support pages for stand-alone CAs support a number of types of certificates that have much of the same functionality as certificate types that are based on templates. You can deploy stand-alone CAs and Web Enrollment Support pages to issue most of the types of certificates that enterprise CAs can issue. However, certificates for logging on by using smart cards logon and for autoenrollment require an enterprise CA to issue and renew the certificates.

The Web Enrollment Support pages work with Microsoft® Internet Explorer 4 and Microsoft® Internet Explorer 5. Use of the Microsoft Enhanced Cryptographic Provider requires Internet Explorer browsers with nonexportable cryptography. Internet Explorer browsers with exportable cryptography work only with the Microsoft Base Cryptographic Provider.

Netscape Navigator version 4.*x* and Netscape Communicator version 4.*x* work with most of the Web Enrollment Support pages. Netscape browsers do not work with the Advanced Certificate Requests form and the Smart Card Enrollment Station page because these pages use ActiveX controls. In addition, Netscape browsers use their own cryptographic security modules rather than CSPs and, therefore, might not support all of the features that are available for the Microsoft CSPs.

Custom Enrollment and Renewal Applications

The standard enrollment and renewal methods that are available in Windows 2000 can meet a wide range of needs. However, if you have special needs, you can develop custom certificate enrollment and renewal applications. The Windows 2000 Certificate Services Entry module supports industry-standard certificate requests by using remote procedure call (RPC) requests or HTTP requests. You can develop custom applications that make certificate requests to Certificate Services CAs. For more information about developing custom applications with Windows 2000 Certificate Services, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Public Key Group Policy

Public Key settings are a subset of Group Policy. You can configure Public Key Group Policy to specify automatic enrollment for computer certificates, trusted root certificates, CTLs for computers and users, and EFS recovery agents and apply the Group Policy to sites, domains, or organizational units.

The Group Policy console is an MMC snap-in. You can use MMC to manage Public Key Group Policy for multiple sites, domains, and organizational units. You can configure Public Key Group Policy separately for users and for computers. You can use the Group Policy console to configure the following Public Key Group Policy settings for computers:

- Specify the certificates in Trusted Root Certification Authorities stores.
- Create CTLs to trust CAs and restrict the uses of certificates issued by the CAs.
- Specify automatic enrollment and renewal for computer certificates.
- Specify alternative Encrypted Data Recovery Agents for EFS.

Public Key Group Policy settings apply for computers within the scope of the Group Policy. For example, you can create an organizational unit and configure Public Key settings that apply only to the computers in that organizational unit.

You also can use the Group Policy console to configure CTLs that apply only to users within the scope of the Group Policy. For example, you can create an organizational unit and configure CTLs that apply only to the users in that organizational unit. For more information about Group Policy, see "Group Policy" in this book and Group Policy Reference.

Certificate Revocation Lists

Windows 2000 supports industry standard X.509 version 2 CRLs. Each CA maintains a CRL for the certificates it issues and publishes the CRL-to-CRL distribution points. CRL distribution points can include Web pages, network shares, or Active Directory. An X.509 version 3 certificate usually contains the CRL distribution point for its issuing CA.

By default, enterprise CAs publish CRLs weekly to Active Directory and stand-alone CAs publish CRLs weekly to the following folder on the CA server:

```
<Drive:>\WINNT\System32\Certsrv\Certenroll
```

where *<Drive:>* is the letter of the disk drive where the CA is installed.

You can use the Certification Authority console to modify the *CRL distribution points*. You also can use the Certification Authority console to manually publish a new CRL or to change the publication schedule.

Certificate revocation checking is supported by Internet Explorer 5, Internet Information Services, and Active Directory mapping services. When revocation checking is enabled, you have the option of caching CRLs on local computers to enhance revocation checking performance. If a certificate lists the CRL distribution point, the revocation checking process checks the local cache to determine whether the CRL is in the cache. If not, the revocation checking process then checks the network for the CRL. If a certificate does not list the CRL distribution point, revocation checking checks the issuing CA for a CRL, if one is available. You also can use the Web Enrollment Support pages to request the latest CRL from a CA.

When revoked certificates expire, they are removed from the next published CRL. For some large organizations with high certificate revocation rates, CRLs might become so large that it places a significant load on the network and computers during CRL publication. However, you can prevent large CRLs by deploying multiple issuing CAs to distribute the certificate load among your users and by issuing certificates with reasonably short lifetimes.

Preinstalled Trusted Root Certificates

The root CA certificates that are contained in the Trusted Root Certification Authorities store are trusted for all Windows applications that use public key certificates for security functions. Windows 2000–based computers include many preinstalled certificates in the Trusted Root Certification Authorities stores. The preinstalled trusted root certificates include root certificates from a variety of commercial CAs and Microsoft. Certificates that are issued by these trusted CAs are trusted on local computers for valid purposes. However, you might not want to trust the preinstalled root certificates, or you might want to add other certificates as trusted root certificates.

You can use the Certificates console to delete or add certificates manually for Trusted Root Certification Authorities stores on each local computer. You also can add trusted root certificates for groups of computers by using Public Key Group Policy.

In addition, you can use the Internet Explorer Administration Kit (IEAK) to create and deploy custom builds of Internet Explorer that have only the root certificates that you want for your enterprise. For example, you can create custom builds that include only a few trusted root certificates and then deploy those custom builds to groups of computers. The computers where the custom builds of Internet Explorer are installed have only the trusted root certificates that you specified. You can create different custom builds to meet the requirements of different groups in your organizations. For more information about using the IEAK, see the *Microsoft® Windows® 2000 Server Resource Kit Internet Explorer Resource Guide*.

Smart Card Support

Smart cards are credit card–sized and contain integrated circuit cards (ICCs). They can be used to store certificates and private keys and to perform public key cryptography operations, such as authentication, digital signing, and key exchange. Smart cards offer the following security enhancements and benefits:

- They provide tamper-resistant storage for protecting private keys and other forms of personal information.
- They isolate security-critical computations involving authentication, digital signatures, and key exchange from other parts of the system that do not have a specific purpose for this data.
- They enable the portability of credentials and other private information between work, home, and remote computers.

In addition, smart cards use Personal Identification Numbers (PINs) rather than passwords. The smart card is protected from misuse by the PIN, which is known only to the owner of the smart card. To use the smart card, a user inserts the card in a smart card reader that is attached to a computer and, when prompted, enters the PIN. The smart card can be used only by someone who possesses the smart card and knows the PIN.

PINs offer more protection than standard network passwords. Passwords (or derivations such as hashes) travel on the network and are subject to brute force or dictionary attacks. The strength of the password depends on its length, how well it is protected, and how difficult it is to guess. In contrast, PINs never travel on the network and cannot be sniffed. Furthermore, dictionary attacks or brute force (key search) attacks (where an attacker tries numerous PIN combinations in an attempt to "guess" the PIN) can be attempted only by someone in physical possession of the smart card. And, the smart card locks after only a few failed attempts to guess the PIN.

Windows 2000 supports industry standard Personal Computer/Smart Card (PC/SC)-compliant Plug and Play smart cards and smart card readers that conform to specifications that have been developed by the PC/SC Workgroup. To work under the Windows implementation of the PC/SC 1.0 Specification, a smart card must conform physically and electrically to the International Standards Organization (ISO) 7816-1, 7816-2, and 7816-3 standards.

Smart card readers attach to standard personal computer peripheral interfaces such as RS–232, PS/2, PCMCIA, and Universal Serial Bus (USB). Readers are considered standard Windows 2000 devices, and they carry a security descriptor and a Plug and Play identifier. Smart card readers are controlled through standard Windows device drivers and are installed and removed by using the Hardware wizard.

Windows 2000 includes drivers for various commercially available Plug and Play smart card readers that are certified to display the Windows-compatible logo. Some manufacturers might provide drivers for noncertified smart card readers that currently work with Windows 2000. Nevertheless, to ensure continuing support by Microsoft, it is recommended that you purchase only those smart card readers that display the Windows-compatible logo.

The Windows 2000 CSPs includes smart card CSPs from Gemplus SCA and Schlumberger Limited. These CSPs support smart cards from the respective vendors and work with all smart card readers that display the Windows-compatible logo. The smart card CSPs store the issued certificate and the private key on the smart card.

Each smart card vendor provides software that you must install and use to initialize and configure smart cards before they can be deployed. You can use the vendor's software to configure PINs and to configure the number of PIN attempts that are allowed to occur before the smart card locks. You also can use the vendor's software to return locked smart cards to service.

For more information about smart cards, see "Choosing Security Solutions That Use Public Key Technology" in this book.

Certificate Mapping

You can use certificate mapping to control access to network resources for domain user accounts. You also can use certificate mapping to control access to Web site resources for Internet Information Services.

Domain User Accounts

You can use the Active Directory Users and Computers console (an MMC snap-in) to map user certificates to individual network user accounts. The mapped certificates are used to authenticate users during the Kerberos authentication process. Authenticated users are granted the rights and permissions for user accounts on the basis of ownership of valid certificates. Smart card logon certificates are a special type of mapped certificate. During the smart card logon enrollment process, the system maps the smart card certificates to the users' corresponding Windows 2000 user accounts automatically.

Before you can map certificates, you must point to the **Active Directory Users and Computers** console, and then click **View** and **Advanced Features**. To map certificates, right-click a user account, and then click **Name Mappings**. When the **Security Identity Mapping** dialog box appears, click **Add** to import the certificates that you want to map to the user account. You can map multiple certificates to a user account. For example, you might issue EFS Recovery Agent certificates to smart cards for designated recovery agents and then map the smart card certificates to EFS recovery user accounts. The smart cards are then required to authenticate the EFS recovery agents when logging on to the network for the EFS recovery accounts, providing an additional level of security for them.

You can map certificates only to individual user accounts; not to security groups. If you map certificates that are not stored on smart cards, users can log on only to the mapped user account from the computer where the private key is located, unless smart cards or roaming profiles are being used.

Internet Information Services

For Internet Information Services, you can map certificates to user accounts that control access to Web resources. The mapped certificates are used either to deny access to Web resources or to grant rights and permissions for the mapped user account. You can map one certificate to one user account (one-to-one mapping) or you can map many certificates to one user account (many-to-one mapping). Many-to-one mapping uses rules to define the certificate criteria for mapping. If certificates match the rules, they are mapped to the appropriate account. For example, you can define rules that map certificates to different user accounts on the basis of the specific CA that issued the certificate. All clients with certificates that are issued by a qualifying CA are mapped to the appropriate user account and granted the respective rights and permissions for that account.

For more information about certificate mapping with Internet Information Services, see "Choosing Security Solutions That Use Public Key Technology" in this book.

Roaming Profile Support

Windows 2000 supports roaming user profiles, which allow certificates to follow users no matter which computer they use to log on. When roaming profiles are enabled, user profiles, including issued certificates and private keys, are stored on the domain controller. The roaming profiles are downloaded to the computer during the logon process for the user. Smart cards also provide roaming capabilities because a user's logon credentials are stored on the smart card. For more information about roaming profiles, see "Introduction to Desktop Management" in this book and Windows 2000 Server Help.

Certificate Services Deployment

You can perform the following activities to deploy Windows 2000 Certificate Services:

- Install certification authorities.
- Configure certification authorities.
- Modify the default security permissions for certificate templates (optional).
- Install and configure support systems or applications.
- Configure Public Key Group Policy.
- Install Web Enrollment Support on another computer (optional).
- Configure security for Web Enrollment Support pages (optional).

For more information about how to install Windows 2000 Certificate Services, see Certificate Services Help. For more information about planning the deployment of the public key infrastructure, see "Planning Your Public Key Infrastructure" in the *Deployment Planning Guide*.

Install Certification Authorities

You must install the CA hierarchies necessary to provide the required certificate services for your organization. Certification hierarchies with Windows 2000 CAs can include a mixture of enterprise CAs and stand-alone CAs. You can install the root CA first and then each subordinate CA in the hierarchy. For example, to create a three-level certification hierarchy, you can install CAs on servers in the following order:

1. Root CA
2. Intermediate CAs
3. Issuing CAs

However, to install the CA software on computers, you are not required to install CAs in this order. Root CAs are certified by self-signed certificates, so they do not depend on another CA to complete the installation. However, the complete installation of child CAs requires the parent CA to process the certificate request and issue the subordinate CA certificate. You can install a subordinate CA at any time, save the certificate request to a file, and submit it to the parent CA later, after the parent CA is installed and running. After parent CAs are installed and running, you can submit the certificate request file by using the Advanced Certificate Request Web pages for the parent CA. After the certificate for the child CA is issued, you can install the certificate for the child CA by using the Certification Authority console. A CA must have a valid CA certificate to start.

Although you can install CAs on domain controllers, it is not a recommended practice. To distribute the network load and prevent excessive load conditions on computers, install CAs on Windows 2000 Server–based computers that are dedicated to providing CA services. Also consider installing the Web Enrollment Support pages on separate Windows 2000 Server–based computers.

For information about installing third-party CAs and using them with Windows certification hierarchies, see the documentation for the third-party CA product.

Upgrading from Certificate Server 1.0

If you upgrade a Windows NT 4.0–based server that is running Certificate Server 1.0 to Windows 2000 Server, Certificate Server 1.0 is upgraded automatically to the new version of Certificate Services. If the CA being upgraded is using a policy module other than the default policy module for Certificate Server 1.0, it continues to use its old policy module, which is referred to as the Legacy policy module. If the CA you are upgrading uses the default policy module that was provided with Certificate Server 1.0, the upgraded CA uses the Certificate Services stand-alone policy.

If you are not upgrading a Certificate Server 1.0 CA and, instead, are installing a separate Windows 2000 CA that is to replace the old CA, you might want to use the older policy module instead of the default policy module that is provided with Certificate Services. If you want to replace the policy module that is provided with Certificate Services with a custom policy module or a policy module developed for Certificate Server 1.0 and Windows NT 4.0, you must first register the policy module DLL file by using the **Regsrv32** command, and then select the policy module by using the Certification Authority console. For more information about using Regsv32 and selecting policy modules, see Windows 2000 Server Help and Certificate Services Help.

Creation of an Issuer Statement for the Certification Authority (Optional)

When you install a CA, you have the option of adding an issuer statement for the CA that appears when users click **Issuer Statement** in the **Certificate General** dialog box. The issuer statement is a policy statement that gives legal and other pertinent information about the CA and its issuing policies, limitations of liability, and so forth.

The issuer statement file must be installed on the server before you install Windows 2000 Certificate Services. This file, named Capolicy.inf, must be placed in the directory in which Windows 2000 Server is installed—the *systemroot* directory. (The default *systemroot* is C:\Winnt.) CAPolicy.inf can contain the text you want to be displayed as the policy statement, or it can contain a URL that points to the policy statement, for example, a Web page. For more information about how to create the Capolicy.inf file, see Certificate Services Help.

Installing Windows 2000 Certificate Services

Before you can install a CA, you must be logged on as either a member of the local Administrator security group for stand-alone computers or a member of the Domain Administrator security group for computers that are connected to the domain.

▶ **To install Windows 2000 Certificate Services**

1. In Control Panel, click **Add/Remove Programs**.

   The **Add/Remove Programs** dialog box appears.

2. Click **Add/Remove Windows Components**.

   The Windows Component wizard appears.

3. In Windows Components, select the Certificate Services check box.
4. Click **Next**, and use the Windows Component wizard to install the CA.

Tables 16.5 through 16.9 describe the available CA configuration options for each page of the Windows Component wizard.

Note After the CA is installed, the computer cannot be renamed, joined to a domain, or removed from a domain. Installing an enterprise CA requires Active Directory, so the CA computer must already be joined to the Windows 2000 domain.

Table 16.5 Certification Authority Type Selection Page

Option	Description
Enterprise root CA	Select to install an enterprise root CA.
Enterprise subordinate CA	Select to install an enterprise subordinate CA.
Stand-alone root CA	Select to install a stand-alone root CA.
Stand-alone subordinate CA	Select to install a stand-alone subordinate CA.
Advanced options	Select to configure advanced options in the **Public and Private Key Selection** page of the wizard.

Table 16.6 Public and Private Key Selection Page

Option	Description
Cryptographic service provider	Select the CSP to be used to generate the public key and private key set for the CA certificate. This CSP also manages and stores the private key. The default CSP is the Microsoft Base Cryptographic Provider or the Microsoft Enhanced Cryptographic Provider, depending on whether the server that is running Windows 2000 contains exportable or nonexportable cryptography. If you want to use another CSP, such as a hardware-based CSP to manage and store the CA's private key, you must select the appropriate CA from the list of CSPs.
Hash algorithms	Select the message digest that is to be used for the digital signature of the CA certificates. The default is SHA-1, which provides the strongest cryptographic security.
Key length	Select a key length from the list, or type a key length for the private key and public key. The default key length is 512 bits for the Base Cryptographic Provider and 1,024 bits for the Enhanced Cryptographic Provider. The minimum key length you can specify is 384 bits, and the maximum is 16,384 bits. Use a key of at least 1,024 bits for CAs. In general, the longer the key, the longer the safe lifetime of the private key. Use the longest key that is feasible and that meets both CA performance requirements and CSP key storage limitations.
Use existing keys	Enables the selection of an existing private key from the list. The existing private key is used for the CA. You might need to use this option to restore a failed CA.
Use the associated certificate	Enables the selection of the certificate that is associated with the existing private key which is used for the CA. This option is not available unless you first select **Use the associated certificate**. You might need to use this option to restore a failed CA.
Import	Imports a private key that is not in the **Use existing keys** list. For example, you might import a private key from an archive for a failed CA.
View Certificate	Select this option to view the certificate associated with the private key in the **Use existing keys** list.

Table 16.7 CA Identifying Information Page

Option	Description
CA name Organization Organizational unit Locality State or province Country/region E-mail	Enter information that is to be used to uniquely identify the CA. This information is included in the CA certificate in the Subject field. The **CA name** that you enter here is used by Windows 2000 to identify the CA, so the **CA name** must be unique for each CA you install in your organization. However, all of the other information that is entered here can be the same if appropriate. Others can view the Subject field in the CA certificate to identify the CA or to find out how to contact the CA.
CA description	Enter a description for this CA (optional).
Validity duration	Enter the duration for the certificate lifetime for the root CA certificate, and select **Years**, **Months**, or **Weeks** from the list. The default certificate lifetime for root CAs is 2 years. You must choose a lifetime that supports your planned certificate life cycles. This option is not available for subordinate CAs because the certificate lifetime is determined by the parent CA.
Expires on	Lists the expiration date for the root CA certificate, which corresponds to the certificate lifetime in **Validity duration**.

Table 16.8 Data Storage Location Page

Option	Description
Certificate database Certificate log	By default, the certificate database and the log are installed at *<Drive:>*\WINNT\System32\CertLog, where *<Drive:>* is the letter of the disk drive where the CA is installed. You have the option of storing the database and the log on different drives to manage storage space. If this is something you want to do, type the new path and folder name in the **Certificate database** box or in the **Certificate log** box, or click **Browse** to select the new location.
Store configuration information in a shared folder	Select to store configuration information in a shared folder, and then type the path and folder name in the **Shared folder** box; or click **Browse** to select an existing folder. Members of the local Administrators security group are granted full control for the folder. Members of the Everyone security group are granted read permissions for the folder. The shared folder acts as a location where users can find information about certification authorities. This option is useful only if you are installing a stand-alone CA and do not have Active Directory.
Preserve existing certificate database	Select to preserve an existing certificate database. This option is available only when you are reusing a private key and the associated certificate from an existing CA configuration. You can use this option to restore a failed CA.

Table 16.9 CA Certificate Request Page (Subordinate CAs Only)

Option	Description
Send the request directly to a CA already on the network	Type the name of the parent CA, or click **Browse** to select the parent CA from a list of CAs. The certificate request is submitted to this CA, and the certificate is then processed and issued to the subordinate CA. If you make a request from a stand-alone CA, the CA is not certified automatically. An administrator must approve the certificate request before the CA can issue the certificate. You must later use the Certification Authority console to install the CA's certificate.
Save the request to a file	Select to save the request to a file, and then type the path and file name in the **Request file** box; or click **Browse** to select the file location. This option saves the certificate request to a request file that you can submit to an offline CA for processing. The CA is not certified automatically. You must later use the Certification Authority console to install the CA's certificate.

Configure Certification Authorities

You can use the Certification Authority console to configure CAs. This includes the following tasks:

- Installing the CA certificate when necessary.
- Configuring exit module settings.
- Configuring policy module settings.
- Scheduling certificate revocation list publication.
- Modifying security permissions and delegate control of CAs.
- Enabling optional Netscape-compatible Web-based revocation checking.

For more information about how to use the Certification Authority console to perform these tasks, see Certificate Services Help.

Installation of the Certification Authority Certificate

If you requested a certificate for a subordinate CA from an offline CA during the installation process, you must later obtain the CA certificate and install it to certify the CA. The CA does not run until the CA certificate is installed. You do not have to do this for root CAs or subordinate CAs that received the certificate from an enterprise CA during the installation process.

To obtain the CA certificate, use the **Submit a Saved Request** page of the Web Enrollment Support pages to submit the certificate request file that was created during the installation process. When the **Issued Certificate** page appears, click **Install this certification path** to install the certification path file for the CA. Then use the Certification Authority console to install the certification path file and certify the CA.

To use the Certification Authority console to install the CA certificate, right-click the CA node. Click **All Tasks,** and then click **Install CA Certificate**. The CA certificate is installed from the issuing parent CA, and then the CA service starts.

Configuration of Policy Module Settings

If the default policy module settings described in this section meet your needs, no further configuration is necessary. To configure the policy module setting by using the Certification Authority console, right-click the CA node and then click **Properties**. When the **CA Properties** dialog box appears, click **Policy Module** and then click **Configure**. When the **Properties** dialog box appears, modify the following settings as necessary.

Default Action (for Stand-alone Certification Authorities)

By default, the **Set the certificate request to pending** check box is selected and the request is held as pending until an administrator approves it for stand-alone CAs. Click **Automatically approve the certificate requests** to configure a stand-alone CA to issue each valid certificate request automatically. Note, however, that this is a major security risk and, thus, is not recommended. This option does not apply for enterprise CAs.

X.509 Extensions

You can click **Add** or **Remove** to modify the CRL distribution points that are listed in the **CRL Distribution Points** box. For example, to ensure that users have convenient access to CRLs, you can add a CRL distribution point for commonly used public folders and the URL for a page on your internal Web site. The CA writes these CRL distribution points into every certificate that it issues to support certificate revocation checking by applications such as Internet Explorer. You must also configure an exit module for the CA to publish its CRLs to any CRL distribution points you add. To disable a CRL distribution point that is listed in the **CRL Distribution Points** box, you can clear the check box next to it.

You can click **Add** or **Remove** to modify the certificate distribution points that are listed in **Authority Information Access**. For example, to ensure that users have convenient access to the certificate for a specified CA, you can add a certificate distribution point for frequently used public folders and the URL for a page on your internal Web site. The certificate for this CA is published to these certificate distribution points. In addition, the CA writes these certificate distribution points into every certificate that it issues. To disable a certificate distribution point that is listed in **Authority Information Access**, you can clear the check box next to a certificate distribution point. When you view the Certification Path dialog box for a certificate that is issued by this CA, and you select the CA's certificate in the path and click **View**, the system looks for the certificate in the order the certificate distribution points are listed in **Authority Information Access**.

Configuration of Exit Module Settings

If the default exit module settings described in this section meet your needs, no further configuration is necessary. To configure the active exit modules with the Certification Authority console, right-click the CA node, and then click **Properties**. When the **CA Properties** dialog box appears, click **Exit Module,** and then click **Add** or **Remove** to modify the active modules that are listed in **Active exit modules**. If you use custom exit modules that you have developed or exit modules provided by third-party vendors, you must install the exit module as an active module.

Install additional exit modules if you want to publish certificates and CRLs to different locations than those that are supported by the default enterprise and default stand-alone exit modules. For example, you might install a custom exit module to publish certificates to a Web page or to a third-party directory service.

No matter what exit modules are installed, certificates are not published unless the publication location is specified in the certificate request. The exit modules enable certificates to be published to the locations specified in certificate requests.

To configure an exit module's settings with the Certification Authority console, right-click the CA node, and then click **Properties**. When the *CA name* **Properties** dialog box appears, click **Exit Modules**. Select the module you want to configure, and then click **Configure**. When the **Properties** dialog box appears, modify the options that are described in Table 16.10.

Table 16.10 Certificate Publication

Option	Description
Allow certificate publication to Active Directory	By default, this option is selected for the default enterprise exit module. If you do not want to publish certificates or CRLs to Active Directory, clear the check box associated with this option. This option is not available for the default stand-alone exit module.
Allow certificate publication to the file system	By default, this option is selected for the default stand-alone exit module. If you do not want to publish certificates or CRLs to the file system, clear the check box associated with this option. By default, the check box for this option is cleared for the default enterprise exit module. If you want to enable certificates to be published to the file system, select the check box associated with this option.

For enterprise CAs, certificates are published to Active Directory as long as the default exit module is active and configured to publish certificates to Active Directory (the default setting). For stand-alone CAs, certificates are published to the local file system as long as the default exit module is active and configured to allow certificates to be published to the local file system (the default setting).

Scheduling Certificate Revocation List Publication

If the default CRL publication schedule meets your needs (a new CRL is published every week), no further configuration is necessary. The following are some examples of how you might modify the default CRL publication:

- Schedule daily rather than weekly publication of CRLs because you expect a high rate of certificate revocations or because you want to ensure greater protection of valuable information that is being protected by public key security functions.
- Schedule biweekly or monthly publication of CRLs because you expect a low rate of certificate revocations.
- Turn off automatic CRL publication for offline CAs, such as stand-alone root CAs or stand-alone intermediate CAs, and instead publish CRLs manually.

To change the CRL publication schedule with the Certification Authority console, right-click the **Revoked Certificates** node of the CA, and then click **Properties**. When the **Revoked Certificate Properties** dialog box appears, configure the CRL publication options that are described in Table 16.11.

Table 16.11 CRL Publishing Parameters

Option	Description
Publish Interval	Type the interval and select **Hours**, **Days**, **Weeks**, **Months**, or **Years**. For example, to schedule biweekly CRL publication, type **2** and select **Weeks**.
Next Publish	Displays the time that the next CRL is scheduled to be published.
Disable Scheduled Publishing	Select to turn off automatic CRL publishing for this CA.
View Current CRL	Select to view the most current CRL for this CA.

Configuration of Certificates to Be Issued

When an enterprise CA is installed, the default issuing policy is configured to issue the following certificate types: Administrator, Domain Controller, Computer, Basic EFS, EFS Recovery Agent, User, Subordinate Certification Authority, and Web Server. You can configure each CA's issuing policy to meet the needs of your organization.

Before you can issue other certificate types besides the default, you must use the Certification Authority console to add the certificate type to the issuing policy. You can also use the Certification Authority console to delete certificate types from an enterprise CA's issuing policy. For example, you might modify the certificate issuing policy for a root or an intermediate CA to issue only Subordinate Certification Authority certificates. You might configure an issuing CA by adding the Trust List Signing certificate type to the default issuing policy and by deleting the Subordinate Certification Authority certificate type from the default issuing policy. You might want to configure a CA to issue only the Enrollment Agent certificate. You can also configure an issuing CA so that it issues only the Smart Card Logon and Smart Card User certificates to support the deployment of smart cards.

To add a certificate type to issuing policy with the Certification Authority console, right-click the **Policy Settings** node of the CA. Click **New**, and then click **Certificate to Issue**. When the **Select Certificate Template** dialog box appears, select one or more of the listed certificate templates, and then click **OK**. The selected certificate templates are added to the issuing policy.

When you select the **Policy Settings** node of a CA, the certificate types that the CA can issue are displayed in the details pane of the console. To delete a certificate template from the issuing policy, select the certificate template and press the DELETE key; or right-click the certificate template, and then click **Delete**.

Permission to enroll for each certificate type is controlled by the ACLs for each certificate template, as described in "Modify the Default Security Permissions for Certificate Templates (Optional)" later in this chapter. You also can use the Certification Authority console to modify security settings for a CA to prevent some users or members of some security groups from enrolling for certificates from that CA.

Modification of Security for a Certification Authority

By default, members of the local Administrators and Authenticated Users security groups and members of the global Domain Admins and Enterprise Admins security groups are granted Enroll permissions, so they can request certificates from the CA. This means that by default all users in the domain can request certificates from the CA for all certificate types that they are authorized to receive. In addition, members of the local Administrators security group and members of the global Domain Admins and Enterprise Admins security groups are granted Manage permissions for the CA. If the default security for the CA meets your needs, no further configuration is necessary.

To configure new security settings for a CA by using the Certification Authority console, right-click the CA node, and then click **Properties**. When the **CA Properties** dialog box appears, click **Security,** and then modify the security settings as needed. Click **Add** or **Remove** to change the user accounts or security groups that are listed. When you select a security group or a user account from the list, the corresponding permissions appear in the **Permissions** box.

To change basic permissions, select a security group or a user account from the list, and then select or clear the appropriate check boxes next to the basic permissions in the **Permissions** box. You can select permissions check boxes in either the **Allow** or **Deny** columns. If you select a check box in the **Allow** column, the corresponding permissions are granted to the selected security group. If you select a check box in the **Deny** column, the corresponding permissions are denied to the selected security group.

To modify advanced permissions, click **Advanced**. When the **Permissions** dialog box appears, click **Add** or **Remove** to change the security groups or user accounts that are listed. Select a security group or a user account, and then click **View/Edit** to modify the advanced permissions.

Table 16.12 contains descriptions of the permissions you can configure for a CA. All of the permissions can be modified in the advanced **Permissions** dialog box.

Table 16.12 Permissions for Certificate Templates

Permission	Description
Manage (basic)	Determines which user accounts and security groups can manage the CA with the Certification Authority console or run command-line programs. By default, members of the local Administrators security group and members of the global Domain Admins and Enterprise Admins security groups are granted these permissions.
Enroll (basic)	Determines which user accounts and security groups can request certificates from the CA. By default, members of the local Administrators and Authenticated Users security groups and members of the global Domain Admins and Enterprise Admins security groups are granted these permissions.
Read (basic)	Determines which user accounts and security groups can read configuration information for the CA. By default, members of the local Administrators and Authenticated Users security groups and members of global Domain Admins and Enterprise Admins security groups are granted these permissions.
Write Configuration (advanced only)	Determines which user accounts and security groups can change configuration data for the CA. By default, these permissions are granted to all user accounts and security groups with Manage permissions.
Read Control (advanced only)	Determines which user accounts and security groups have read permission to view the security settings for the CA. By default, these permissions are granted to all user accounts and security groups with Read Configuration permissions.
Modify Permissions (advanced only)	Determines which user accounts and security groups can change permissions for CA security. By default, these permissions are granted to all user accounts and security groups with Manage permissions.
Modify Owner (advanced only)	Determines which user accounts and security groups can change the owner of the CA object. By default, these permissions are granted to all user accounts and security groups with Manage permissions.
Revoke Certificates (advanced only)	Determines which user accounts and security groups can revoke certificates. By default, these permissions are granted to all user accounts and security groups with Manage permissions.
Approve Certificates (advanced only)	Determines which user accounts and security groups can approve certificate requests for stand-alone CAs. By default, these permissions are granted to all user accounts and security groups with Manage permissions.
Read Database (advanced only)	Determines which user accounts and security groups can gain access to and read the information in the certificate database. By default, these permissions are granted to all user accounts and security groups with Manage permissions.

Enabling Netscape-compatible Web-based Revocation Checking

Netscape-compatible Web browsers support a proprietary online certificate revocation checking method that checks for revoked certificates at a location that is listed in an extension field of the certificate. To enable Netscape-compatible, Web-based revocation check extensions to be added to every certificate, run the following **Certutil** command from the command prompt on the CA server:

```
certutil -SetReg Policy\RevocationType +AspEnable
```

Then stop and start the Certification Authority service. Certificates that are issued by the certification authority after it is restarted contain the extension.

Modify the Default Security Permissions for Certificate Templates (Optional)

For enterprise CAs, Enroll permissions are controlled by ACLs for each certificate template. An enterprise CA grants certificate requests only for user accounts or computer accounts with Enroll permissions. The ACLs for certificate template are preconfigured to enable various security groups to enroll for certificate types.

By default, members of the Domain Admins security group for the domain where the CA is installed are granted Enroll permissions for all certificate types. Members of the Domain Users security group for the domain where the CA is installed are granted Enroll permissions for the following certificate types: Basic EFS, Authenticated Session, Exchange User, Exchange Signature Only, User, and User Signature Only. Members of the Enterprise Admins security group are granted Enroll permissions for all certificate types except for the Basic EFS, Authenticated Session, Exchange User, Exchange Signature Only, User, and User Signature Only.

If you want to enable other security groups to enroll for certificates, you must edit the ACLs for the certificate templates (for the domain where the CA is installed) to add the security group and assign Enroll permissions to them. In addition, if you want security groups in another domain to be able to enroll for certificates from an enterprise CA, you must add the other domain's security group to the ACLs of the certificate templates for the domain where the CA is installed.

You can use the Active Directory Sites and Services console (an MMC snap-in) to modify the ACLs for certificate templates. Before the Certificate Templates container appears, you must point to the Active Directory Sites and Services console and then click **View** and **Show Services Node**. For more information about how to use the Active Directory Sites and Services console, see Active Directory Help.

To show the Certificate Templates container, expand the Services container and the Public Key Services container, as shown in Figure 16.11.

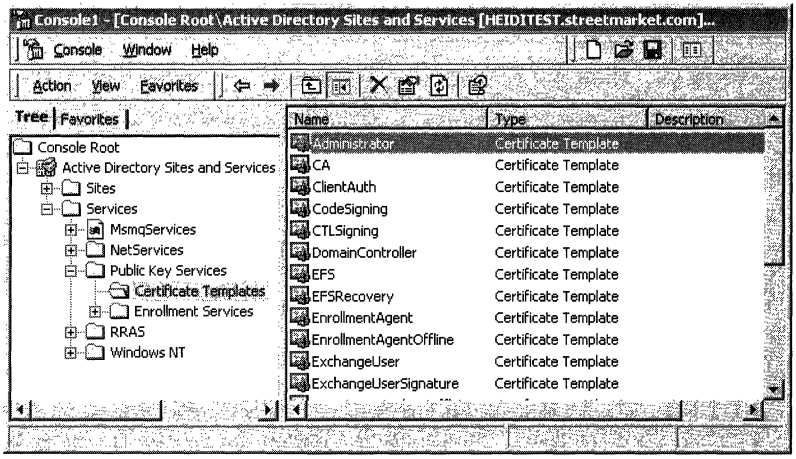


Figure 16.11 Certificate Templates Container

To edit ACLs for a certificate template, click **Certificate Templates**. Then, right-click the certificate template in the details pane, and click **Properties**. When the **Certification Authorities Properties** dialog box appears, click **Security** and modify the security permissions as needed. For more information about how to edit ACLs for certificate templates, see Certificate Services Help.

For example, to ensure that only a few trusted individuals can obtain an Enrollment Agent certificate, you might modify the ACLs for the Enrollment Agent certificate template to delete the default security groups and add a special security group with Enroll permissions. You might also modify the ACLs for the Code Signing certificate template so that only certain developers who are members of a special code signers security group can enroll for code singing certificates.

Note When you change the ACLs for certificate templates, the changes might take a few minutes to replicate to other domain controllers.

Install and Configure Support Systems or Applications

You must install any systems or applications that are required to support the public key infrastructure. Supporting systems and applications can include the following:

- Smart card readers at local computers.
- Secure mail and key management systems.
- Custom certificate enrollment and renewal applications.
- Training and support Web sites to educate users about certificate services and to provide customer support for users.
- Third-party public key infrastructure and certificate services.

Configure Public Key Group Policy

You can use the Group Policy console to configure Public Key Group Policy for sites, domains, and organizational units or local computer policy. Most features of the public key infrastructure and certificate services work without your having to configure Public Key Group Policy settings. However, you must configure Public Key Group Policy if you want to do any of the following:

- Use automatic enrollment for computer certificates.
- Add trusted root certificates for groups of computers.
- Create CTLs for computers and users.
- Designate EFS recovery agent accounts.

▶ **To add a Group Policy console to MMC**

1. Open MMC.
2. Click **Console**, and then click **Add/Remove Snap-in**, or press CTRL+M.

   The **Add/Remove Snap-in** dialog box appears.
3. Click **Add**.

   The **Add Standalone Snap-in** dialog box appears.
4. Select **Group Policy** from the list of snap-ins, and then click **Add**.

   The **Select Group Policy Object** dialog box appears, with Local Computer listed in the **Group Policy Object** box.

5. Click **Finish** to manage local computer policy.

   –Or–

   Click **Browse** to select another Group Policy (or to create and select a new Group Policy), and then click **Finish** to manage the selected Group Policy.

   The **Add Standalone Snap-in** dialog box appears. Click **Add** again to add multiple Group Policy snap-ins.

6. When you are finished adding snap-ins, on the **Add Standalone Snap-in** dialog box, click **Close**.

   The **Add/Remove Snap-in** dialog box appears and displays the snap-ins that are to be installed in MMC.

7. In the **Add/Remove Snap-in** dialog box, click **Close**.

For Group Policy for sites, domains, and organizational units, there is a Public Key Policy container for both computers and users. Figure 16.12 shows an example of the Public Key Policies container for computers in the Default Domain Policy. To display the Public Key Policies containers for computers, expand the **Computer Configuration** node, expand the **Security Settings** node, and then click **Public Key Policies**.

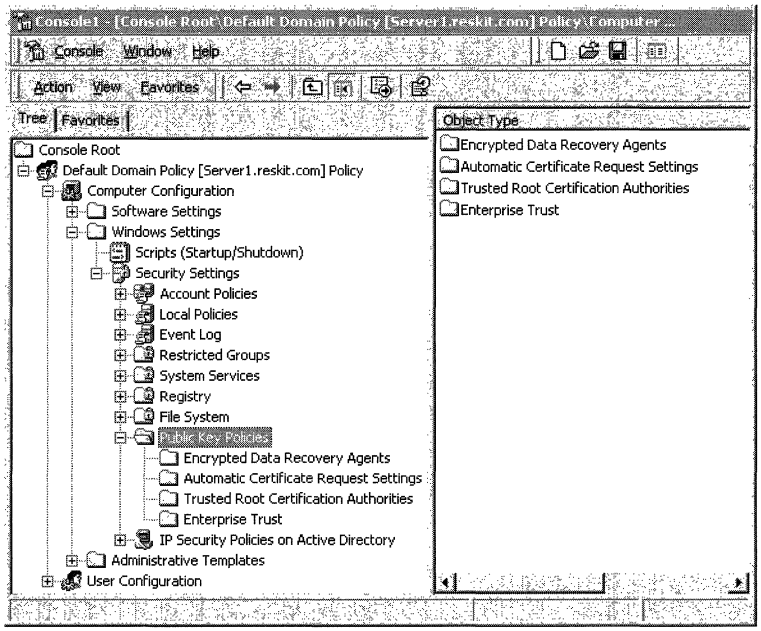


Figure 16.12 Public Key Policies for Computers

Figure 16.13 shows the Public Key Policies container for users in the Default Domain Policy. To display the Public Key Policies containers for users, expand the **Computer Configuration** node, expand the **Security Settings** node, and then click **Public Key Policies**.

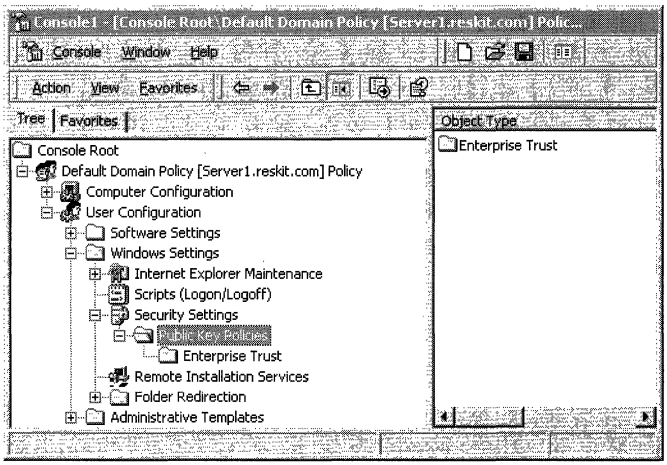


Figure 16.13 Public Key Policies for Users

The Public Key Policies containers are used for the following tasks:

- **Automatic Certificate Request Settings** for configuring autoenrollment for computer certificates.
- **Trusted Root Certification Authorities** for adding trusted root CA certificates to the Trusted Root Certification Authorities store.
- **Enterprise Trust** for configuring CTLs. (This is the only container that appears for users.)
- **Encrypted Data Recovery Agents** for configuring EFS recovery agents. (This is the only container that appears for Local Computer policy.)

For users, you can configure CTLs only. For Local Computer policy, you can configure EFS Recovery Agents Policy only. For more information about how to configure Public Key Group Policy, see Certificate Services Help and Group Policy Reference.

Note Changes to Group Policy do not take effect immediately. User and computer Group Policy is refreshed periodically (every 90 minutes, by default), when users log on, and when computers are started. You also can use the Secedit /refreshPolicy command-line option to refresh policy settings manually from the command prompt at each local computer.

Automatic Certificate Enrollment

You can specify automatic enrollment and renewal for computer certificates. When autoenrollment is configured, the specified certificate types are issued automatically to all computers within the scope of the Public Key Group Policy. Computer certificates that are issued by autoenrollment are renewed automatically from the issuing CA. Autoenrollment does not function unless at least one enterprise CA is online to process certificate requests.

To configure autoenrollment, in the **Public Key Policies** node, right-click the **Automatic Certificate Request Settings** node, and then click **New** and **Automatic Certificate Request**. When the Automatic Certificate Request wizard appears, configure autoenrollment by using the options that are described in Table 16.13.

Table 16.13 Automatic Certificate Request Wizard

Option	Description
Certificate Template page	Select a certificate template in the **Certificate templates** box, and then click **Next**. All computers that are within the scope of the autoenrollment policy with Enroll permissions for this certificate template are issued that certificate type the next time the computer restarts and logs on to the domain.
Certification Authority page	Select the check box next to one or more CAs that are listed in the **Certification authorities** box. If you select multiple CAs, certificate requests for autoenrollment are processed by the first CA that is available. After selecting the CAs, click **Next** and complete the wizard.

Root Certificate Trust

When you install an enterprise root CA or a stand-alone Root CA, the certificate of the CA is added automatically to the Trusted Root Certification Authorities Group Policy for the domain. You also can add certificates for other root CAs to Trusted Root Certification Authorities Group Policy. The root CA certificates that you add become trusted root CAs for computers within the scope of the Group Policy. For example, if you want to use a third-party CA as a root CA in a certification hierarchy, you must add the certificate for the third-party CA to the Trusted Root Certification Authorities Group Policy.

To add a certificate for the root CA to the Trusted Root Certification Authorities Group Policy, in the **Public Key Policies** node, right-click **Trusted Root Certification Authorities**, and then click **All Tasks** and **Import**. When the Certificate Import wizard appears, use the wizard to import a certificate file for the certificate of the root CA and add it to Group Policy. The certificate is added to the **Trusted Root Certification Authorities** store of all computers within the scope of Group Policy the next time it is refreshed on each computer.

Certificate Trust Lists

You can create CTLs to trust specific CAs and to restrict the uses of certificates issued by the CAs. For example, you might use a CTL to trust certificates that are issued by a commercial CA and restrict the permitted uses for those certificates. You might also use CTLs to control trust on an extranet for certificates that are issued by CAs that are managed by your business partners. You can configure CTLs for computers and for users.

Before administrators can create CTLs, they must have a valid trust list signing certificate, such as the Administrator certificate or the Trust List Signing certificate that are issued by enterprise CAs. The trust list signing private key for the administrator is used to sign the CTL for integrity. If the trust list signing certificate for an administrator is invalid, all CTLs that have been created and signed by that administrator also are invalid.

To create a CTL for computers or for users, in the **Public Key Policies** node (for the **Computer Configuration** node or for the **User Configuration** node), right-click the **Enterprise Trust** node, and then click **New** and **Certificate Trust List**. When the Certificate Trust wizard appears, configure the CTL by using the options that are described in Tables 16.14 through 16.18.

Table 16.14 Certificate Trust List Purpose Page

Option	Description
Type a prefix that identifies this CTL (optional)	Enter an option prefix for the CTL. This prefix is used to identify the CTL.
Valid duration (optional)	Specify an optional lifetime for the CTL. Enter the number of months in the **Months** box and the number of days in the **Valid duration (optional)** box. If you do not specify a lifetime, the CTL expires when the trust list signing certificate expires.
Designate Purposes	Select a check box next to one or more of the listed purposes in the **Designate purposes** box. The CTL establishes trust only for certificates that are valid for the selected purposes. A certificate might support all of the listed purposes, but you can restrict the purposes for which certificates are trusted.
Add Purpose	Click to add purposes to the **Designate purposes** box. When the **User Defined Purpose** dialog box appears, enter an object identifier for the new purpose in the **Object ID** text box.

Table 16.15 Certificates in the CTL Page

Option	Description
Current CTL Certificates	Displays the certificates of the root CAs that are to be trusted by this CTL. Certificates with certification paths to this root CA are trusted for all designated purposes specified by the CTL.
Add from Store	Adds a root certificate from the Trusted Root Certification Authorities store. When the **Select Certificate** dialog box appears, select all of the certificates that you want to add, and then click **OK**.
Add from File	Adds a root CA's certificate from a file.
Remove	Deletes the certificate that is selected in the **Current CTL Certificates** box.
View Certificate	Select this option to view the certificates that are selected in the **Current CTL Certificates** box.

Table 16.16 Signature Certificate Page

Option	Description
Use this certificate	Displays the trust list signing certificate for the private key that is to be used to sign the CTL.
Select from Store	Adds a trust list signing certificate from the Personal store for the administrator. When the **Select Certificate** dialog box appears, select the certificates you want to use, and then click **OK**.
Select from File	Adds the trust list signing certificate from a file.
View Certificate	Select this option to view the certificate listed in the **Use this certificate** box.

Table 16.17 Timestamping Page

Option	Description
Add a timestamp to the data	Adds a timestamp to the CTL. The timestamp is used to determine the valid lifetime of the CTL. If a timestamp is not used, the computer clock is used instead.
Timestamp service URL	Type the URL for a timestamp service that is to be used for the timestamp.

Table 16.18 Name and Description Page

Option	Description
Friendly Name	Type the optional name that is to appear in MMC when the CTL is displayed. To help you distinguish between CTLs, choose unique friendly names for all of the CTLs that you create.
Description	Type an optional description to describe this CTL. This description can let others know the purpose of the CTL.

EFS Recovery Agents

By default, the local Administrator users account for the first domain controller that is installed in the domain is the EFS recovery account for that domain. You can specify alternative recovery agents for EFS. Use the Group Policy console to designate alternative EFS recovery agents by adding the EFS Recovery Agent certificates into Public Key Group Policy, which means you must first issue EFS Recovery Agent certificates to designated recovery agent user accounts on local computers.

When you are configuring the EFS recovery settings, you have two choices: you can add recovery agent certificates that are published in Active Directory, or you can add recovery agent certificates from a file that is located on a disk or in a shared folder that is available from the computer where you are configuring Public Key settings. If you add recovery agent certificates from files, you must first export the appropriate certificates to the disk or shared folder that is to be used to add the files during the EFS recovery Group Policy configuration process.

To add an EFS recovery agent, in the **Public Key Policies** node, right-click **Encrypted Data Recovery Agents**, and then click **Add**. When the Add Recovery Agent wizard appears, add the appropriate recovery agent certificates by using the options described in Table 16.19.

Table 16.19 Add Recovery Agent Wizard

Option	Description
Recovery agents	Displays the certificates you choose for recovery agents.
Browse Directory	Browses Active Directory and adds a recovery agent certificate for a user account. Use this option when the certificate is published in Active Directory.
Browse Folders	Adds a recovery agent certificate from a file.

When you select **Encrypted Data Recovery Agents**, the EFS recovery agent certificates that are applied by Group Policy appear in the details pane of the console. These are the recovery agent certificates that are used by EFS within the scope of Group Policy. To delete an recovery agent certificate from the Group Policy settings, select the certificate. Next, either press DELETE, or right-click the certificate template, and then click **Delete**.

Install Web Enrollment Support on Another Computer (Optional)

You can install Windows 2000 Certificate Services with the Web Enrollment Support pages on the same server as the CA (the default configuration for the CA installation process). You also have the option of installing the Web Enrollment Support pages on another Windows 2000–based server. Installing the CA and the Web Enrollment Support package on different computers reduces the load that would otherwise be required for the CA computer. You might choose this option when the CA must support a high volume of certificate services traffic or when you are installing certificate services on less powerful computers.

The Web Enrollment Support pages are installed at the following location:

```
<Drive:>\WINNT\System32\CertSrv
```

where *<Drive:>* is the letter of the disk drive where the Web Enrollment Support pages are installed.

Folder CertSrv contains Web files (Active Server Page files, graphics files, and so forth) and two folders (CertEnroll and CertControl) that contain additional support files and ActiveX controls for the Web pages.

Trusting the Computer for Delegation

For enterprise CAs, the Web Enrollment Support pages work from a computer other than the CA computer only if the computer (where the Web Enrollment Support pages are installed) is trusted for delegation. You do not need to trust the other computer for delegation for the Web Enrollment Support pages to work with stand-alone CAs.

You can trust a computer for delegation by using the Active Directory Users and Computers console. Before you can install the Web Enrollment Support pages, you must be logged on to the computer as a member of the Domain Admins security group.

▶ **To trust a computer for delegation**

1. Expand the Active Directory Users and Computers node for the domain.
2. Select the container with the computer that you want to trust.

   The computers in the container appear in the details pane of the console.
3. Double-click the computer that you want to trust.

   The **Properties** dialog box for that computer appears.
4. In the **General** dialog box, click **Trust computer for delegation** to select the check box, and then click either **OK** or **Apply**.
5. Restart the computer so that the new delegation setting can take effect.

   The Web Enrollment Support pages will not work until after the computer has been restarted.

For more information about the Active Directory Users and Computers console, see Active Directory Help.

Installing the Web Enrollment Support Pages

You can use the Windows Components wizard to install the Web Enrollment Support pages on another computer other than where the CA is installed. Before you can install the Web Enrollment Support pages, you must be logged on to the computer as a member of the Domain Admins security group. You can install the Web Enrollment Support pages only on a Windows 2000–based server on which Internet Information Services is installed.

▶ **To install Web Enrollment Support pages on a computer other than where the CA is installed**

1. In Control Panel, click **Add/Remove Programs**.

   The **Add/Remove Programs** dialog box appears.

2. Click **Add/Remove Windows Components**.

   The Windows Components wizard appears.

3. In the **Windows Components** page, select the **Certificate Services** check box.
4. Click **Details**, and then clear the **Certificate Services** check box. Verify that the **Certificate Services Web Enrollment Support** check box is selected, and then click **OK**.
5. Click **Next**.

   The **Certificate Services Client Configuration** page appears.

6. Type the domain name of the server computer with the CA in the **Computer Name** box.

   –Or–

   Click **Browse** to locate and select the computer.

   The **CA Name** box displays the name of the CA that is running on the server you have selected. The Web Enrollment Support pages are installed to work with this CA.

7. Click **Next**, and complete the Windows Component wizard.

After the Web Enrollment Support pages are installed, test the Web pages to be sure that they work properly with the CA. For example, use the Web Enrollment Support pages to request a certificate or a CRL from the CA. You might also want to change the default security settings for the Web Enrollment Support pages.

Configure Security for Web Enrollment Support Pages (Optional)

The folders CertSrv, CertEnroll, and CertControl are added as virtual directories to the Default Web Site for Internet Information Services. For enterprise CAs, CertSrv and CertControl are configured for authenticated access with basic authentication and integrated Windows authentication enabled. Authenticated access authenticates users and grants access to Web resources on the basis of the users' Windows 2000 user accounts. Authenticated access is required because enterprise CAs must process certificate requests according to the information that is contained in the requestor's Windows 2000 user account. For stand-alone CAs, CertSrv and CertControl are configured for anonymous access to provide all users with access to the Enrollment Support pages. For enterprise CAs, anonymous authentication is turned off by default; otherwise the Web Enrollment Support pages do not work for enterprise CAs.

Integrated Windows authentication grants access to Web pages on the basis of the logon credentials of the users of Internet Explorer. Users are granted access to the Web pages when their logon credentials match a valid Windows 2000 user account. Integrated Windows authentication is not a part of the HTTP standard and is supported only by Microsoft® Internet Explorer version 2.0 or later and Internet Explorer 5. Integrated Windows authentication does not work across proxy servers or other firewall applications.

If integrated Windows authentication fails because of a firewall or another problem, the browser prompts the user to enter his or her user name and password for basic authentication. Users of third-party browsers also are prompted to enter their user names and passwords for basic authentication.

Basic authentication is a part of the HTTP version 1.0 standard, so most browsers support this authentication method. It grants access to Web pages after users have transmitted their Windows 2000 user names and passwords. However, a user must enter the correct user name and password before access is granted. User passwords are transmitted in plaintext, so they can be intercepted easily by someone who "sniffs" communications between the Web browser and the Web server. For enterprise Web Enrollment Support pages, basic authentication is enabled to ensure that all browsers have access to the Web pages. Because sending passwords as plaintext presents a security risk, you might want to turn off basic authentication or turn on digest authentication.

If you need to support only Internet Explorer, you can use the Internet Information Services console (an MMC snap-in) to configure security for CertSrv and CertControl, which turn basic authentication off and prevent passwords from being transmitted as plaintext. If you need to support other browsers, you can configure security for CertSrv and CertControl to require secure channels with the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols. With secure channels, passwords that are sent for basic authentication are encrypted. However, the performance of the Web Enrollment Support pages might be reduced because of the extra load of encryption that is required for secure channels.

Internet Information Services also supports digest authentication, which is a new feature of HTTP version 1.1. With digest authentication, passwords are sent in a secure manner as message digests (hashes) that can be deciphered only by the Windows 2000 Key Distribution Center (KDC) service for Kerberos authentication. If browsers support HTTP version 1.1 (recent version of browsers usually support HTTP version 1.1), you can use the Internet Information Services console (an MMC snap-in) to configure security for CertSrv and CertControl to turn off basic authentication and to turn on digest authentication. If you turn on both basic authentication and digest authentication, digest authentication is used if it is supported by the browser; otherwise basic authentication is used.

If you turn on integrated Windows authentication, basic authentication, and digest authentication, authentication is done in the following order of priority:

1. Integrated Windows authentication
2. Digest authentication
3. Basic authentication

The highest-ranked authentication method that is supported by browsers is used to authenticate users. If anonymous access is turned on, authenticated access is used only when NTFS file protection security has been configured to control access for Web site resources.

To ensure that the Web Enrollment Support pages work correctly with new security configurations, test the Web pages with all versions of the browsers that you intend to support.

For more information about security for Internet Information Services Web sites, see "Choosing Security Solutions That Use Public Key Technology" in this book. For information about how to use the Internet Information Services console to configure security and authenticated access for Web site resources, see Internet Information Services Help.

Integrate with Third-Party Certificate Services (Optional)

The Windows 2000 public key infrastructure is interoperable with various third-party certificate services that comply with the standards recommended by the Public Key Infrastructure X.509 (PKIX) working group of the Internet Engineering Task Force (IETF). However, interoperability between commercially available PKIX-compliant products is not guaranteed because the technology is still in an early stage of development. For more information about interoperability, see "Choosing Security Solutions That Use Public Key Technology" in this book.

In general, Windows 2000 Certificate Services provides many benefits that third-party CAs do not because Certificate Services are fully integrated with the Windows 2000 public key infrastructure and Active Directory. However, you can use third-party certificate services with Windows 2000 to deploy CAs and issue certificates for your organizations.

To work properly with Windows 2000 public key infrastructure, third-party CAs must support industry standard X.509 version 3 certificates and X.509 version 2 certificate revocation lists. X.509-compliant certificates from third-party CAs can be used for most public key–based Windows 2000 security solutions. However, third-party CAs can't be used for features that require enterprise CA integration with Active Directory. For example, third-party CAs can't be used to issue Smart Card Logon certificates or Smart Card User certificates for Windows 2000 domains or to autoenroll certificates for computers.

You can use compliant third-party CAs to form all or part of your certification trust chains. Third-party root CAs are not added automatically to Trusted Root Certification Authorities stores. You can configure Public Key Group Policies to add third-party root CAs to Trusted Root Certification Authorities stores and to create CTLs that trust third-party CAs.

To ensure that third-party certificate services work as intended with the Windows 2000 public key infrastructure, test third-party solutions thoroughly in labs and pilot programs. For more information about the capabilities of specific third-party solutions, contact the appropriate third-party vendors.

Ongoing Certificate Services Tasks

Ongoing tasks for Windows 2000 Certificate Services include the following activities:

- Using the Web Enrollment Support pages.
- Requesting certificates by using the Certificate Request wizard.
- Viewing information about certificates.
- Exporting certificates and private keys.
- Backing up and restoring certification authorities.
- Approving or denying certificate requests.
- Revoking certificates.
- Publishing certificate revocation lists.
- Renewing certification authorities.
- Recovering encrypted data.
- Using the Certificate Services command-line programs.

For more information about how to perform Certificate Services tasks, see Certificate Services Help.

Using the Web Enrollment Support Pages

To use the Web Enrollment Support pages, open the following URL with your Web browser:

http://<*servername*>/certsrv

where <*Servername*> is the name of the server computer where the Web Enrollment Support pages are installed.

When the **Welcome** page appears in your browser window, choose one of the options described in Table 16.20.

Table 16.20 Welcome Page Options

Option	Description
Retrieve the CA certificate or certificate revocation list	Retrieves the CA's certificate or the most current CRL. When you click **Next**, the **Retrieve The CA Certificate or Certificate Revocation List** page appears. You can also use this page to establish trust for the CA on the local computer by installing the certification path for the CA's certificate in the certificate store of the local computer.
Request a certificate	Requests a basic certificate or to submit a certificate request by using advanced options, as described later in this chapter. When you click **Next**, the **Choose Request Type** page appears.
Check on a pending certificate	Checks the status of a pending certificate request and installs the certificate after the request has been approved. When you click **Next**, the **Check On a Pending Certificate Request** page appears. Use this option for certificate requests that are sent to stand-alone CAs. If you don't check the status of pending certificates within 10 days, the pending certificates are not issued and you must request the certificate again.

After you have selected an option, click **Next**. Different Web pages appear for each option.

Choosing the Type of Certificate to Request

You can use the **Choose Certificate Type** page to request user certificates or to submit a certificate request by using the advanced options that are described in Table 16.21.

Table 16.21 Choose Certificate Type Page

Option	Description
User certificate request	Select one of the certificate types listed. For enterprise CAs, you can select **User Certificate**. For stand-alone CAs, you can select either **E-Mail Protection Certificate** or **Web Browser Certificate**. The Web Browser and E-Mail Protection certificates for stand-alone CAs together provide most of the functionality of the User certificate type for enterprise CAs (except for EFS functionality). When you click **Next**, the **User Certificate - Identifying Information** page appears.
Advanced request	Makes a certificate request by using advanced options. When you click **Next**, the **Advanced Certificate Requests** page appears.

After you select an option, click **Next**, and then complete the certificate request process by using the Web pages that appear.

When you request a certificate from an enterprise CA, the CA uses the certificate template and user account information in Active Directory to verify your user account and determine whether to approve or deny the certificate request. However, by default, stand-alone CAs store certificate requests as "pending" until a CA administrator approves or denies the request. Use the stand-alone Web pages to submit a request for a certificate from the stand-alone CA, and then return later to the Web pages to check the status of the pending request. When the request has been approved, you are prompted to install the issued certificate. You can configure stand-alone CAs to grant certificate requests immediately, but this is a significant security risk and is not recommended.

Submitting User Certificate Requests

You can use the **User Certificate - Identifying Information** page to request user certificates by using the options that are described in Table 16.22.

Table 16.22 User Certificate - Identifying Information Page

Option	Description
Identifying Information (stand-alone CAs only)	Enter identification information that is to appear in the certificate including Name, E-Mail, Company, Department, City, State, and Country/region. Enterprise CAs obtain this information from Active Directory. This information is included in the Subject field of the certificate when it is issued.
More options	Displays advanced options for choosing the CSP or for choosing strong private key protection.
Enable strong private key protection	Provides strong private key protection. When this option is selected, the system prompts the user for permission before it performs cryptographic operations with the user's private key.
CSP	The default CSP is the Microsoft Base Cryptographic Provider or the Microsoft Enhanced Cryptographic Provider, depending on whether the Windows 2000–based client that requests the certificate is exportable or not. You have the option of choosing a CSP from the selection list, which is used for the private key. The CSP you choose must support the type of certificate to be issued. For example, a smart card CSP cannot support a Basic EFS certificate.

After you configure options in the **User Certificate - Identifying Information** page, click **Next**. For enterprise CAs, requests are submitted to the CA and approved immediately. For a stand-alone CA, certificate requests are held as "pending" until an administrator approves the certificate request. You must return to the **Welcome** page within 10 days and select the **Check on a pending certificate** option to determine whether a pending certificate request has been approved. When the certificate is issued, the **Issued Certificate** page appears so that you can install the certificate.

Submitting Advanced Certificate Requests

You can use the **Advanced Certificate Requests** page to request certificates by using advanced options that are described in Table 16.23.

Table 16.23 Advanced Certificate Requests Form

Option	Description
Submit advanced requests to this CA using a form.	Submits an advanced certificate request by using a Web form. When you click **Next**, the **Advanced Certificate Request** form appears.
Submit a certificate request using a base 64 encoded PKCS #10 file or a renewal request using a base 64 encoded PKCS #7 file	Submits a certificate request by using a certificate request or a certificate renewal file. When you click **Next**, the **Submit a Saved Request** page appears.
Request a certificate for a smart card on behalf of another user by using the Smart Card Enrollment Station	Requests smart card certificates for other users. When you click **Next**, the **Smart Card Enrollment Station** page appears.

After you select an option, click **Next**, and then use the Web pages that appear to submit the advanced request.

Advanced Certificate Request Form

You can use the **Advanced Certificate Request** form to submit certificate requests by using the options that are described in Table 16.24.

Table 16.24 Advanced Certificate Request Page

Option	Description
Identifying Information(stand-alone CAs only)	Type identification information that is to appear in the certificate, including Name, E-mail, Company, Department, City, State, and Country/region. Enterprise CAs obtain this information from Active Directory. This information is included in the Subject field of the certificate when it is issued.
Intended Purpose(stand-alone CAs only)	Choose the intended purpose of the certificate that is to be requested from the selection.
Certificate Template(enterprise CAs only)	Choose the certificate template from the selection list that is to be used by the enterprise CA to process the certificate request and issue the certificate.

(continued)

Table 16.24 Advanced Certificate Request Page *(continued)*

Option	Description
CSP	The default CSP is the Microsoft Base Cryptographic Provider or the Microsoft Enhanced Cryptographic Provider, depending on whether the Windows 2000 client that requests the certificate is exportable or not. You have the option of choosing a CSP from the selection list, which is to be used for the private key. The CSP you choose must support the type of certificate that is to be issued. For example, a smart card CSP cannot support a Basic EFS certificate.
Key Usage	Select the basic purpose of the certificate that is to be issued. The options are **Exchange**, **Signature**, or **Both**. If you click **Exchange**, the key can be used for symmetric key exchange only. If you click **Signature**, the key can be used for digital signing only. The default is **Both**, so the key can be used for both purposes.
Key Size	For a Key Usage of Exchange or Both, you can enter a key length from 384 bits to 1,024 bits. The minimum recommended key length is 512 bits. For a Key Usage of Signature, you can enter a key length from 384 bits to 16,384 bits. Key generation for very large signing keys can take a considerable amount of time.
Create new key set	This is selected by default, so a new private key and public key set are created for the issued certificate. Click **Select the container name** to enter a container name for the private key in the **Container name** box.
Use existing key set	Uses an existing private key and public key set. You also can enter the name of the key container in the **Container name** box. You must not reuse private keys if the maximum safe lifetime of the key might be exceeded.
Enable strong private key protection	Provides strong private key protection. When this option is selected, the system prompts the user for permission before it performs cryptographic operations with the user's private key.
Mark keys as exportable	Enables the private key to be exported. Private keys that are used for digital signing (signatures) cannot be enabled for export.

(continued)

Table 16.24 Advanced Certificate Request Page *(continued)*

Option	Description
Use local machine store	Stores a certificate that is to be issued in the HKEY_LOCAL_MACHINE subtree of the system registry for the local computer. You must be an administrator to use this option. The default certificate storage location for user certificates is the Personal certificate store for the user. Select this option to request and install computer certificates for the local computer.
Hash Algorithm	Select the message digest (hash) algorithm that is used to sign the certificate request and ensure its integrity. The default algorithm is SHA-1. You can choose another algorithm from the selection list, which is used to sign the certificate request.
Save request to a PKCS #10 file	Saves the certificate request to a file rather than submitting the request to the CA. You must also type a file name in the **File name** box. You can submit the request file to a CA later.
Attributes	Enter additional attributes for the requested certificate in the **Attributes** box. For more information about certificate attributes and the syntax to use, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

For enterprise CAs, the Advanced Certificate Requests form enables you to request all certificate types that are supported by the enterprise CA's certificate issuing policy. Enterprise CAs use certificate templates and information in the logged-on user's user account to process and issue the requested certificates. For offline certificate templates, you must type identifying information in the following fields of the Web form:

- Name
- E-mail
- Company
- Department
- City
- State
- Country/region

This information is included in the Subject field of the certificate when it is issued. For online certificate templates, this information is obtained from the Windows 2000 user account of the logged on user.

For stand-alone CAs, you also can choose the following types of certificates in **Intended Purpose**:

- Secure mail
- Client authentication
- Server authentication
- Code signing
- IP security authentication
- Timestamp signing
- Other

Certificate uses are based on the object identifier contained in the Extended Key Usage field of X.509 version 3 certificates. You can optionally choose Other types of certificates from the selection list and enter the object identifier in the **Usage OID** box. Some object identifiers for certificate types that are not included in the **Intended Purpose** selection list include the following:

- EFS local file encryption (1.3.6.1.4.1.311.10.3.4)
- EFS recovery agent (1.3.6.1.4.1.311.10.3.4.1)
- Certificate Trust List Signing (1.3.6.1.4.1.311.10.3.1)
- Enrollment Agent (1.3.6.1.4.1.311.20.2.1)

For more information about the available types of certificates and their object identifiers, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

After you have configured the options in the **Advanced Certificate Request** page, click **Next**. For enterprise CAs, requests are submitted to the CA and approved immediately. For a stand-alone CA, certificate requests are held as "pending" until an administrator approves the certificate request. You must return to the **Welcome** page within 10 days and select the **Check on a pending certificate** option to determine whether a request has been approved. When the certificate is issued, the **Issued Certificate** page appears so that you can install the certificate.

Submit a Saved Request Page

You can use the **Submit a Saved Request** page to submit a request file to the CA by using the options that are described in Table 16.25.

Table 16.25 Submit a Saved Request Page

Option	Description
Saved Request	Paste the certificate request into the **Saved Request** box, or click **Browse** to locate and select a request file that is to be inserted in **Saved Request**. Requests can be either base 64 encoded PKCS #10 certificate requests or PKCS #7 renewal requests.
Certificate Template(enterprise CAs only)	Choose the certificate template from the selection list that is to be used by enterprise CAs to process the certificate request and issue the certificate.
Attributes	Enter additional attributes for the requested certificate in the **Attributes** box. For more information about certificate attributes and the syntax you must use, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

After you have configured the options in the **Submit a Saved Request** page, click **Next**. For enterprise CAs, requests are submitted to the CA and approved immediately. For a stand-alone CA, certificate requests are held as "pending" until an administrator approves the certificate request. You must return to the **Welcome** page within 10 days and select the **Check on a pending certificate** option to determine whether a pending certificate request has been approved. When the certificate is issued, the **Issued Certificate** page appears so that you can install the certificate.

Smart Card Enrollment Station Page

To enable central and secure administration of your smart card program, the Web Enrollment Support pages include the **Smart Card Enrollment Station** page so that trusted administrators or security personnel can enroll for smart card certificates on the behalf of other users. Things to keep in mind for using this station include the following:

- Only administrators with Enrollment Agent certificates can use the Smart Card Enrollment Station page. Requests for smart card certificates must be signed with the administrator's Enrollment Agent certificate.
- By default, only members of the Domain Admins and Enterprise Admins security groups can request and obtain Enrollment Agent certificates.
- By default, only members of the Domain Admins and Enterprise Admins security groups can request and obtain Smart Card Logon and Smart Card User certificates.
- Issued certificates are stored on the user's smart card, which is inserted into the smart card reader at the smart card administrator's workstation.
- Certificates for logging on with a smart card must be mapped to the user's network account by an enterprise CA. Therefore, you cannot use stand-alone CAs to enroll users for certificates that are used for the smart card logon process. However, you can use stand-alone CAs to enroll users for client authentication and secure mail certificates, which are stored on smart cards.
- You can issue any type of certificate to a smart card to provide extra security for private keys or to enable users to easily transport certificates. However, not all applications or services support smart cards. For example, you can store a Basic EFS certificate on a smart card, but EFS does not support smart cards.

You can modify the Enroll permissions for the Enrollment Agent, Smart Card Logon, and Smart Card User certificate templates to allow other users and security groups to enroll for these certificates. For example, you can modify the ACLs for the smart card certificate templates to grant the Domain Users security group (all user accounts in the domain) Enroll permissions so that they can request or renew their own smart card certificates. However, this weakens the overall security provided by smart cards and is not recommended.

In addition, when someone has an Enrollment Agent certificate, they can enroll for a certificate and generate a smart card certificate on behalf of anyone in the organization. The resulting smart card might then be used to log on to the network and impersonate the real user. The unauthorized impersonator can have all the rights and permissions that are granted to the authorized user. For this reason, it is strongly recommended that your organization maintain strict security policies over who can be issued this certificate type.

For example, to minimize the risk of Enrollment Agent certificate misuse, you can configure one dedicated subordinate CA with restrictive administrative controls to issue Enrollment Agent certificates for your organization. After the initial Enrollment Agent certificates have been issued, the administrator of the CA can disable the issuance of Enrollment Agent certificates until they are needed again. By restricting which administrators can operate the CA service on the subordinate CA, the service can be kept online for the generation and distribution of CRLs, if necessary. Other CAs in the hierarchy can conceivably still issue Enrollment Agent certificates if their issuing policy settings are changed, but you can determine whether inappropriate Enrollment Agent certificates are issued by regularly checking the Issued Certificates log for each CA.

You also can change the ACLs on the Enrollment Agent, Smart Card Logon, and Smart Card User certificate templates to grant Enroll permissions to a small group of trusted administrators only. For example, you might allow only members of a smart card security officers security group to have Enroll permissions for the Enrollment Agent, Smart Card Logon, and Smart Card User certificate templates.

Tip In Windows 2000, only one certificate and one private key can be stored on a smart card. Windows 2000 Certificate Services includes the Smart Card User certificate template, which supports network logon authentication, client authentication for Web communications, and secure mail. To provide maximum functionality for smart cards, you can issue this certificate to smart card users rather than the Smart Card Logon certificate, which is valid only for network logon authentication.

You can use the **Smart Card Enrollment Station** page to enroll users for smart card certificates by using the options described in Table 16.26.

Table 16.26 Smart Card Enrollment Station Options

Option	Description
Identifying Information(stand-alone CAs only)	Type identification information that is to appear in the certificate, including Name, E-mail, Company, Department, City, State, and Country/region. Enterprise CAs obtain this information from Active Directory.
Intended Purpose(stand-alone CAs only)	From the selection list, choose the intended purpose of the certificate that is to be requested.
Certificate Template(enterprise CAs only)	From the selection list, choose the certificate template that is to be used by the enterprise CA to process the certificate request and issue the certificate. For example, choose either Smart Card Logon or Smart Card User.

(continued)

Table 16.26 Smart Card Enrollment Station Options *(continued)*

Option	Description
Cryptographic Service Provider	Choose the smart card CSP that is appropriate for the user's smart card. For example, choose the Gemplus GemSAFE Card CSP for Gemplus smart cards or the Schlumberger Cryptographic Service Provider for Schlumberger smart cards.
Administrator Signing Certificate	From the selection list, click **Select Certificate** to choose your Enrollment Agent certificate. You cannot use an Enrollment Agent certificate that belongs to someone else.
User to Enroll	Click **Select User** to select a user account from Active Directory for which you are enrolling the smart card certificate.

After you have configured all of the options, insert the user's smart card in the smart card reader. Then click **Enroll** to request the smart card certificate. The PIN confirmation process and dialog boxes that appear differ depending on the specific smart card CSP that is used.

For the Schlumberger Cryptographic Service Provider, the **Smart Card PIN Confirmation** dialog box appears. For the Gemplus GemSAFE Card CSP, an untitled dialog box appears. Use the dialog box to confirm the PIN for the smart card. You also have the option of changing the PIN. Table 16.27 describes the options for the Schlumberger CSP dialog box. Table 16.28 describes the Gemplus CSP dialog box.

Table 16.27 Smart Card PIN Confirmation Dialog Box (Schlumberger CSP)

Option	Description
Please enter your PIN	Type the correct PIN for the smart card that is inserted in the smart card reader. Click **OK** to submit the PIN for confirmation by the CSP.
Change PIN after Confirmation	Select this check box to change the PIN. When you click **OK**, the CSP confirms the PIN you typed in the **Please enter your PIN** box, and then displays the **Change PIN on Smartcard** dialog box. Type the new PIN in the **New PIN** box; type it again in the **Confirm New PIN** box. Click **OK** to change the PIN.

Table 16.28 Untitled Dialog Box (Gemplus CSP)

Option	Description
Unlabeled box	Type the correct PIN for the smart card that is inserted in the smart card reader. Click **Change** to change the PIN, or click **OK** to submit the PIN for confirmation by the CSP.
Change	Changes the PIN. The CSP confirms the PIN you typed in the unlabeled box and displays the **Please Enter New PIN Code** dialog box. Type the new PIN in the top (unlabeled) box, and then type it again in the bottom (unlabeled) box. Click **OK** to change the PIN.

It is recommended that you assign a unique PIN for each smart card that is issued. Your policies for PINs can be much less restrictive than your policies for network passwords. In general, network passwords require long and complex composition, and it is recommended that users change them often. Users are more likely to write down their complex passwords because they are hard to remember. However, PINs can be changed infrequently and can be relatively short and easy to remember so that users are less likely to write them down. PINs are managed by the smart card CSP and can be changed only when smart card certificates are issued or renewed.

After the smart card PIN is confirmed or successfully changed, the smart card CSP generates the public key and private key set, and then stores the private key and the certificate on the user's smart card. When the smart card certificate is issued, the **Status** section of the **Smart Card Enrollment Station** page appears with a message that explains that the smart card is ready. Click **View Certificate** to display the certificate and verify that the user account information and the certificate type are correct. Click **New User** to submit another certificate request by using the Smart Card Enrollment Station page.

Installing the Certificate After It Is Issued

For enterprise CAs, the certificate is approved and issued after a short time unless the request is denied. For stand-alone CAs, certificate requests are held as "pending" until an administrator approves the request and the CA issues the certificate.

When certificates (except smart card certificates) are issued by CAs, the **Issued Certificate** page appears. Click **Install this certificate** to install the certificate in the Personal certificate store for the logged-on user. If you are requesting a certificate for a computer, you must select the **Use local machine store** option on the **Advanced Certificate Request Form** page to install the certificate in the Personal store for the computer rather than in the Personal store for the logged-on user.

For subordinate CA certificates, click **Install this certification path** to install the certification path file for the CA. You then can use the Certification Authority console to install the certification path file and certify the CA.

Requesting Certificates with the Certificate Request Wizard

You can request certificates for Windows 2000–based computers by using the Certificates console. When you right-click the Personal store for a user or for a computer and then click **All Tasks** and **Request New Certificate**, the Certificate Request wizard appears. You can use the Certificate Request wizard to request a certificate from an active enterprise CA. The Certificate Request wizard lists all certificate types that the user or computer is eligible to obtain. You can select a certificate type and submit it to any active CA that is configured to issue that type. If no CA is available to process certificate requests or the user or computer is not eligible for any certificate types, the Certificate Request wizard does not appear.

You have the option of selecting the **Advanced** check box on the first page of the Certificate Request wizard to choose advanced options. The advanced options enable you to select the CSP that is used with the certificate (as long as the CSP supports the cryptographic operations required for that certificate type). For user certificates only, users can also select strong private key protection as an advanced option. You also have the option of selecting the **Enable strong private key** check box, which means that the system prompts the user for permission before conducting cryptographic operations with the user's private key. Strong private key protection is available only for user certificates, not for computer certificates.

When you are choosing strong private key security, you can select either **Medium security** or **High security**. For **Medium security**, the system prompts the user for permission before using the private key, but it does not require a password. For **High security**, the user also must specify a password, which is used to protect the private key.

When you are requesting EFS user certificates, you can choose **Enable strong private key**; but EFS does not support a user interface, so users are never prompted for EFS user operations. However, strong private key protection works for recovery agent certificates. When you are requesting recovery agent certificates, consider choosing **Enable strong private key** and **High security** to provide an additional level of security for EFS recovery operations. Likewise, consider choosing **High security** to password protect the private keys for smart card enrollment agent certificates, code signing certificates, and trust list signing certificates, which might be misused to cause significant damage to your network resources.

When the CA issues the requested certificate, you can choose to view the certificate or install the certificate in the Personal store for the selected user or computer. Users also can request certificates from CAs with the Web Enrollment Support pages.

Viewing Information About Certificates

When you double-click a certificate, or right-click the certificate, and then click **Open**, the **Certificate** dialog box appears, in which you can view the following:

- General information
- Details information
- Certification path information

Figure 16.14 is an example of the **General** dialog box.

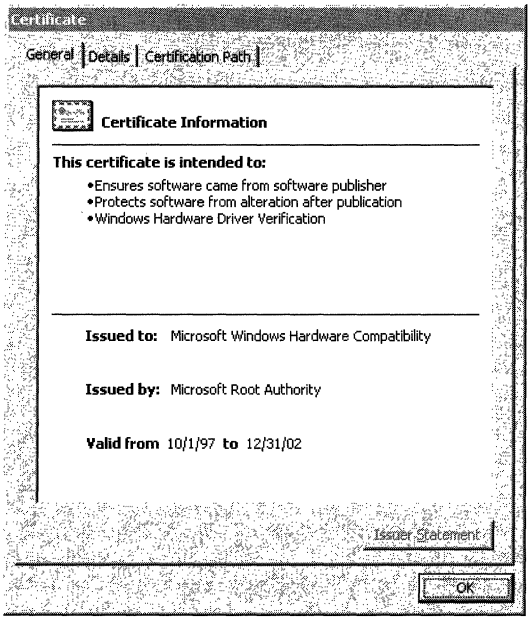


Figure 16.14 Certificate General Dialog Box

The **Certificate General** dialog box lists general information about the certificate, including the intended purposes of the certificate, the issuing CA, and the validity dates. If there is a problem with the certificate, a warning message with additional information appears in the dialog box. **Issuer Statement** is grayed out because the issuing CA does not provide a statement. However, if the issuing CA provides a statement, you can click **Issuer Statement** to obtain additional information about the certificate from the issuing CAs Web site.

Figure 16.15 is an example of a **Certificate Details** dialog box.

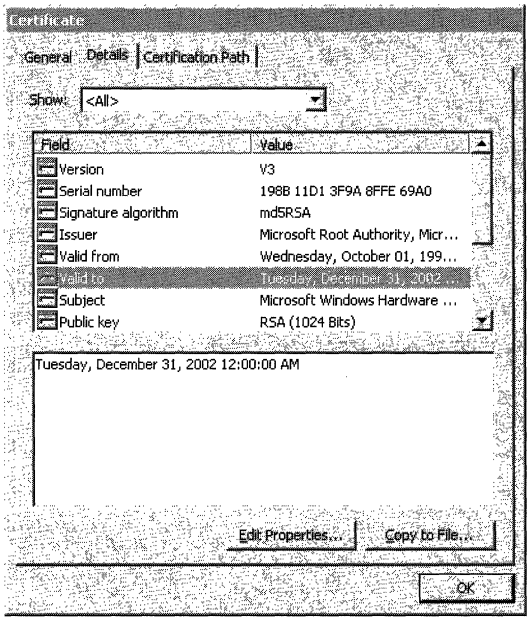


Figure 16.15 Certificate Details Dialog Box

The **Certificate Details** dialog box shows the information that is contained in the certificate, such as **Version**, **Valid to**, and **Friendly Name** (not shown). When you select an item in the **Field** column, the contents of the **Value** column for that item appear in the display box at the bottom of the dialog box. For example, in Figure 16.15, **Valid to** is selected and "Tuesday, December 31, 2002 12:00:00 AM" appears in the display box.

You can click **Edit Properties** to edit the **Friendly Name** and **Description** for the certificate, which appear in the Certificates console. You can also click **Edit Properties** to restrict the purposes for which the certificate can be used. By default, all of the purposes that are listed in the certificate are enabled. However, you can choose to disable all purposes (thus making the certificate invalid), or you can choose to trust the certificate for specific purposes only. For example, if a certificate is valid for code signing, secure mail, and secure Web communications, you can choose to trust it for secure mail only.

You can click **Copy to File** to export the certificate. If key export is enabled for the certificate, you also have the option of exporting the private key.

Figure 16.16 shows an example of a **Certification Path** dialog box.

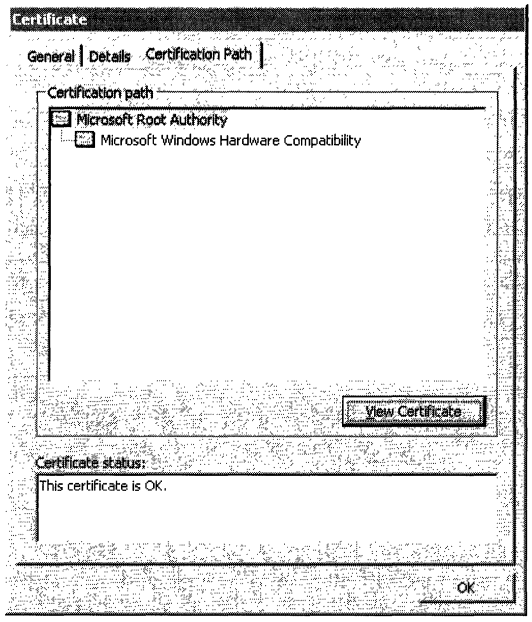


Figure 16.16 Certificate Certification Path Dialog Box

The **Certificate Certification Path** dialog box provides a graphic representation of the certification path and lists the trust status of the certificate. The **Certificate status** box describes the status of the certificate. If there is a problem with the certificate or the path, a warning appears in the **Certificate Certification Path** dialog box with information that explains the problem. Common problems include the parent certificate not being in the Trusted Root Certification Authorities store or a certificate in the **Certification path** box that does not validate properly. You can select a certificate in the **Certification path** box and click **View Certificate** to view information about the selected certificate.

Exporting Certificates and Private Keys

When you right-click a certificate and then click **All Tasks** and **Import**, or when you click **Copy to File** in the **Certificate Details** dialog box, the Certificate Export wizard appears. You can use the Certificate Export wizard to export the selected certificate to a file and to optionally export the private key if enabled to do so. If the private key is exported, the key is stored in a password protected encrypted file format. You must specify a password that is then used to lock and unlock the exported key. You cannot access the exported private key again without the password.

Of course, because password protection provides relatively weak protection, someone who has access to an exported private key can launch a brute force or dictionary attack and decode the encryption scheme in a relatively short period of time. Therefore, to avoid the compromise of private keys, you must carefully control the export of private keys and provide adequate security for any medium that contains exported private keys.

Important Private keys that are used for digital signing must never be exported or stored in a file or an archive. Someone other than the legitimate key owner might be able to gain access to the duplicate and impersonate the owner. If a copy of a signing key exists, the authentication, integrity, or nonrepudiation provided by the key is compromised. Therefore, Windows 2000 does not permit the export of private keys that are used for signing.

For standard Windows 2000 Certificate Services certificates, private key export is enabled only for EFS user certificates and recovery agent certificates. Key export is enabled for EFS certificates, so that you can maintain a key recovery archive. The export of private keys is enabled by an attribute that is included in the certificate when it is created. When you use the Advanced Certificate Request Web pages, you have the option of enabling private key export for custom certificates that you issue for key exchange purposes only. You cannot use the Advanced Certificate Request Web pages to enable private key export for custom certificates that are used for the purpose of both key exchange and signatures.

You must enable the export of private keys only for keys that are used to store long-term (persistent) data, such as encrypted files on your hard disk. For example, if you issue secure mail certificates that have the purpose of confidential mail only (not signing mail), you might want to enable key export so that you can archive the keys securely for recovery purposes. If so, you also need to issue secure mail certificates that are used for signing mail only and that have private key export disabled.

Backing Up and Restoring Certification Authorities

It is recommended that CAs be backed up regularly so that the CA can be restored if there is a server disaster such as a hard disk failure. If a hard disk fails, you can lose data that has changed since the last back up, such as the following information:

- Changes to the configuration of Certificate Services
- Record of certificates issued
- Record of certificate requests
- Certificate request queue
- Record of certificates revoked

To minimize the effect of a server disaster, you can use Windows 2000 Backup to back up and restore the CA as part of your server backup and restore program. You also can back up and restore the Certificate Services configuration data, the private key, the certificate, and the certificate database for the CA by using the Certification Authority console.

Windows 2000 Backup and Restore

You can use Windows 2000 Backup to schedule and perform periodic backups for the server where the CA is installed. If the server fails (for example, as a result of a hard disk failure), you can use Windows 2000 Backup to restore the server and its services by using the most current backup set.

In Windows 2000 Backup, schedule and perform the following types of backups:

- Normal (full) backups. Backs up the entire server file system and the system state.
- Differential backups. Backs up all changes to the server file system and the system state since the last normal backup.
- Incremental backups. Backs up all changes to the server file system and the system state since the last back up.

Although you have the option of backing up the file system without the system state, back up files with the system state to ensure full recovery of the server. Because Certificate Services depends on the Web Enrollment Support pages, you must also make sure to backup Internet Information Services at the same time.

Windows 2000 Backup supports a wide range of storage devices, such as hard disks, tape drives, removable disks, recordable CDs, or an entire library of disks or tapes organized into a media pool and controlled by a robotic changer. For more information about how to use Windows 2000 Backup, see Windows 2000 Server Help.

Certification Authority Console Backup and Restore

You can use the Certification Authority Backup wizard and the Certification Authority Restore wizard (available from the Certification Authority console) to back up and restore the following types of CA data:

- Private key and certificate
- Certificate database

You can back up all data or only selected data for the CA. For example, you can backup only the private key and certificate, or you can back up only the certificate database. You also can choose to perform a normal (full) backup or an incremental back up. You can back up CA data to an empty folder on any NTFS, FAT, or FAT32 storage device that is supported by Windows 2000.

If a server disaster occurs, you can restore the CA from the most current backup set. You must first restore the last normal backup, and then restore each incremental backup in the order in which they were backed up.

When you back up the CA's private key, you must provide a password. The private key is stored in a password-protected, encrypted format for protection and confidentiality of the key. You must supply the original password before you are permitted to restore the private key. For more information about how to use the Certification Authority console to backup and restore CAs, see Certificate Services Help.

Backup Strategies

It is recommended that you schedule and perform frequent backups to ensure that the CA can be restored with the minimum disruption to Certificate Services. Typical backup strategies usually include the following combinations of periodic normal (full), differential, and incremental backups.

Daily Normal Backups Normal backups are the most complete and easiest to restore. However, normal backups take the most time, consume the most storage space, and place the greatest load on servers and the network.

Weekly Normal and Daily Differential Backups Daily differential backups take less time, consume less storage space, and place less load on servers and the network than daily normal backups do. However, restoring the data takes longer because you must restore the last normal backup and then the last differential backup.

Weekly Normal and Daily Incremental Backups Daily incremental backups take less time, consume less storage space, and place less load on servers and the network than daily differential backups do. However, restoring the data takes longer because you must restore the last normal backup and each incremental backup in order since the last normal backup.

In addition, you can alternate normal backups with differential or incremental backups at any interval that meets your needs. For example, you might want to perform normal backups every three days and perform daily differential backups in between the normal backups.

Choose backup strategies that meet the backup storage capacity and load restrictions of your networks. Back up Certificate Services at least daily so that no more than one day's worth of certificate transactions is lost if the hard disk that contains the certificate database fails.

In addition to routine backups, you can use the Certification Authority Backup wizard to create an archive that contains the CA's private key, certificate, and configuration data. The archive is then updated only when the CA's data changes. The archive can be used to restore CAs to service even if something happens to the routine backup sets.

Restore Considerations

When the restore of a CA is complete, it is important that you make a new full backup of the certificate server database. This is necessary to truncate the restored log files and to establish a base backup set for future restores. Backups that are performed after a restore cannot be mixed with backups (either full or incremental) that are taken before the restore—that is, after a Certificate Services database is restored and has progressed to a subsequent state, you cannot use the prerestoration backups to restore the database to that subsequent state.

When you are restoring a failed CA with Windows 2000 Backup, you must restore Internet Information Services as well as Certificate Services, or else Internet Information Services fails to start when the system is restarted. Certificate Services requires that the Internet Information Services be running to support the Web Enrollment Support pages.

When you are restoring Certificate Services, if the database logs are not manually deleted, Certificate Services is brought up-to-date. If the logs are manually deleted, Certificate Services is restored to the point in time that the backup was performed. By default, the certificate database and the request log are installed at the following location:

```
<Drive:>WINNT\System32\CertLog
```

where *<Drive:>* is the letter of the drive where the CA is installed.

Revoking Certificates

The Windows 2000 Certificate Services certificate database records information for each certificate the CA issues. You can use the Certification Authority console to revoke issued certificates. For example, you might revoke the certificates issued for employees who are terminated or who have transferred to another unit. You also might revoke certificates when you suspect or discover that private keys have been compromised or misused. Until they expire, revoked certificates are published in the certificate revocation list.

When a certificate has been revoked, it is invalid and cannot be made valid again. If you revoke a certificate by mistake, you can re-issue a new valid certificate to take the place of the revoked certificate.

To use the Certification Authority console to revoke a certificate, select the Issued Certificates container for the CA and click the certificate in the details pane of the console. Then click **Action**, **All Tasks**, and **Revoke Certificate**. When the **Certificate Revocation** dialog box appears, select a reason code from the list in the **Select a reason code** box, and then click **Yes** to revoke the certificate. Reason codes include: Unspecified, Key Compromise, CA Compromise, Change of Affiliation, Superceded, Cease of Operation, and Certificate Hold. Revoked certificates are moved to the Revoked Certificates container of the CA.

Publishing Certificate Revocation Lists

Windows 2000 Certificate Services publishes periodic certificate revocation lists (CRLs). However, you can also use the Certification Authority console to manually publish a new CRL at any time. For example, you might want to publish a new CRL immediately after revoking certificates.

To publish CRLs manually, right-click the Revoked Certificates container for the CA. Then click **All Tasks** and **Publish**. When the **Certificate Revocation List** dialog box appears, click **Yes** to replace the old CRL.

Approving or Denying Certificate Requests

Windows 2000 Certificate Services stores pending requests in the Pending Request queue. For enterprise CAs, requests are processed automatically and the request is either approved or denied. By default, for stand-alone CAs, certificate requests are stored in the Pending Request queue for review by a CA administrator. You can use the Certification Authority console to review pending requests and either approve or deny the request. Approved requests are issued by the CA.

▶ **To use the Certification Authority console to approve a certificate request**

1. Click the **Pending Requests** container for the appropriate CA.

   The pending certificate requests appear in the details pane of the console.

2. Right-click the appropriate certificate request, and then click **All Tasks** and **Issue.**

   The CA issues the certificate.

▶ **To use the Certification Authority console to deny a certificate request**

1. Click the **Pending Requests** container for the appropriate CA.

   The pending certificate requests appear in the details pane of the console.

2. Right-click the appropriate certificate request, and then click **All Tasks** and **Deny.**

   The **Deny Certificate Requests** dialog box appears.

3. Click **Yes** to deny the certificate request.

   The certificate request is moved to the Failed Requests container.

Renewing Certification Authorities

If a CA's certificate expires, the CA can no longer provide certificate services. Before the CA certificate expires, you can use the Certification Authority console to renew the CA to provide uninterrupted certificate services. The interval that is required for CA renewal depends on the certificate life cycle that you designed for the public key infrastructure.

After you renew a CA, the CA continues to issue certificates by using the new CA certificate, and the cycle starts over. The prerenewal CA certificate remains trusted, so nonexpired certificates that were issued by the prerenewal CA continue to be trusted until they expire or are revoked.

You have the option of renewing the CA certificate by using the existing key set of the prerenewal CA certificate. However, the longer a key set is in use, the greater the risk the key set might be compromised. The risks of longer key lifetimes involves many complex factors, including key length and protection from attacks. For more information about risk factors for cryptographic keys, see "Cryptography for Network and Information Security" in this book.

▶ **To use the Certification Authority console to renew a CA certificate**

1. Select the CA node, and then click **Action**. Then click **All Tasks** and **Stop Service** to stop the CA. If you skip this step, you are later prompted to stop the CA.
2. Click **Action**, and then click **All Tasks** and **Renew CA Certificate**.

   The **Renew CA Certificate** dialog box appears.
3. Click **Yes** to generate a new key set, or click **No** to reuse the old key set. Then click **OK**.

   For root CAs, the certificate is renewed and no further action is required. For subordinate CAs, the **Complete this CA Installation** dialog box appears.
4. Type the domain name of the server for the parent CA in the **Computer Name** box, or click **Browse** to select the server.

   The **Parent CA** box displays the name of the CA that is running on the server computer that you have selected.
5. Click **OK**.

   The renewal request is sent to the parent CA to process. When the parent CA issues the new certificate, the CA certificate of the child CA is renewed.

Root CA certificates are renewed with the same lifetime as the original certificate. Subordinate CA certificates are renewed with the lifetime that is determined by the parent CA.

Recovering Encrypted Data

Windows 2000 supports the encryption of persistent data by EFS and secure mail systems. Encrypted data is usually readable only to the user who possesses the required private key to unlock the data. However, if the user's private key is lost or damaged, the encrypted data becomes unusable unless there is a means to restore the plaintext or the private key to the user. Furthermore, if a user who has encrypted information leaves the organization or is terminated, organizations can lose access to valuable encrypted information unless there is a means for someone else besides the user to recover the encrypted information.

When you deploy EFS or secure mail, implement a recovery program and policies to ensure that users' encrypted data can be recovered. EFS provides for recovery agents (trusted administrators) who can recover encrypted files. Many secure mail systems, such as Microsoft® Exchange Server, provide a key recovery database so that trusted administrators can restore users' private keys when necessary for users to read their encrypted mail (for example, when a user's private key is corrupted).

Recovery for Encrypting File System

EFS provides for data recovery agents. By default the domain Administrator user account (the local Administrator account for the first domain controller installed in the domain) is issued an EFS recovery certificate. You can use this account to recover files encrypted by EFS users in the domain. The private key for EFS recovery is stored on the local computer where the EFS recovery account is located. You must perform EFS recovery operations on the computer where the private key that is used for recovery resides.

You can configure Encrypted Data Recovery Agents policy to designate alternative recovery agents. For example, to distribute the administrative workload in your organization, you can designate alternative EFS recovery accounts for categories of computers grouped by organizational units. You can use Encrypted Data Recovery Agent policy to designate recovery accounts on computers to be used for EFS recovery operations.

You must deploy a CA to issue EFS Recovery Agent certificates to the EFS recovery accounts you want to designate by means of Encrypted Data Recovery Agents policy. You can issue certificates for EFS recovery with an enterprise CA or a stand-alone CA.

For enterprise CAs, by default, members of the Domain Admins and Enterprise Admins security groups are granted permissions to enroll for EFS Recovery Agent certificates. To change the default certificate enrollment settings, modify the ACLs for the EFS Recovery Agent certificate template. You can request an EFS Recovery Agent certificate by using the Certificate Request wizard or by using the Advanced Certificate Request page for an enterprise CA.

For stand-alone CAs, you can use the Advanced Certificate Requests form to request a recovery agent certificate by entering **1.3.6.1.4.1.311.10.3.4.1** as the object identifier in the **Usage OID** box.

The **cipher** command-line program is used to recover EFS files. The recovery operation decrypts the encrypted file to plaintext, which is readable by others. Therefore, administrators must take precautions when they are transferring the plaintext back to the user to ensure that the confidentiality of the information is preserved. For more information about **cipher**, see Windows 2000 Server Help.

For EFS encrypted files, the recovery agent information is refreshed every time the file system performs an operation on the file (for example, when the file is opened, moved, or copied). However, if an encrypted file is dormant for a long time, the recovery agents can expire. To ensure that dormant encrypted files can be recovered, maintain archives of the recovery agent certificates and private keys. To create an archive, export the certificate and its private key to a secure medium and store it in a safe location. When you export private keys, you must provide a secret password for authorizing access to the exported key. The secret key is stored in an encrypted format to protect its confidentiality.

To recover dormant files with expired recovery agent information, import the appropriate expired recovery agent certificate and private key from the archive to a recovery account on a local computer and then perform the recovery. To view recovery agent information for an encrypted file, use the **efsinfo** tool. For more information about **efsinfo**, see Windows 2000 Tools Help.

For more information about EFS and EFS recovery, see "Encrypting File System" in this book.

Recovery for Secure Mail

The Windows 2000 public key infrastructure does not provide a key recovery system for secure mail. However, to provide key recovery services, you can deploy secure mail systems, such as Exchange Server.

Exchange Server maintains users' private keys in a central protected store. Security administrators can use the Key Management server (KM server) to recover keys and restore the keys to users as necessary. For more information about KM Server, see Exchange Server Help and the *Microsoft® BackOffice® Resource Kit.*

Anyone who can obtain a user's private key can impersonate that user in e-mail transactions or read confidential mail that is intended for that user. Therefore, it is recommended that administrators take precautions when transferring keys back to users to ensure that the confidentiality of the keys is preserved.

Using the Certificate Services Command-Line Programs

Windows 2000 Server provides the following three command-line programs for Certificate Services:

- CertUtil.exe
- CertReq.exe
- CertSvr.exe

These command-line programs provide extended functionality and control of certificate services. The use of the command line is primarily intended for developers and knowledgeable certification authority administrators.

For more information about the command-line programs, see Certificate Services Help.

CertUtil.exe

You can use CertUtil.exe to perform the following tasks:

- Dump certificate services configuration information, certificate requests, certificates, or certificate revocation lists to files.
- Get the certification authority (CA) configuration string.
- Retrieve the CA signing certificate.
- Revoke certificates.
- Publish or retrieve a certificate revocation list.
- Determine if a certificate is valid or if the encoding length is incompatible with old enrollment controls.
- Verify one or all levels of a certificate chain.
- Resubmit or deny pending requests.
- Set attributes or an integer or string value extension for a pending request.
- Verify a public/private key set.
- Decode files that are based on hexadecimal or base 64.
- Encode files to base 64.
- Shut down the Certificate Services server.
- Display the database schema.
- Convert a Certificate Server version 1.0 database to a Windows 2000 Certificate Services version 2.0 database.
- Back up and restore the CA keys and database.
- Display certificates in a certificate store.
- Display error message text for a specified error code.
- Import issued certificates that are missing from the database.
- Set and display certification authority registry settings.
- Create or remove Certificate Services Web virtual roots and file shares.

CertReq.exe

You can use CertReq.exe to request certificates from a certification authority. CertReq submits certificate requests by using PKCS 10 certificate request files and PKCS 7 certificate renewal files. You also can use the advanced options on the Web Enrollment Support pages to submit certificate requests by using PKCS 10 and PKCS 7 files.

CertSrv.exe

CertSrv.exe is the server engine program that is run when the Certification Authority service starts. For troubleshooting purposes only, you can run CertSrv as a stand-alone application in a command prompt window. When CertSrv is running in the diagnostics mode, it displays a log of its actions in the console window. You can start CertSrv as a service through **Services** in Control Panel.

Disaster Recovery Practices

Disasters, such as hard disk failure or a compromised CA certificate, can disrupt certificate services. You can take various steps to minimize the impact of such disasters and to ensure timely recovery from server or network disasters. The following practices can reduce the risk of failed or compromised CAs:

- Using preventive practices for servers.
- Providing security for certification authority servers.
- Protecting private keys for certification authority servers.
- Developing recovery plans.

Using Preventive Practices for Servers

The server where a CA is installed can fail, resulting in a disruption of certificate services. You can use the following preventive practices to reduce the risk of CA failures and to minimize the disruption of CA services:

- Provide duplicate CA services so that if one server is offline, another server can still issue the appropriate certificates.
- Back up CAs frequently so that they can be restored with a minimal loss of data.
- Install certificate services on hard disks by using disk arrays and redundant array of independent disks (RAID) Level 5 protection.
- Prepare recovery plans and train administrative staff on recovery plans.
- Maintain records of all server and CA configuration information so that exact configurations can be easily restored.
- Maintain replacement servers in standby or in ready stores for immediate recovery.

Providing Security for Certification Authority Servers

Computers that run CA services can be priority targets for attack by intruders who maliciously want to disrupt network services or compromise the security of network and information systems. If intruders can gain unauthorized access to a CA server or exploit weaknesses in the security of the server, they can gain access to valuable network resources and compromise the security of the affected portion of the certification trust chain. Therefore, you should provide higher security for CA servers than for regular servers.

The risk of attacks on your CAs depends on many factors, including how secure your networks are, the value to be gained by a successful attack, and the costs of attempting the attack. If the CA is inside your firewall and used on the intranet for routine business purposes, the risk of attack might be low. However, if the CA is outside your firewall and used for an extranet, the risk of attack might be high.

If a CA is compromised, there can be considerable damage and cost to your organization. The damages and costs of a compromised CA include the following:

- Stolen proprietary information.
- Efforts spent investigating and stopping the intruder's attacks on the network.
- Failed or disrupted network services.
- Destroyed or corrupted network resources.
- Efforts to recover from the CA compromise and redeploy new CAs and certificates.

A compromised root CA is far more costly than a compromised intermediate CA or issuing CA. You can deploy multiple CA hierarchies to reduce the impact of a single compromised CA on your organization.

To determine what security measures are appropriate for CAs, weigh the estimated cost of providing security measures against the estimated cost of compromised CAs. Security measures for CA servers can include the following:

- Maintaining servers in secure data centers and controlling physical access to trusted administrators.
- Using hardware CA devices or hardware-based CSPs to provide maximum security for the CAs' private keys.
- Configuring server security settings for high security levels, such as those security levels provided by the High Security template.
- Using the Windows 2000 system key (SysKey) to provide additional encryption protection of CA servers' protected stores.

- Performing security auditing to monitor for potential attacks on CA servers.
- Restricting user rights assignments to limit user rights to the appropriate administrator group. (No other users or groups have rights or permissions to view or perform any tasks on the local CA computer.)
- Disabling unnecessary services so they cannot run on CA servers; running unnecessary services provides a mechanism that intruders can exploit.
- Implementing security policies and procedures to control the deployment of CAs in the enterprise.

Choosing security measures for a CA involves weighing the costs of implementing and maintaining those security measures against the risks of potential attack on the CA and the potential costs of a CA compromise. Higher risks of attacks on the CA and higher costs of a CA compromise generally justify higher costs for security measures to protect the CA. Provide the most protection for root CAs, and provide more protection for intermediate CAs than for issuing CAs.

For example, your organization decides to protect a large amount of highly valuable and confidential information by using public key security solutions. You also decide to acquire expensive hardware CA devices for root CAs and store the root CAs in maximum security vaults that are located at your headquarters for safekeeping. You authorize access to the root CAs for trusted administrators so that they can certify intermediate CAs for each of your business units. The intermediate CAs are offline Windows 2000 CAs, which are disconnected from the network and maintained in locked data centers by the administrator for each business unit. The intermediate CAs are used to certify issuing Windows 2000 CAs as necessary to meet the certificate needs of each business unit. Issuing CAs are Windows 2000 enterprise CAs or stand-alone CAs that are maintained in secure data centers by each business unit. Your organization's security policy includes strict procedures and controls for requesting, authorizing, and implementing root CAs, intermediate CAs, and issuing CAs in the enterprise.

However, if your organization uses public key security solutions to protect information with relatively low value, you might decide to deploy offline Windows 2000 root CAs that are locked in data centers, rather than expensive hardware-based CAs locked in vaults. You might allow business units to maintain intermediate CAs and issuing CAs outside data centers. You also might place fewer restrictions on requesting, authorizing, and implementing CAs.

You can deploy Windows 2000 Certificate Services by using the Microsoft Base CSPs to provide CA security that meets a wide range of needs. However, when you need to provide the highest security for CAs, consider using hardware-based CAs. For more information about hardware-based cryptography solutions that work with Windows 2000 Server and Certificate Services, contact the appropriate hardware vendors.

Protecting Private Keys for Certification Authority Servers

If intruders can access a CA computer either physically or through the network, they might decode the private key and then impersonate the CA to gain access to valuable network resources. Intruders who impersonate a CA can cause widespread damage by stealing information, disrupting network services, or destroying network resources. A compromised CA key undermines and invalidates all security protection provided by that CA and any CA hierarchy deployed below it. To reduce the risks of intruder attacks on CA keys, consider using the following practices.

Provide Security for Certification Authority Servers Provide security for CA servers as discussed earlier in this chapter. Providing physical security minimizes the risk that intruders can gain access to the CA server or the protected store (whether hardware-based or software-based) where the CA key resides. Providing network and server (software) security minimizes the risk that intruders can gain access to the CA server or exploit applications and services that are running on the server to compromise the CA key.

Provide Enhanced Security for Certification Authority Keys Use hardware-based CSPs when you want to provide maximum security for private keys because keys are stored on tamper-resistant hardware devices and keys are never exposed to the operating system. Use SysKey to provide extra protection for CAs' private keys that are stored by Microsoft CSPs.

Use Large Keys for Certification Authorities Large CA keys reduce the risks of key attacks, but large keys also require more storage space as well as more computer processing power to sign certificates. Consider using the largest key lengths that are feasible depending on key storage requirements and CA performance requirements.

For example, a 4,096-bit CA key generally provides a great deal of key protection, but signing certificates with such a long key takes a long time, even if you are using crypto-accelerator boards. A 4,096-bit CA key might perform acceptably for root CAs or intermediate CAs that are used infrequently only to certify subordinate CAs. Although, some CAs with hardware-based CSPs might not support the storage of a 4,096-bit key.

However, a 4,096-bit CA key would likely cause unacceptably slow performance for most issuing CAs. For issuing CAs, use key lengths that are as long as feasible and that enable adequate CA performance to support your long-term certificate services goals. You can often use crypto-accelerator boards to improve performance and enable longer keys for issuing CAs. Test the performance CAs in labs and in pilot programs by using the proposed CA key lengths before you deploy CAs in the production environment.

Use Appropriate Lifetimes for CA Keys The longer CA keys are valid, the greater the risk of key compromise because attackers have more time to attempt cracking the key. There is no simple formula to determine maximum key lifetimes. However, the adequacy of longer key lifetimes depends largely on how well protected the key is and how long the key is. In general, longer keys can have longer key lifetimes. Likewise, keys with more secure storage can have longer lifetimes. For example, keys stored in tamper-resistant hardware crypto-devices are safer than keys stored on local computer hard disks. Therefore, for the same-sized keys, keys stored in hardware crypto-devices usually can have longer safe key lifetimes than keys stored by software CSPs on hard disks.

For more information about the major risk factors for cryptographic keys, see "Cryptography for Network and Information Security" in this book.

Developing Recovery Plans

You can develop recovery plans to help restore CAs if certificate services fail or CAs are compromised. It is recommended that you test recovery plans to ensure that they work as intended. Hold training sessions for your staff to ensure that they know how to use the recovery plans.

Recovery plans can include the following:

- Recovery procedures and checklists for administrators to follow
- Recovery toolkits or pointers to the toolkits
- Contingency plans

Failed Certification Authority

A CA can fail for a variety of reasons, such as a server hard disk crash, a failed network card, or a server motherboard failure. Some failures can be corrected quickly by locating and correcting the problem within the CA server. For example, you can replace a failed network card or a failed motherboard and restart the computer to restore certificate services.

If a hard disk has failed, you can replace the hard disk and restore the server and the CA from the most recent backup set. If the CA is damaged or corrupted, you can restore the CA from the server's most recent backup set. If you must replace the server, configure the new server with the same network name and IP address as the failed CA server. Then install the CA with the original configuration information and the original private key and certificate for the CA.

Select the Windows Component wizard **Advanced options** when you are installing the CA to enable you to reuse the key and the associated certificate. In the **Public and Private Key Selection** page, you must click **Use existing keys**, select the key from the list, and then click **Use the associated certificate**. You can also click **Import** to import a private key from archives. The CA information that is contained in the certificate is automatically used for the **CA Identifying Information** page. The CA is installed as the original CA.

If **Use the associated certificate** is grayed out, you cannot use the subject information contained in the certificate. If so, you must configure the **CA Identifying Information** page exactly as the original, or else the process cannot work. Furthermore, on the **CA Certificate Request** page, you must click **Save the request to a file** instead of requesting a certificate from an online CA (otherwise, the parent CA issues a new certificate for the CA). After the CA is installed, you can use the Certification Authority console to install the original certificate to certify the CA.

You must click **Preserve existing certificate database** on the **Data Storage Location** page to preserve an existing CA database. Otherwise, you might overwrite the existing database and destroy the information that is contained in the database.

It is important to keep in mind that only the associated certificate works with the private key because the certificate contains the complementary public key. It is also important to remember that the identifying information for the CA must match the Subject information in the certificate Subject field *verbatim* or else the CA does not work. The following information that is entered on the **CA Identifying Information** page during installation of the CA is used for the certificate Subject field:

- CA name
- Organization
- Organizational unit
- Locality
- State or province
- Country/region
- E-mail

The information in the Subject field is case sensitive, so review the information on the **CA Identifying Information** page carefully before you complete the installation process. You can view a certificate's Subject information with the **Certificate Details** dialog box by selecting **Subject**.

After a replacement CA is installed and running, you can use Windows 2000 Backup or the Certification Authorities Restore wizard to restore the CA configuration data from the most recent backup set.

Compromised Certification Authority

When a CA is found to be compromised, the only solution is to revoke the CA's certificate. Revoking a CA's certificate invalidates the CA and its subordinate CAs, as well as invalidating all certificates issued by the CA and its subordinate CAs. If you discover a compromised CA, it is recommended that you perform the following activities as soon as possible:

- Revoke the compromised CA's certificate.
- Publish a new CRL containing the revoked CA certificate.
- Remove compromised CA certificates from Trusted Root Certification Authorities stores and CTLs.
- Notify all affected users and administrators of the compromise and inform them that certificates issued by the affected CAs are being revoked.
- Repair security holes that led to the compromise.

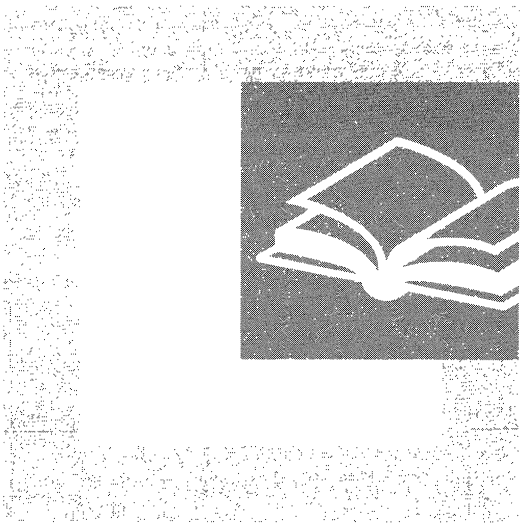
To restore the CA hierarchy, you must redeploy new CAs to replace the compromised hierarchy. You must then reissue the appropriate certificates to users, computers, and services.

Additional Resources

- For more information about how to develop custom applications by using Windows 2000 Certificate Services and how to use the services of Microsoft CryptoAPI and CSPs, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.
- For more information about Internet Engineering Task Force (IETF) drafts and recommendations, see the Internet Engineering Task Force (IETF) link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.
- For more information about Public Key Cryptography Standards (PKCS), see the RSA Data Security link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.
- For more information about the FIPS 140-1 standard, see the National Institute of Standards and Technology (NIST) link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. Search for "Security Requirements for Cryptographic Modules."
- For more information about FIPS 140-1 as a de facto international standard for cryptographic modules, see the International Organization for Standardization link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. Search for "International Standard 15408: Evaluation Criteria for Information Technology Security."
- For more information about the "Certified for Microsoft Windows" program and a list of currently compatible smart card products, see the Microsoft Windows Hardware Compatibility List link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.
- For more information about the security technologies in Microsoft products, including cryptography export restrictions and licensing requirements, see the Microsoft Security Advisor link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

PART 3

Enterprise Technologies

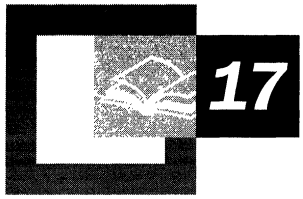


The successful implementation and efficient use of the Distributed file system and Windows Clustering continue to be an important issues for enterprise administrators. Part 3 examines how these features are enhanced in Microsoft® Windows® 2000 and provides a useful source of troubleshooting information.

In This Part

CHAPTER 17

Distributed File System



Distributed file system (Dfs) is a Microsoft® Windows® 2000 service that assists network administrators in managing file resources on distributed enterprise networks. Dfs also makes finding files on networks easier for users by creating a single hierarchical view of the shared data on different computers in one or more domains.

Dfs runs on Windows 2000–based servers; but, unlike the NTFS file system or the file allocation table (FAT) file system, it is a background process that integrates file systems and shared folders into a single, logical namespace. This namespace reflects the network storage resources that are available to users on your network.

The information in this chapter is for administrators who want to understand how Dfs works, who need to design a Dfs system, and who must implement and support Dfs on their networks.

In This Chapter

Related Information in the Resource Kit

- For more information about Active Directory™, the directory service that is included in Windows 2000, see the chapters about Active Directory in this book.
- For more information about the File Replication service, see "File Replication Service" in this book.

Introduction to Dfs

Distributed file system (Dfs) consists of software residing on network servers and clients that transparently links shared folders located on different file servers into a single namespace for improved load sharing and data availability. Dfs organizes these resources in a tree structure, starting with a root located on a Windows 2000–based server. From the root, you can define links to shared folders distributed throughout your organization's local or wide area networks, without regard to their physical location. Instead of seeing a physical network of dozens of file servers, each with a separate directory structure, users now see a few logical directories that include all of the important file servers and shared folders. Each shared folder appears in the most intuitive place in the directory, no matter where the folder actually resides.

Dfs does for servers and shared folders what file systems do for hard disks. File systems provide uniform named access to collections of sectors on disks; Dfs provides a consistent naming convention and mapping for collections of servers, shared folders, and files. In addition, Dfs is not limited to a single file protocol and can support the mapping of servers, shared folders, and files regardless of the file client being used, provided the client already supports the native server and shared folder and that the client shares a common protocol with the Dfs server and referral computer to which it connects.

What Dfs Does

Dfs provides the ability to logically group shared folders on different servers and to transparently link shared folders into a single, hierarchical namespace.

Historically, with universal naming convention (UNC) names, a user or application would specify the physical server and shared folder to gain access to file information—for example, to *\\Server\Shared_folder\Path\File_name*. Even though a UNC can be used directly, if it is an often-used UNC, it is usually mapped to a drive letter, where the drive letter denotes *\\Servers\Shared_folder*. The user then navigates beyond the redirected drive mapping to the data to which he or she wants to gain access. An example might be the following, where *X* is the drive letter:

```
copy X:\path\more_path\…\file_name
```

As networks continue to grow in size and as organizations begin to use existing storage, both internally and externally, for purposes such as intranets, mapping a single drive letter to individual shares becomes inefficient. And, despite their ability to directly reference UNC naming, users can become overwhelmed by the increasing number of places to which they must gain access to retrieve data.

Dfs solves these problems by linking file servers and shared folders to a simpler, more meaningful namespace. This new namespace permits shared folders to be related to each other within a single hierarchy. Because Dfs maps the physical storage to a logical representation, the net benefit is that the physical location of data becomes transparent to users and applications.

Features and Benefits

The features of Dfs and their benefits for administrators and users are presented in Table 17.1.

Table 17.1 Dfs Features and Benefits

Feature	Description	Benefit
Unified namespace	Linking together network storage resources creates a single, hierarchical structure that behaves as a single, high-capacity hard disk.	Provides a unified view of shared folders that you can customize for individual users and groups. A single drive mapping can reflect hundreds of shared folders.
Name transparency	Users can navigate the logical namespace without having to know the physical locations of the data. The actual data can be moved to any server and the logical Dfs namespace can be reconfigured without affecting the user's view of the namespace.	Provides increased administrative flexibility. You can move shared folders between servers or disk drives without affecting the users' access to the data.
Flexible storage management	Individual shared folders that are part of the Dfs root can be taken offline without affecting the remaining portion of the namespace. You can remove and replace an entire server by adding the path of a new server to the namespace.	Allows management of physical network storage independent of its logical representation to users. The Dfs namespace can be extended at any time to incorporate additional disk storage.

(continued)

Table 17.1 Dfs Features and Benefits *(continued)*

Feature	Description	Benefit
Graphical administration	Each root in the Dfs topology is administered with an easy-to-use graphical interface in Microsoft Management Console (MMC). This permits configuration of shared folders and Dfs links and the administration of remote Dfs roots.	Reduces training time and the need for full-time server administrators.
Load sharing	Roots and replicas hosted on two or more computers provide load sharing. As users request files from the node in the Dfs namespace, they randomly select one of the computers for the accessed node.	Randomly distributes file access across multiple disk drives or servers to balance loads and improve response time during peak usage periods.
Availability	Roots and replicas hosted on two or more computers can be specified as a replica set under the same logical Dfs name to provide alternate locations for accessing data. If one copy becomes unavailable, another is selected automatically. Copy replication can be manual or automatic.	Keeps important data accessible at all times, even when a server or disk drive fails or when a shared folder or file becomes corrupted. Dfs roots also can be replicated to eliminate them as single points of failure.
Security integration	No additional overhead is necessary for implementing security. A user who connects to a Dfs root is permitted access only to files for which he or she has appropriate permissions. Using the optional FRS replication, identical permissions can be set on physically separate replicas.	Uses standard Windows 2000 file and directory permissions for easy administration and secure access.

(continued)

Table 17.1 Dfs Features and Benefits *(continued)*

Feature	Description	Benefit
Intelligent client caching	Information about the Dfs namespace for a shared folder is not cached on a client until a user actually requests access to the folder. The next time the client requires access to the same portion of the Dfs namespace, the client uses the cached referral rather than obtaining a new referral.	Using a local cache on the client minimizes network traffic and improves user response times.
Dfs awareness for clients running Microsoft® Windows® 95 and Windows® 98	Because Dfs support has been integrated into the server message block (SMB) redirector, no additional memory is required to run the Dfs clients for Microsoft® Windows NT® version 4.0 and Windows 98. Dfs includes an add-on component that permits Windows 95 users to navigate the Dfs namespace. However, clients that are running Windows 98 and Windows 95 can gain access to non-SMB volumes only through a server-based gateway such as Gateway Service for NetWare.	Requires no additional resources on client systems for Dfs functions. Extends Dfs benefits to Windows 98 and Windows 95 users.
Interoperability with other network operating systems	Any shared folder that is accessible through a redirector on Windows 2000 can participate in the Dfs namespace. This can be through client redirectors or through server-based gateway technology.	Supports heterogeneous network operating systems such as MS-DOS®, Microsoft® Windows® version 3.*x*, Windows 95, Windows 98, and Windows NT 4.0 and earlier.

Basic Dfs Concepts

Windows 2000 Dfs comes in two configurations: domain-based and stand-alone.

Domain-based Dfs stores its configuration information in Active Directory. Because this information is made available on multiple domain controllers in the domain, domain-based Dfs provides high availability for any distributed file system in the domain. A domain-based Dfs root has the following characteristics:

- It must be hosted on a Windows 2000 domain controller.
- It has its topology published automatically to Active Directory.
- It can have root-level shared folders.
- It supports root and file replication through the File Replication service (FRS).

Stand-alone Dfs stores its configuration in the registry of the local computer. It is intended for backward compatibility with previous versions of Dfs. A stand-alone Dfs root has the following characteristics:

- It does not use Active Directory (or FRS).
- It cannot have replicas at the root level.

New Dfs implementations must use a domain-based configuration to take advantage of Active Directory. Stand-alone Dfs is best for earlier implementations of Dfs.

Nomenclature

These are the key terms that are associated with Dfs:

Dfs topology. Overall logical hierarchy of a distributed file system, including elements such as roots, links, shared folders, and replica sets, as depicted in the Dfs administrative console. This is not to be confused with *Dfs namespace*, which is the logical view of shared resources seen by users.

Dfs root. The share at the top of the Dfs topology that is the starting point for the links and shared files that make up the Dfs namespace. A Dfs root can be defined at the domain level for domain-based operation or at the server level for stand-alone operation. Domain-based Dfs can have multiple roots in the domain but only one root on each server.

Root replica. The server that duplicates a Dfs root to provide greater availability. The server that is hosting the Dfs root is responsible for handing out referrals to clients for shared folders. If that server becomes unavailable and a root replica has not been created, the Dfs namespace becomes inoperative. Replicas can also be created for existing Dfs links.

Dfs link. Part of the Dfs topology that lies below the Dfs root and forms a connection to one or more shared folders or another Dfs root. It does this by mapping a DNS name to the standard UNC of the target shared folder.

Dfs shared folder. Files or folders in the Dfs namespace that are shared by users with proper permissions. Shared folders can exist at the root level (domain-based Dfs only) or be referred to by Dfs links.

Replication policy. Configuration rules that enable automatic replication of files between computers where replication has been enabled. Replication is supported on the root and child folders of domain-based Dfs. A stand-alone Dfs cannot take advantage of automatic replication because it does not use Active Directory. The replication policy can be different for each replica set in the Dfs namespace.

Partition knowledge table (PKT). A table that maps root and replica nodes in the Dfs namespace to Active Directory sites and physical servers. For a domain-based Dfs root, the PKT is stored in Active Directory and made available to each domain controller in a domain. For a stand-alone Dfs root, the PKT is stored in the individual server's registry. When a Dfs client gains access to a shared folder in the Dfs namespace, it caches that portion of the PKT for the length of time specified in the TTL.

Referral. The referral is the physical server and share residing in the PKT that clients connect to.

Time-To-Live (TTL). The length of time that a Dfs client stores the referral information from the PKT when it accesses a shared folder. Dfs clients request a new portion of the PKT when the TTL expires or when the client is restarted. The TTL resets if the shared folder is visited before expiration. It is configurable on a per-link basis.

Revision level. Refers to Dfs client compatibility. There are three revisions of Dfs clients that can be viewed in Network Monitor traces. Clients that are running Windows NT version 4.0, Windows 98, and Windows 95 support Dfs revision level 2; clients that are running Windows 2000–based support revision level 3. Version 1 clients do not exist. Dfs clients and servers negotiate the highest common protocol revision supported.

A few of the terms in the version of Dfs included in Windows NT 4.0 were renamed in Windows 2000. Table 17.2 shows a list of these terms and their replacements.

Table 17.2 Dfs Nomenclature Map

Dfs in Windows NT 4.0	Dfs in Windows 2000
Alternate volume	Replica
Child node, junction, inter-Dfs link	Dfs link
Fault-tolerant Dfs	Domain-based Dfs
Dfs volume, leaf, branch	Shared folder

Processes

The following processes are used by Dfs:

- Maintaining the PKT
- Caching referrals by clients
- Gaining access to a Dfs shared folder
- Linking logical names to physical addresses
- Replicating shared folders
- Switching between replicas during failover
- Establishing security

Maintaining the Partition Knowledge Table (PKT)

The Dfs topology is stored in the server–based PKT. When Dfs roots and links are accessed by users, the machine caches that portion of the PKT and connects to one of the servers in the referral list.

The PKT maps the logical Dfs namespace into physical referrals, as shown in Table 17.3. (Replicas appear as a list for a single Dfs link.)

Table 17.3 PKT Location Mapping

Dfs path	Link [server and share]	Time-To-Live
DFS name #1	UNC name #1	5 minutes (default)
	UNC name #2	5 minutes
	UNC name #3	5 minutes
DFS name #2	UNC name #4	5 minutes
	UNC name #5	5 minutes

A flag in the PKT indicates whether the shared folder is hosted on an earlier version of Windows NT or if it is located on Novell NetWare, network file system (NFS), or computers that are running Windows 98–based, or Windows 98–based computers.

The PKT also stores site information, which is used to connect users to Dfs roots and links in the same site. Windows 2000 computers accessing Windows 2000 domain based Dfs Roots and links give preference to servers in the same Active Directory site when they exist.

All other combinations of clients and servers do not provide site awareness but gain load balancing for Dfs Links.

Note The site information is stored in the PKT when Dfs configuration is created. If you move the Dfs server to a different site, you must redefine the Dfs configuration. This is important if you are prestaging servers from a central location.

The PKT is a sorted lookup table that requires about 400 bytes per entry. One PKT resides in Active Directory for each Dfs root in a domain-based Dfs. The PKT for a stand-alone Dfs resides locally in the registry.

Caching Referrals by Clients

Clients that have access to the Dfs namespace cache portions of the server-based PKT locally to improve performance. When a user traverses a Dfs link in the namespace, the client receives a referral from the appropriate Dfs server and then adds a PKT entry to its local cache. When the client needs to revisit that portion of the Dfs namespace again, it uses the mapping from its locally cached PKT.

When the Dfs client attempts to navigate a Dfs link, it first looks to its locally cached PKT entries. If the referral cannot be resolved, the client contacts the Dfs root for an updated PKT entry and resets the TTL. If the referral still cannot be resolved, an error occurs. If the referral is properly resolved, the client adds the referral to its local table of entries.

When a Dfs client obtains a referral from the PKT, the referral is cached for a period of time defined by the TTL parameter on the Dfs server. If the client reuses that referral, the TTL is renewed; otherwise, the cache expires. If a replica set exists for a particular referral, all replicas are sent to and cached by the client. The client then randomly selects which referral to use.

In Windows 2000, the TTL interval is assigned on a per-link basis. If the physical location of the underlying shared folder (or shared folders, if there are replicas) is fairly dynamic, you would want to set TTL for the Dfs link to a smaller value. This would cause the client to go back to the server for a fresh copy of the referral more frequently. Similarly, if the physical location of the underlying shared folder is static, a larger TTL value can be used. If you set the TTL value too large and the client accesses the Dfs link before TTL expiration, the client will not receive a new referral to learn about changes to the link.

Suppose, for example, that you had a Dfs link called \\Company\Sales\Contracts\Today, which contained the set of contracts that were created on the current day. This link refers to a physical folder on the \\Sales\Contracts share that corresponds to the current day. So today the Dfs link might refer to \\Sales\Contracts\1999\1231, but tomorrow the link would be modified to refer to \\Sales\Contracts\2000\0101, and so on. Users will always refer to \\Company\Sales\Contracts\Today to get to the current day's folder and do not need to be concerned with the underlying mapping. However, you would have to set the TTL value for this Dfs link to a short enough value to make sure the client goes back to the server to get the updated referrals each day.

Checking Referrals

Windows 2000–based clients contain a shell extension to Windows Explorer that you can use to do the following:

- See all the referrals for a Dfs link.
- Select a referral for a Dfs link.
- Refresh the referral cache for a Dfs link.

For more information about the Dfs tab provided by the shell extension, see "Tracking Shared Folders" later in this chapter.

Gaining Access to a Dfs Shared Folder

Accessing a shared file or folder in a Dfs namespace is done in exactly the same manner as a Windows NT 4.0, Windows 95, or Windows 98 client accesses any UNC path. This implies that anywhere that a physical UNC can be used, you can use a Dfs name that refers to an object in the logical Dfs namespace.

This includes the ability to specify a point in a Dfs namespace as the share that corresponds to a Dfs link in another Dfs namespace. This is how you can build up more complex hierarchies of Dfs namespaces from existing sets of Dfs namespaces.

Access to a domain-based Dfs is achieved through either of the following conventions using the shell or **net use** command:

\\Domain_name\Dfs_root

\\Server_name\Dfs_root

A Dfs client on Windows 2000 and Windows NT 4.0 can also enter a **net use** command to gain access to any point in the Dfs namespace; this is sometimes referred to as a deep **net use**.

NET USE * *\\Domain_name\Dfs_root\Dfs_path\Shared_folder*

NET USE * *\\Server_name\Dfs_root\Dfs_path\Shared_folder*

Because a domain-based Dfs root is hosted in a Windows 2000 domain, it is accessible by way of the domain name. This removes the burden of the user having to know the physical location of the share; now he or she has to traverse only the logical namespace or namespaces that exist in the domain. The second convention is also supported. A user can use this convention to gain access to a Dfs namespace by specifying one of the servers that hosts the domain-based Dfs root. In this case, that specific server is always used for referrals. In Windows 2000, both the domain name and server name can be specified as either a DNS name or a NetBIOS name.

Older Dfs-aware clients (Windows NT 4.0, Windows 95, and Windows 98) cannot connect to a domain-based Dfs root by its domain name until they are upgraded with an appropriate service pack. They can, however, connect to individual Dfs root servers that participate in a domain-based Dfs by using the second naming convention:

\\Server_name\Dfs_root\Dfs_path\Shared_folder

Gaining access to a stand-alone Dfs is always through the following convention:

\\Server_name\Dfs_root\Path\File

Linking Logical Names to Physical Addresses

When a client specifies the logical name of a shared folder, the referral process provides its physical address. If a Dfs link to another server is encountered, the process is the same. However, in this case, it is important to note that the referral process expressly searches for the longest referral—the one with the most backslashes (\)—that can be resolved from the requested path. This ensures that with a single referral, the final destination has been resolved.

For example, in Figure 17.1, *Dfs_link* represents a link from Server1 to Server2. Because Dfs can resolve only \\Server1*Dfs_root*\ locally, it fetches the longest path from the PKT: \\Server1*Dfs_root**Dfs_link*\. However, because *Dfs_root* is linked to a second server and share (\\Server2*Share*\), that server and share are substituted for the referral. In other words, when a client requests access to \\Server1*Dfs_root**Dfs_link**Share**File*, Dfs returns the longest path known from PKT knowledge. It first looks at the local cache, then asks the root server, and finally consults Active Directory. In this example, the referral for \\Server1*Dfs_root**Dfs_link* would map to the other server and share. Thus, the referral returns a physical address of \\Server2*Share**File*.

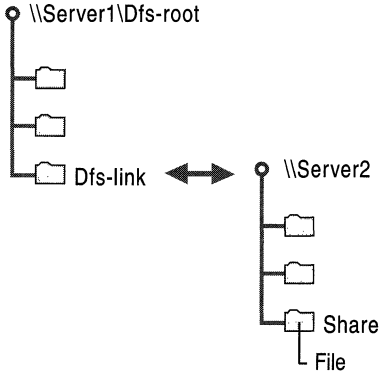


Figure 17.1 Referral Process Across a Dfs Link

The Dfs-aware redirector, SMB Services, and Dfs driver collaborate to reroute path-based operations to the server and share hosting the file or directory. For more information about the application programming interfaces (APIs) that provide this functionality, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Switching Between Replicas During Failover

Referrals are cached locally to maintain performance, and if replicas are available, all replicas are provided to the Dfs client. The client arbitrarily chooses which referral to use. Selection is random, although preference is given to replicas within the same site as the client.

After a referral is selected from the replicas, a session setup is performed (credentials are passed to the new server if a prior connection does not exist). If the selected referral fails, a failover process begins. The speed and implications of the failover depend on what the client was doing at the time of the failure, how the failure occurred, and how tolerant of delays an application is.

Scenario 1

A client is browsing through a replicated folder. The computer hosting the replica loses power or drops off the network for some reason. In order to fail over, the client must first detect that the hosting computer is no longer present. How long this takes depends on what protocol the client is using. Many protocols, such as TCP/IP, account for slow and loosely connected WAN links, and, as such, might have retry counts up to two minutes before the protocol itself times out. After that occurs, Dfs immediately selects a new replica. If none are available from the local cache, the Dfs client consults with the Dfs root to see whether the administrator has modified any PKT entries. If no replicas are available at the root, a failure occurs; otherwise, Dfs initiates a fresh replica selection and session setup.

Scenario 2

A client is browsing through a replicated folder. The computer hosting the replica loses the hard disk containing the replica, or the replica itself is deactivated. In this scenario, because the server hosting the replica is still responding to the client request, the failover to a fresh replica is nearly instantaneous.

Scenario 3

A client has open files. The computer hosting the replica loses power or drops off the network for some reason. In this scenario, you have the same protocol failover process described in Scenario 1. In addition, the failover depends on the application that previously had file locks from the previous replica to detect the change and establish new locks.

New attempts to open files trigger the same failover process that is described in Scenario 1. Operations on already open files fail with appropriate errors.

Scenario 4

A client has open files. The computer hosting the replica loses the hard disk containing the replica, or the replica itself is deactivated. In this scenario, you have the same rapid failover process that is described in Scenario 2. In addition, the failover depends on the application that previously had file handles from the previous replica to detect the change and establish new handles.

Replicating Files

The load balancing and fault tolerance of Dfs makes it well suited for software distribution shares, web content and internal documentation. Administrators may optionally enable automatic replication of files and folders between Windows 2000 computers using the Replication Policy command in the Dfs Administrative console. The replication policy can be different for each Dfs root and link in the Dfs namespace.

Replication of Dfs content is performed by the File Replication Service (FRS) which provides multimaster updates. For more information about file replication, see "File Replication Service" in this book.

Establishing Security

As each Dfs link is crossed and cached for the first time, the Dfs-aware client establishes a session setup with the server on the other side of the link. The credentials the user originally used to connect with Dfs are used (for example, **net use** * *Server\Dfs_Share* **/u:***domain\user*). If the user did not supply credentials, the credentials that are cached when the user logged on to his or her workstation are used.

ACLs

File access control lists (ACLs) are administered at each individual shared folder. There is no mechanism to administer ACLs systemwide from the Dfs root, nor is there an attempt to keep ACLs consistent between replicas. Several reasons account for this:

- A centrally administrated logical ACL database can be bypassed because users can issue the **net use** command directly to the physical resource.
- The logical Dfs root can cross between FAT and NTFS volumes, as well as contain shares from other network operating systems. There is no reasonable way to set an inherited Deny ACL that starts on a NTFS volume, passes to FAT, passes back to NTFS, and concludes on NetWare share.
- A tool that searches the logical namespace and sets ACLs appropriately would require a complicated message and transaction engine to ensure that the ACLs would be queued and updated over loosely connected or unreliable networks.
- Storage quotas available in Windows 2000 would require an additional burden of tallying storage for all possible users across all possible volumes to establish when users have exceeded their storage allotment.

Replicating Permissions

FRS replicates changes to file permissions on NTFS. If you change one replica's ACLs, they also change for each member of the replica set. If FRS is not being used to replicate shared folders automatically, you must set the permissions on each copy of a shared folder and manually propagate any changes that occur.

Getting Started

Following is an overview of how to get started with Dfs.

Administrator Perspective

Dfsgui.msc is the administrative console for Dfs in Windows 2000. With it you can create the Dfs root and build a Dfs topology. Remember the following sizing limitations when you implement Dfs:

- The maximum number of characters per file path is 260.
- The maximum number of replica members in a domain-based set is 256.
- The maximum number of Dfs roots per server is 1.
- The maximum number of Dfs roots per domain is unlimited.
- The maximum number of Dfs links or shared folders in a Dfs root is 1,000.

The general steps for creating the Dfs topology are listed here. For more information about specific procedures, see Windows 2000 Server Help.

1. Create a Dfs root.

   Dfs roots must be created on partitions generated with the version of NTFS used in Windows 2000 for security and content reasons. New roots must be domain-based to allow file replication.

2. Add Dfs links.

   You can add a Dfs link under the Dfs root to reference any root or link in the Dfs tree. If the referenced resource is not a computer that is running Windows 2000, the link can reference only shared folders.

   Note Linking from one Dfs link to another Dfs link is possible only if the second link is in a separate Dfs namespace. For example, you cannot have *Server**Dfs_root*\Dfs_link1\Dfs_link2 unless Dfs_link2 lies under another Dfs root.

   ---

3. Specify replicas for a Dfs root or link.

   Each Dfs root and shared folder can be replicated to back up its contents. Dfs-aware clients automatically select the nearest replica on the basis of the site topology information.

4. Manage replication for Dfs.

   You can remove replicas, take replicas offline or online, and enable or disable replication.

Administering a stand-alone Dfs root requires Admin privileges. Domain-based Dfs root administration requires Domain Admin privileges.

Client Perspective

Users browse through and connect to Dfs shared folders in the same way they normally connect to SMB shares. For access to a shared folder in the Dfs tree, Dfs-aware clients can issue any of the traditional methods to gain access to network resources by using Windows Explorer, File Manager, and the command line. Table 17.4 contains examples of each.

Table 17.4 Gaining Access to a Shared Folder

Method	Example
Windows Explorer	*\\Server\Dfs_root\Target_share*
File Manager (mapped drive)	*\\Server\Dfs_root*
Command prompt	**Dir** *\\Server\Dfs_root\Target_share*
Net Use command	*\\Server\Dfs_root*
Net Use command (deep root)[1]	*\\Server\Dfs_root\Target_share\Target_file*
Domain-based Dfs root in Active Directory[2]	*\\domain.xxx\Dfs_root*

[1] Supported by Windows NT 4.0 and Windows 2000 clients only.

[2] Supported by Windows 2000 clients only.

Architecture

This section describes the major components that make up Dfs and explains how they work. At the end is a list of improvements over Dfs 4.*x*.

Block Diagrams

Figure 17.2 shows the components that make up the Dfs console, service, and client in Windows 2000.

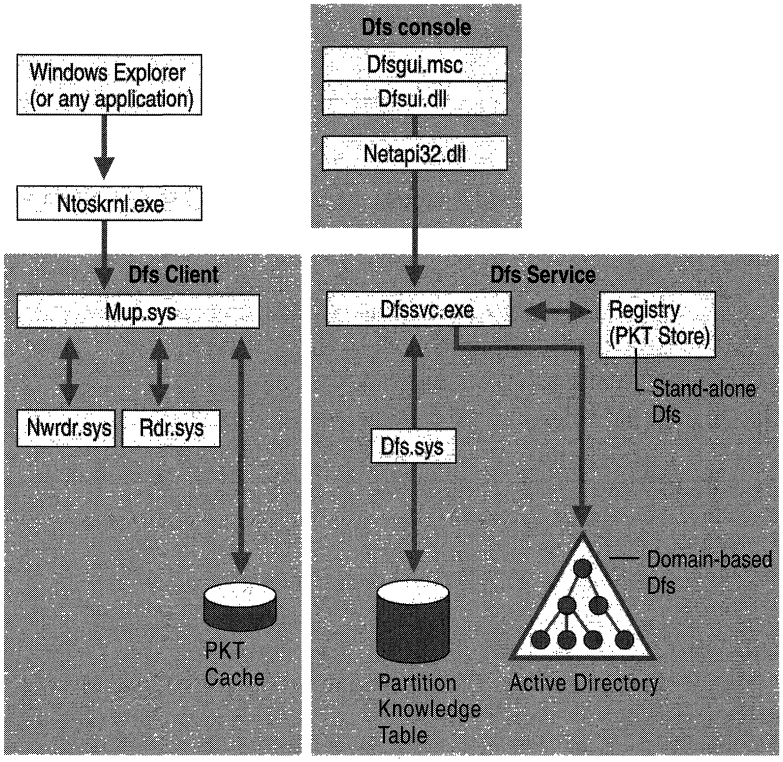


Figure 17.2 Dfs Architecture for Windows 2000

The Dfs administrative console can be located on any Windows 2000–based computer, not necessarily the same one as the server or the client. Its binary files include the following:

- Dfsgui.msc and Dfsui.dll, which implement the Dfs snap-in.
- Netapi32.dll, which contains the NetDfs*xxx* administrative functions, including the following:
  - NetDfsAdd, which adds a new Dfs link or adds a shared folder to an existing link.

- NetDfsEnum, which enumerates and returns information about all Dfs links.
- NetDfsGetInfo, which returns information about a specific Dfs link.
- NetDfsRemove, which removes a share from a link. It removes the link itself if the share is the last associated with the specified link.
- NetDfsSetInfo, which associates information with a link.
- NetDfsGetClientInfo, which returns client-cached information about a specific link.
- NetDfsSetClientInfo, which associates information with a client's cached copy of a link.

For more information about the Dfs APIs, see the MSDN Platform Software Development Kit (SDK) link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

The Dfs service resides on the Dfs server where the root is located. Its binary files include the following:

- Dfssvc.exe, which provides server-side support for NetDfs*xxx* APIs that configure and maintain the Dfs topology. Dfs service can be started or stopped by using a **net start dfs** or a **net stop dfs** command.
- Dfs.sys, which is the kernel component of the Active Directory or registry-based configuration that is maintained by Dfssvc.exe. It translates the Dfs configuration into responses for Dfs clients.

Binary files on the Dfs client include the following:

- Mup.sys, which implements Dfs client support and redirector selection. "Mup" stands for multiple UNC provider.
- Rdr.sys, which handles communications to the Dfs server and Windows-based file servers that use the Server Message Block (SMB) protocol.
- Nwrdr.sys, which handles communications to non-Windows-based file servers that use NetWare Core Protocol (NCP).

Figure 17.3 shows the components that make up the Dfs-aware client for Windows 95 or Windows 98.

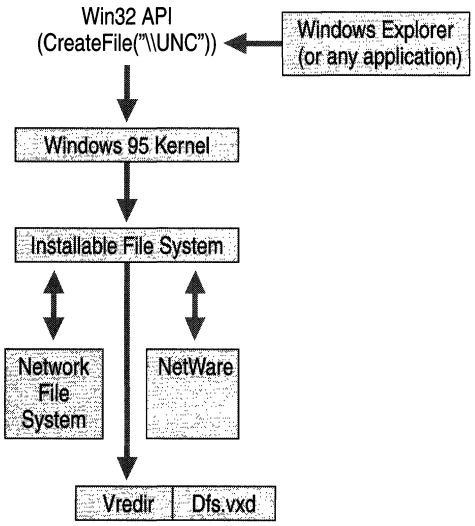


Figure 17.3 Dfs Client Architecture for Windows 95 and Windows 98

Binary files on the Dfs client for Windows 95 and Windows 98 include the following:

- Dfs.vxd, which implements Dfs client support.
- Vredir.vxd, which is the SMB redirector for Windows 95 and Windows 98.

How Dfs Works

As shown in Figure 17.4, the bulk of the Dfs code resides in four files: Netapi32.dll and Mup.sys on the client, and Dfssvc.exe and Dfs.sys on the server. Netapi32.dll contains the NetDfs*xxx* APIs that make remote administration of Dfs servers possible. The Dfs console uses remote procedure calls (RPCs) to communicate with the Dfs server. The server and the client use SMB protocol to communicate.

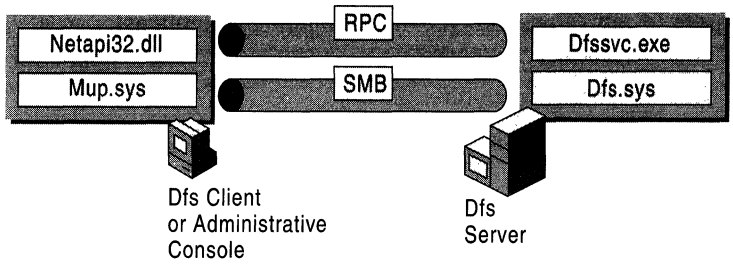


Figure 17.4 Core Dfs Files and Communication Protocols

When a user on a Dfs client requests access to a shared folder, the Dfs client intercepts the request and checks the local cache for a valid referral containing the UNC for the requested shared folder. If one is found, the user is referred to the specified shared folder transparently.

If the target shared folder has never been requested before or if the data in the cache for it has expired, the Dfs client asks the Dfs server for a referral. The Dfs server looks in the PKT and returns a referral to the client.

If the referral contains a replica set, the server uses the IP address of the client to determine the site in which the client resides. It then randomizes the list of replicas, giving preference to those located in the same site as the client. The client receives the referral and connects to the first available server in the randomly ordered list using the appropriate protocol.

The referral is stored in the local PKT cache and locked. If the TTL has not expired, the client always selects the first replica on the list. If a failover occurs, the client walks down the list for an available replica. If no replicas are available, the client gets a new replica list from the Dfs server.

Windows 2000 Improvements of Dfs 4.*x*

Dfs server enhancements include the following:

- The Dfs service is installed automatically with Windows 2000.
- The Dfs service can be paused and stopped, but not removed from the administrative console.
- Dfs is integrated into the Active Directory namespace for domain-based Dfs.
- Dfs roots hosted by more than one domain controller eliminate the root as a single point of failure.
- Support for the File Replication service to permit automatic replication of file changes between Dfs replicas.
- The Dfs administrative tool is now graphical by way of MMC.

- Status flags indicate the availability of replicas.
- Dfs links can connect to other links on other Windows 2000–based servers without a fresh referral.
- The expiration (TTL) of referrals that are cached by Dfs clients is configurable on links in the Dfs namespace.
- Dynamic configuration of the Dfs topology — that is, you do not need to restart the server when adding or removing Dfs roots.
- Support for Cluster service.

Client improvements in Dfs depend on the host platform.

Windows 95 The Dfs-aware client is available and can be downloaded. However, it can negotiate referrals only for SMB volumes. All other volumes appear as empty directories. Similarly, the **net use** command cannot be used beyond the share level.

Windows 98 The Dfs-aware client is built in, but it is subject to the same limitations as those cited for Windows 95.

Windows NT 4.0 A built-in Dfs-aware client supports connections to non-SMB volumes such as NetWare, NFS, and NCP, and allows deep **net use** commands. This is also true for Windows 2000–based clients.

Windows 2000 Built-in Dfs-aware clients select replicas based on site location. If two replicas are located in different sites, the client prefers the local replica. If the replicas are on the same site, one is chosen randomly. The clients also support links to non-SMB volumes and deep **net use** commands.

Windows 2000–based clients also contain a shell extension to Windows Explorer that lets you see all the referrals for a Dfs link, select a referral for a Dfs link, and refresh the referral cache for a Dfs link.

Scripting

A command-line program, Dfscmd.exe, can be used to modify the Dfs namespace with scripting commands. You can use Dfscmd.exe to do essentially anything you can do with the Dfs administrative tool except enable or disable FRS replication.

The syntax for Dfscmd is:

dfscmd [*options*]

Substitute **/HELP** for *options* to see the Help shown in Figure 17.5. For domain-based Dfs, *\\Dfsname\Dfsshare* means *\\Domain_name\Domain_dfs_name*. For stand-alone Dfs, *\\Dfsname\Dfsshare* means *\\Server_name\Dfs_root*.

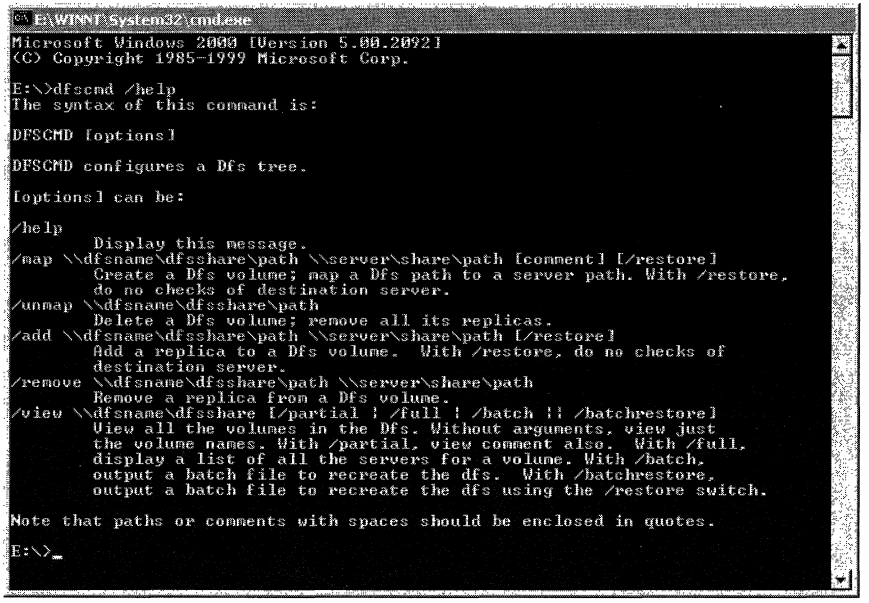


Figure 17.5 Dfscmd Help

Design Guidelines for Dfs

Modern enterprise networks present a variety of file storage and access problems for administrators and users alike. This section describes how Dfs can help solve those problems. It also discusses specific design issues and offers guidelines that must be considered when rolling out a Dfs solution.

Problems That Dfs Solves

An administrator can use Dfs to build a single hierarchical file system or a small set of them that extend across the enterprise. This logical representation of network storage allows the following:

- Use of a single namespace
- High availability
- Load sharing
- Capacity expansion
- Easier intranet and Internet publishing

Unified File System Namespace

As file servers crop up throughout the enterprise — often in a grassroots fashion — it becomes more and more difficult for users to find the information for which they are looking. Shared folders are often distributed throughout an enterprise on many file servers across a wide area network. Many organizations have hundreds of file servers throughout the enterprise. This presents the user with the problem of where to start looking for information. Because shared folders are usually associated with physical servers, the user must first determine what physical server is hosting the shared folder. For example, product information might be stored on \\Building 4\Marketing2\Prod_Info or on \\Corporate\Floor 4\Sales\Prod_Info.

An administrator can use Dfs to address this situation by consolidating a large set of physical shared folders into one or more logical namespaces. The underlying shared folders do not have to be modified in any way to work with Dfs. Dfs is a technology that changes the user's view of the physical storage. Dfs allows a unified file system namespace to be developed that masks the physical locations of the underlying shared folders from the users.

In the previous example, users might not necessarily know whether they must look on a Sales server or a Marketing server for the information they want. And even if they did, they might not know the physical name of the server. By using Dfs, an administrator can publish the \\Building 4\Marketing2\Prod_Info shared folder into a logical namespace called \\Company\ProductInformation, for example. Other entries in that logical namespace might be \\Company\Benefits, \\Company\Legal, and so forth—each of which might reside on a physically different file server or shared folder. The logical namespace can be built into a deeper hierarchy, so that \\Company\Benefits\Medical and \\Company\Benefits\Retirement might also be added—and refer to different servers or shared folders as well.

Users no longer need to know the physical location of shared folders; instead, they can browse through the logical namespace and Dfs transparently navigates them to the appropriate underlying shared folder.

Note that Dfs cannot solve all of the enterprise's naming issues nor can it catalog all corporate information. Dfs is specifically a solution that targets file systems. Other namespaces are still outside the scope of Dfs, such as public folders in Microsoft® Exchange Server client/server messaging and groupware, the Printers folder, and Active Directory. Dfs can work with some of these other technologies, but it is not meant to consolidate these disparate namespaces.

High Availability

Windows 2000 Dfs provides the ability to create multiple shared folders (called replicas) with the same logical name. This can be done at any point in the namespace hierarchy — at the root or at any point beyond. This means that when a file server or shared folder is brought down, either planned or unexpectedly, one or more other servers can still service a client's request for file data.

If a client has access to a shared folder through the logical namespace and the underlying physical resource is unavailable, Dfs automatically fails over to a replica. Dfs can be optionally configured to work with the File Replication service to maintain consistency in the data that is stored in a replica set.

Load Sharing

The ability of Dfs to support physically separate replicas with the same logical name provides a degree of load sharing. For example, suppose that \\Company\StockInfo is a heavily used sharepoint. This logical sharepoint can be associated with multiple shared folders on different computers even in different sites. Dfs allocates user requests to the underlying shared folders in a distributed manner.

Capacity Expansion

What happens today if your file server is running out of free space and the server is already at physical capacity? In most situations, you add additional disks to another server and create a new shared folder that is associated with the new storage.

As new shared folders are created to account for added physical capacity, the user has the added burden of first locating the additional physical servers and shared folders and then having to map additional drive letters to them.

You can use Dfs to expand a hierarchy beyond the physical capacity of the storage system by transparently linking to additional storage on different servers. In Dfs terminology, this is referred to as adding Dfs links. If you run out of storage on a server, you publish additional shared folders from another server into the logical namespace.

For example, suppose you have a shared folder named \\Sales\Info. Beneath the shared folder you have two subfolders, \Internal and \External. Physically, they are referred to as \\Sales\Info\Internal and \\Sales\Info\External. If the \\Sales server runs out of capacity and cannot be physically expanded, you are likely to install a new server (or use an existing one) and move some of the data to that server (called \\Sales2, for example). The result is that users would now need to know about two physical servers and shared folders to gain access to all the sales information (that is, \\Sales for internal information and \\Sales2 for external information).

You can instead use Dfs to eliminate this impact on the user. Suppose that before the server ran out of capacity, you published the \\Sales\Info shared folder into the logical namespace \\Company\Sales. Users would have access to the two folders as \\Company\Sales\Internal and \\Company\Sales\External. After the new server is installed, you can add \\Sales2\External to the logical namespace \\Company\Sales\External and leave the original link for \\Company\Sales in place.

The result is that the user still refers to the same logical namespace for these folders, but Dfs transparently connects them to different servers and shared folders. When your server ran out of space and you moved some of the storage to a different server, the user did not have to change anything.

Intranet/Internet Publishing

Another handy way to use Dfs is with Web publishing. With Internet Information Services (IIS), you usually set up a network shared folder as the location where content is to be served up by the Web server. This network shared folder is usually a physical location. Any of the content within that Web site, such as subfolders, to which access is gained in a relative manner must reside on that same physical server and shared folder. However, Web sites are usually set up where portions of the site are maintained and administered by different groups.

Dfs works well in this situation. Instead of referencing a physical shared folder as the content location to be managed by IIS, you reference a portion of the logical namespace. The nature of Dfs allows portions of the logical namespace to reside on different servers in the network. So, for example, the \\Company\Intranet namespace might map to the \\Intranet\Root shared folder, the \\Company\Intranet\Sales namespace might map to the \\Sales\Intranet shared folder, and the \\Company\Intranet\Marketing namespace might map to the \\Marketing\Intranet shared folder. Each group in the organization can maintain its own portions of the larger Web site on its own server.

In addition, the logical namespace always remains constant, so it does not matter whether a portion of the Web site content is moved to a different server or shared folder. As long as the Dfs namespace is updated to point to the moved content, nothing on the Web server has to be modified, and the links still work.

Finally, because Dfs can provide high availability and load sharing, a Web farm can be built up in which you have multiple servers that host the same content for a site. As Figure 17.6 shows, if one server fails, another can take its place without the user's knowledge.

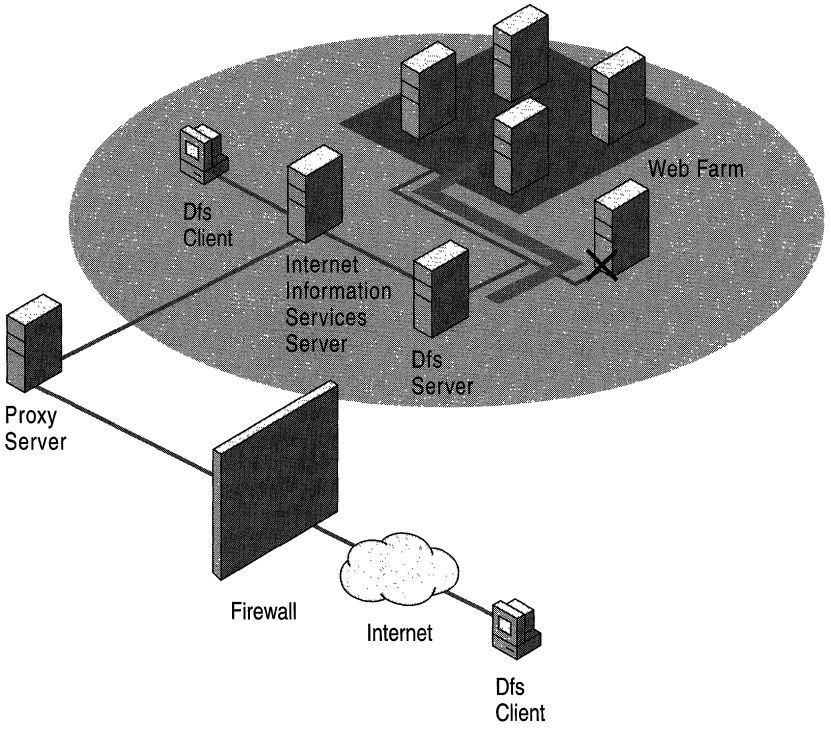


Figure 17.6 Intranet Availability

Naming Strategy

When you roll out Dfs, you have the opportunity to start anew with a consistent namespace design. It is important to develop standards for the enterprise-wide namespace, or at least for a domainwide namespace. Developing naming standards first — and making sure you adhere to the naming standards during implementation — makes using and managing the Dfs namespace easier both from a user perspective and an administrative perspective. Both of these contribute toward lower administrative costs. Even if you do not expect to deploy Dfs until a later phase of a Windows 2000 deployment, it is important to begin thinking about the namespace design early in the deployment planning process.

The ability to define Dfs namespaces at the domain level (with domain-based Dfs roots) makes it easy to publish and manage the shared folders that exist only at the server level. Even if the shared folders themselves do not follow any naming standard, they can still be abstracted into the logical Dfs namespace in a cohesive fashion. It does not matter what names the underlying shared folders have been assigned. They can still be placed into the logical Dfs namespace at an appropriate level as Dfs links and assigned names that are consistent with Dfs namespace standards. This provides the best of both worlds. It allows access to the shared folders through their legacy physical names, and it also allows access to them through their logical names in the Dfs namespace.

It is recommended that you develop Dfs namespace standards that comply with other enterprise naming standards for domains, servers, and so on. Some of the objects that can be included or referenced by the Dfs namespace standards are described in the following sections.

Domain Naming

Windows 2000 domains can be referenced by their FQDN name, such as division1.Reskit.com. They can also be referenced by a NetBIOS name, such as Division1. When you define your domain namespace, you might want to keep in mind that users view and use these names in UNCs when they gain access to a domain-based Dfs root.

Server Naming

When you develop an overall namespace design for Windows 2000, it is recommended that you also define naming standards for servers. Windows 2000–based servers can be referenced by a DNS name, such as salesfiles.division1.Reskit.com, or a NetBIOS name, such as SalesFiles.

When attempting to gain access to a stand-alone Dfs root, the user must specify the server name (for example, \\salesfiles.division1.Reskit.com or \\salesfiles). When gaining access to a domain-based Dfs root, the user *can* specify the server name. Remember that for a domain-based Dfs root, you must always gain access to the physical server name when you set up the Dfs namespace. It is, therefore, important to develop a consistent naming scheme for the file servers in your enterprise. For example, you probably would not want one server named \\SalesFiles and another named \\MktgFandPSrv.

Dfs Root Naming

The Dfs root name is significant primarily to users. It is the point beyond the server or domain name that is at the top of the hierarchy of their logical namespace. Standardized and meaningful names at this level are very important, especially if you have more than one Dfs namespace in a domain, because this is where users begin their journey into that namespace. The contents of the Dfs namespace must be as clear as possible to the users so that they do not follow the wrong path and have to backtrack.

A stand-alone Dfs root name is exactly the same as the name of the underlying physical share and, therefore, must be unique to the server. With a domain-based Dfs root, the name of the root can be different from the name of the underlying physical share and must be unique to the domain.

Dfs Link Naming

Remember that a Dfs link is essentially a logical folder within the Dfs namespace that points to a physical shared folder. In all cases, the Dfs link names are exposed to users. As with Dfs roots, it is, therefore, important to develop standardized, meaningful names for the Dfs links.

It is recommended that you keep the user perspective in mind when you develop standards for Dfs links. An important design goal is to develop a Dfs namespace that minimizes erroneous navigation within the hierarchy that is represented in the namespace. It is important to make the namespace as clear as possible at all levels in the namespace. Keep in mind that comments entered in the Dfs administrative console are not visible to users.

This is probably even more important for a Dfs namespace than for a physical namespace. This is because it might be possible for the user to jump to a shared folder on a different computer when he or she selects a Dfs link in the namespace. This might mean that a session has to be set up with that physical server (if one does not already exist), which might delay access. Therefore, you want to minimize the number of times that users traverse a wrong path. Clear and meaningful naming standards can help.

It is recommended that you also try to keep folders at the same level in the Dfs namespace consistent in context. For example, you probably would not want to have Dfs links of New York, Seattle, and Milan mixed with other links named Sales, Marketing, and Consulting.

Shared Folder Naming

Active Directory supports the concept of a shared folder that is essentially an object published in the directory that corresponds to a physical or logical share. A shared folder cannot be accessed by a UNC directly, but it can be searched for and queried by using any of the tools that are included with Active Directory.

Some organizations might find it useful to publish some common Dfs roots into Active Directory as shared folders. Folders that reside deeper in a Dfs namespace can also be published as shared folders, for example \\Company\Sales\Internal\CurrentYear. Because the shared folders are published with a freeform description, it is useful to establish naming conventions to ensure some level of consistency.

Dfs Namespace Strategy

These are some of the questions that you might ask yourself when you develop the Dfs namespace or namespaces for your organization.

Determining How Many Dfs Namespaces Are Needed

One of the first questions that needs to be addressed is how many Dfs namespaces you want to put into your domain: a single Dfs namespace or multiple Dfs namespaces. The answer depends on a number of possibilities, including who is to use the domain, what you are going to be publishing in the namespace, how deep you want to allow your namespace to extend, and so forth.

If your domain has a broad scope—geographically, organizationally, or functionally—you might want to define multiple Dfs roots. For example, you might have Users, Projects, and Software_installs. On the other hand, if the domain has a narrow scope, you might want to define a single Dfs namespace.

Determining How Deep the Dfs Namespace Needs to Extend

Remember that with Windows 2000–based clients, deep hierarchies are less of an issue because users are able to use the **net use** command to delve into the hierarchy. However, if you have a large number of Windows 95–based, Windows 98–based, and Windows NT 4.0–based clients that are going to be using the namespace, you might want to limit the depth of the hierarchy that you present to them. Remember that the path length cannot exceed 260 characters.

Determining Whether to Use a Hierarchy of Dfs Roots

Must you build larger, more extensive Dfs namespaces out of smaller, more focused Dfs namespaces? You might want to do this is if you want to present specific Dfs roots to some users as the true top of the hierarchy and also present a set of those same Dfs roots to other users as only the Dfs links in a larger hierarchy. By using a hierarchy of Dfs roots, you can scale the namespace as your organization grows and tailor the namespace for distributed management.

Deciding What Must Be in the Root of the Namespace

Consider the root of the Dfs namespace to be a launching point into the namespace — that is, a placeholder in the namespace. You probably do not want your roots to get too cluttered with files. You might place a single file in the root (a readme file) that describes the contents and purpose of the namespace. If you take this approach, the underlying shared folder that corresponds to the Dfs root is likely to be an empty directory or one with a single file in it.

Deciding Whether to Use a Domain-based Dfs Root or a Stand-alone Dfs Root

Use a domain-based Dfs root for new Dfs namespaces that you are creating in a Windows 2000 domain. Stand-alone Dfs roots lack many of the advantages offered by the domain-based Dfs roots and are provided primarily for backward compatibility.

Deciding What Type of Shares to Publish in the Dfs Namespace

Only Windows NT 4.0–based and Windows 2000–based servers can host Dfs roots for the creation of large hierarchies of Dfs namespaces. All other physical shares can be included only as Dfs links. They cannot host Dfs roots or link to other shared folders. In Windows NT 4.0, these were called leaf nodes and included shares that were published on Windows NT Workstation, Windows 95, Windows 98, Microsoft® Windows® for Workgroups, and all non-Microsoft shares for which client redirectors are available.

It is recommended that you also keep in mind that architectural differences exist between the Windows 95 or Windows 98 file subsystem and Windows NT 4.0 and Windows 2000 file subsystems. These differences do not allow a Windows 95–based or Windows 98–based client to link directly to a non-SMB share. The only way that Windows 95 or Windows 98 can link to a non-SMB share is if that share has an SMB gateway in place (for example, Gateway Service for NetWare). If you have a large number of Windows 95–based or Windows 98–based clients that need access to Dfs links and you do not have the necessary gateway, it is recommended that you restrict your shares to SMB.

Note that shares other than Windows NT 4.0 and Windows 2000 also have the limitation that they cannot participate in file replication.

Deciding What Type of File Systems to Publish in the Dfs Namespace

The underlying file system for shared folders that are published in a Dfs namespace must be NTFS to take advantage of its security features. In addition, if you are going to set up shared folders as replicas for a Dfs link and you are going to take advantage of Windows 2000 FRS, the file system must be NTFS.

Determining When to Use Replicas

It is recommended that you consider replicas for both Dfs roots and Dfs links in the following situations:

- You have critical business data that must have a high degree of availability.
- You have servers that are servicing shared folders that are overburdened, and you want to distribute the load among a set of servers.
- You want to place the underlying shared folders closer to the consumers of the data (assuming that you have defined a site topology for appropriate segments of your Windows 2000 network).

Deciding Whether to Publish All Shared Folders in a Dfs Namespace

It is recommended that you consider publishing only shared folders that are well established and not likely to be retired in the near future. If you have shared folders whose underlying physical name is dynamic in nature, include it in the Dfs namespace only if you can tolerate the added administrative overhead or develop automated scripts to update the Dfs links.

Also consider limiting the total number of links that are published within a Dfs namespace. A practical limit would be about 1,000 links per Dfs root.

Deciding What Must Be the Relationship Between Shared Folders and Dfs Links

The relationship between your shared folders and Dfs links depends on your naming strategy for shared folders and your naming strategy for Dfs links. They might end up mapping one to one, or they might not. Keep in mind that when you specify the shared folder that is mapped to a Dfs link, you can specify either the name of the actual shared folder itself or a directory that includes the shared folder level. For example, if the name of the shared folder is \\Server\Projects, you can create a link in the Dfs namespace that starts at \\Server\Projects\New.

You must be careful not to create loops in the Dfs namespace. Windows 2000 does not automatically check for loops in your namespace.

Replication Strategy

You must consider using the following two types of replication with Dfs to achieve high availability:

- Replication of Dfs knowledge (Dfs roots).
- Replication of Dfs content (replica sets).

You also need to keep in mind that site topology influences how replication occurs.

Dfs Roots

All Dfs knowledge (such as that in the PKT) for a domain-based Dfs is maintained in Active Directory. Updates to the Dfs configuration initially take place on the Windows 2000 domain controller hosting the PDC FSMO role. Domain controllers may have a different view of the Dfs configuration until multimaster replication from the PDC makes changes fully replicated between all domain controllers in a domain.

With regard to Dfs, this means that the Dfs namespace might not always be exactly the same on all domain controllers. In fact, it will not be the same when changes are made to the Dfs namespace until those changes replicate to all domain controllers. In addition, because all servers that host a domain-based Dfs root obtain their knowledge from domain controllers, there is also a period of time when the servers do not all have the same view of the Dfs namespace.

Be aware of the amount of data that must be replicated for Dfs knowledge. An entry in the PKT uses approximately 400 bytes per Dfs link. A mid-size Dfs namespace with 100 links would require approximately 40 kilobytes (KB). Any changes to that Dfs namespace causes the entire 40 KB to be replicated to all domain controllers in the domain.

Replica Sets

Multiple copies of a shared folder (or replicas) for a Dfs link can be configured with or without content replication. Even if replication is desired, it is not mandatory to use FRS to keep replicas synchronized with one another. In fact, by default, FRS is not enabled for Dfs replica sets. However, you generally do want to make sure the underlying shared folders for Dfs links are kept synchronized to present the same data to users regardless of the folder to which they want access. Microsoft strongly recommends using FRS for automatic replication of Dfs shared folders.

However, if the information in the shared folder is static, its replication can be handled with a one-time copy of data to a replica. Even if the information in the shared folder is dynamic but changes infrequently, you might want to keep the shared folders synchronized by downloading the initial copies over the network and then manually updating them with changes.

Keep in mind that with FRS, as with Active Directory because it uses multimaster replication, the files in the replica set may not always be completely synchronized. This means that one client's view of a shared folder in a Dfs namespace can be different from another client's view of the same folder. This can happen when the clients have been referred to different replicas (that is, physically different shared folders) for the same point in the logical namespace. Shared folders do become consistent with time, but you will experience temporary inconsistencies when updates are being made.

You might want to adjust the schedule and rules for what gets replicated at what times. For example, if you have little available bandwidth between the servers that host the underlying replicas, you might want to schedule replication for off-hours. With dynamic data, you always have to make the trade-off between the amount of latency that you can tolerate with keeping your replicas synchronized and the amount of bandwidth that is consumed. For more information about scheduling replication, see "File Replication Service" in this book.

Site Topology

The Active Directory Sites and Service topology comes into play in a number of ways with respect to Dfs. It is important to understand how Dfs and any dependent services, such as FRS, use the site topology, so that the enterprise requirements for Dfs are taken into account when the overall network architecture is developed.

- Domain controllers, which maintain the PKT and other Dfs namespace information, follow the site topology to replicate with one another.
- When a server that hosts a domain-based Dfs root starts up, it contacts a domain controller to obtain the PKT. The site topology is referenced to determine the closest domain controller.
- When a Dfs client contacts a server hosting a domain-based Dfs root to locate a replica, it must first contact a domain controller to get a list of servers for the Dfs root. The client then selects one of the servers from the list. The Sites and Service topology is used to contact the closest domain controller and also to find the closest Dfs root server.
- When a Dfs client selects the share for a Dfs link, a Dfs root server refers the client to a list of replicas that are mapped to the Dfs link. The list is prioritized to give preference to replicas located in the same site as the client.
- When you run the Dfs administrative console and connect to a domain controller, the site topology is referenced to find the closest domain controller.

It is recommended that you consider all of these things when you determine the placement of domain controllers, domain-based Dfs root servers, replicas, and Dfs clients with respect to one another in the Sites and Service topology. The greatest concern, of course, is to place replicas within the same sites as the primary users of their information.

Security Strategy

Dfs does not enforce access control lists (ACLs) on the Dfs namespace, Dfs roots, or Dfs links beyond what is enforced by the underlying physical storage. So when a user gains access to a file or folder through a Dfs namespace, the only ACLs that apply are those that are in place on the destination shared folder. Standard Windows 2000 share, folder, and file permissions apply at the final destination.

If a user does not have sufficient permissions for a shared folder at an intermediate point in the Dfs namespace, that folder appears empty to the user.

Because permissions are applied on the underlying shares, files, and folders, Windows 2000 guidelines must be used to apply the ACLs. In most cases, avoid using individual accounts and use Windows 2000 groups instead to specify the ACLs.

Remember that if you have replicas set up for a Dfs link, you must ensure that the ACLs that are applied to the physical resources are the same for all replicas. If you use FRS to synchronize replica content, the file ACLs are replicated automatically as well.

As an administrator, you must be a member of the Domain Admins global group to administer a domain-based Dfs namespace. For a stand-alone Dfs namespace, you need only be an administrator for the server hosting it. There is no finer level of granularity for administering Dfs. Note that to add a shared folder to a Dfs link in a Dfs namespace, you do not need explicit permissions to that shared folder. Of course, any users who want access to that replica would require the necessary permissions.

Because a domain-based Dfs namespace exists at a domain level and because you must be a member of the Domain Admins group to administer a Dfs namespace, Dfs administration falls into a model of centralized administration. However, even if Dfs namespaces are centrally administered, the actual shared folders can be administered in a manner that is appropriate for the enterprise. Dfs administration does not affect the administrative model for the underlying shared folders. Centralized administration of the shared folders and distributed administration of shared folders work equally well.

Migration Strategy

Dfs is by no means an all-or-nothing deployment You can implement it incrementally until you are familiar with its operation.

Existing Windows NT Shared Folders

As you begin to deploy Dfs into an organization, you can incorporate existing shared folders into your Dfs namespace or namespaces at a pace that makes sense for the organization. Dfs provides an alternate logical view of the physical topology, but that topology exists with or without Dfs. Thus, legacy names can still be used to gain access to shared folders even when they are published to the Dfs namespace.

An organization can choose to publish as much or as little of its physical storage in the Dfs namespace as it thinks is appropriate. The Dfs namespace strategy described earlier dictates the decision of what to publish in the Dfs namespace. A reasonable schedule can then be established for publishing shared folders to the Dfs namespace.

Dfs 4.*x*

There are no tools to migrate a Dfs 4.*x* implementation to a Windows 2000 domain-based Dfs root. During an upgrade of Windows NT 4.0 to Windows 2000, any Dfs 4.*x* roots are converted automatically to Windows 2000 stand-alone Dfs roots. Note that Dfs 4.*x* implementations can be administered with the Windows 2000 Dfs administrative console.

However, an organization might be motivated to migrate its legacy-based stand-alone Dfs roots to domain-based Dfs roots to take advantage of file replication, load sharing, Active Directory integration, and so on. Again, you can choose to migrate your organization's stand-alone Dfs roots at a pace with which you are comfortable. Some roots can remain as stand-alone; others can be migrated to domain-based. Both can coexist in a Windows 2000 domain.

Platform Interoperability

Compatibility between different computers that host Dfs roots, clients, and shared folders depends on the operating system, its version, and whether Dfs is stand-alone or domain-based. Table 17.5 summarizes this information for Windows 2000.

Table 17.5 Platform Interoperability

Platform	Host Dfs cClient?	Host Dfs rRoot?	Act as Dfs tTarget?
UNIX or NetWare file servers	No	No	Yes, with NCP for NetWare and NFS for UNIX
MS-DOS, Microsoft® Windows 3.*x*, Windows for Workgroups	No	No	Yes, with SMB support
Windows 95	Yes, download client for Dfs 4.*x* and Dfs 5.0[1]	No	Yes
Windows 98	Yes, client for Dfs 4.*x* and stand-alone Dfs 5.0 included; download client for domain-based Dfs 5.0[1]	No	Yes
Windows NT 4.0 with Service Pack 3	Yes, client for Dfs 4.*x* and stand-alone Dfs 5.0 included No, client for domain-based Dfs 5.0 to be included in future service pack	Yes, stand-alone server only	Yes
Windows 2000	Yes, client for Dfs 5.0 included	Yes, stand-alone and domain-based server or domain controller	Yes

[1] The Dfs 5.0 client for Windows 95 and Windows 98 is included in the Active Directory Client Pack for Windows 95/98 (often abbreviated as DSClient Pack for Win9*x*).

Disaster Recovery Strategy

If you implement the root and file replication capabilities provided by Dfs, you can protect yourself against the loss of the server hosting a Dfs root or a server that contains the shared folders themselves. However, this does not eliminate the need to back up at least one copy of all your data. Remember that automatic replication can propagate corrupted information as well as valid data. When you repair or replace the server that has failed, you might have to restore the Dfs topology or shared data stored on it.

Therefore, it is important to plan for disaster recovery of the Dfs namespace. As noted earlier in "Replication Strategy," there are two areas of concern with Dfs:

- Saving your data — that is, the shared folders in the Dfs namespace.
- Saving the Dfs topology.
  - For stand-alone Dfs, this means backing up the Dfs configuration stored in the registry.
  - For domain-based Dfs, this means backing up the Dfs configuration stored in Active Directory and in the registry of each server.

For more information about backing up and restoring replicated files, see "File Replication Service" in this book.

For more information about backing up and restoring Active Directory, see "Active Directory Backup and Restore" in this book.

For more information about system recovery, see "Repair, Recovery, and Restore" in the *Microsoft® Windows® 2000 Server Resource Kit Server Operations Guide.*

Note The dfscmd command contains a BATCH option that can be used to back up and restore the Dfs topology you have configured on a root server to a floppy disk. For more information, see "Scripting" earlier in this chapter.

Implementing Dfs

Rolling out Dfs requires that you evaluate its ramifications on the other features of Windows 2000:

- Active Directory
- Load sharing
- File replication
- Windows Clustering

First, however, it is worthwhile to take a look at Dfs server and client setup.

Setup Considerations

System requirements for the Dfs server or for a Dfs client are limited to the amount of memory consumed by the PKT on each of them.

Dfs Server

Dfs service is a core Windows 2000 service that is installed automatically on Windows 2000–based servers. Any server that participates as a Dfs root (either stand-alone or domain-based) must be running Dfs service. All Windows 2000–based domain controllers must also be running Dfs service. The servers that run Dfs service can be domain controllers or member servers such as workgroup servers in the domain.

Dfs service is responsible for maintaining an up-to-date version of the PKT and for handing out referrals to clients that map the logical namespace to physical topology. A server that hosts a stand-alone Dfs root is completely responsible for maintaining the one and only copy of the PKT for the Dfs namespace.

A server that hosts a domain-based Dfs root uses a local copy of the PKT for the Dfs namespace that it obtains from Active Directory. The server synchronizes itself with Active Directory in two ways:

- When the server starts up or the Dfs service is restarted, the server goes to a domain controller in the domain and obtains the PKT.
- When changes to Dfs are made in Active Directory, all servers that participate in the domain-based Dfs receive the changes when Active Directory propagates the changes throughout the domain.

Dfs Client

For a client to take advantage of Dfs, specific extensions must be added to the file subsystems of the underlying operating systems. The availability and capabilities of these Dfs-aware extensions are as follows:

Windows 2000 Ships in the box and provides full access to stand-alone and domain-based Dfs roots. This includes a shell extension to view the underlying DNS-to-UNC mappings for shared folders in a Dfs namespace.

Windows NT 4.0 Workstation/Server Included in the box with Service Pack 3 to provide access to stand-alone Dfs roots.

Windows 98 Included in the box to provide access to stand-alone Dfs roots. Download the Active Directory Client Pack for Windows 95 or Windows 98 from http://www.microsoft.com for domain-based Dfs.

Windows 95 Download the Active Directory Client Pack for Windows 95 or Windows 98 from http://www.microsoft.com.

TTL Guidelines

Dfs clients cache referrals from the PKT for TTL minutes unless access to the cached Dfs path is gained before the TTL expires or before the client is restarted. Restarting the client removes all PKT caching information and is the only way of forcing Windows NT 4.0–based clients, Windows 98–based clients, and Windows 95–based clients to re-randomize their selected alternates except waiting for the TTL to expire. Windows 2000–based clients can use the **Clear History** button in the Dfs property sheet of Windows Explorer to flush the cache.

The referral for a Dfs shared folder never expires as long as the Dfs client continues to gain access to the shared folder before the TTL expires. Windows NT 4.0 used a TTL of 7 days. Administrators may use the Dfs administrative console to define the TTL for each Dfs root or link. The default TTL of 5 minutes ensures that new servers are readily found.

Setting the TTL value too short decreases any caching benefits on the part of the client and increases the number of connection attempts to the Dfs server, but ensures an accurate status of available shared folders and root and alternate servers. Increasing the TTL reduces network traffic and queries to the Dfs root, but leaves newly added servers underused until the client requests a new PKT that contains the changes for the Dfs tree.

If you set the TTL expiration too long in a multiserver environment, you risk obsolete referrals. Suppose that you set the TTL for a month, but your clients gain access to the shared folders at least once a week. In this case, the client would never refresh its cache and learn about changes to the Dfs namespace. If you only have a single server, a long TTL is acceptable because, if the server becomes unavailable, all clients refresh their caches automatically to gain access to a replacement server.

Consider the number and stability of the servers being used, how dynamic the data is, and how often clients want or need access to the data. For shares that are represented by a single shared folder, such as home directories, long TTLs on the order of two to seven days might be appropriate. If a single replica represents a volume, the Dfs client's failover mechanism locates the replacement server without interruption to the client. For an environment where new shares are being added or removed constantly, such as a large-scale development project that contains daily builds of an application to which hundreds of users have access, consider using TTLs in the 30-minute to 3-hour range.

▶ **To change a TTL value for a Dfs link**

1. Start the Dfs administrative console.
2. Expand the Dfs tree in the left pane, and right-click the Dfs link that is being changed.
3. Click **Properties** on the resulting pull-down menu.
4. On the **General** property page, change the referral cache time.

Autodisconnect Guidelines

Use long autodisconnect values for paths that are used several times a day, such as home directories. Use shorter autodisconnect times for infrequently used data or paths that represent install servers, where file copies or application installs might last from 10 to 60 minutes. Take into account the additional roles of the server (for example, file and printer sharing, Web server, application server) in addition to its role of hosting the Dfs root. Here you are trading off increased network traffic that is a result of establishing sessions for maintaining an increased number of sessions on the server.

For more information about how Autodisconnect works, see the Microsoft TechNet link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. Search the Technical Support section of this site for Knowledge Base articles and other sources of technical information.

Dfs and Active Directory

Dfs uses the distributed systems infrastructure of Windows 2000 to find and manage file resources that are located on different computers in an enterprise. In fact, it is Dfs acting in concert with Active Directory, the directory service for Windows 2000, that provides the foundation of this distributed systems infrastructure.

Active Directory maintains all knowledge of the domain-based Dfs roots that exist in the domain. Specifically, domain-based Dfs depends on Active Directory for the following functionality:

- Storing complete knowledge of the Dfs topology underlying the Dfs namespace.
- Updating and querying Dfs knowledge by using the LDAP protocol.
- Keeping all participating computers for a Dfs root synchronized in their view of the Dfs topology.
- Exposing the site topology for intelligent replica selection.
- Keeping the content of file replicas synchronized with one another by using the File Replication service.

Active Directory does not maintain any information or provide any functionality for a stand-alone Dfs root.

You can replicate Dfs roots on servers in a domain, whether or not you have multiple domain controllers in your domain. Of course, it always makes sense to have at least one replica for a domain controller in a domain. In smaller organizations or remote offices, where domain controllers might be providing multiple services, it often makes sense to host Dfs roots on the domain controllers themselves.

Dfs and Load Sharing

Dfs load sharing occurs at the client level that is opposite Windows 2000 Network Load Balancing (formerly called Windows Load Balancing Service or WLBS) that takes place on the server side. Dfs takes advantage of the share redundancy provided by replica sets to distribute demand.

The root and child nodes of a Dfs link must be backed by more than one physical server when using Load Balancing. Dfs provides a degree of load balancing because clients randomly select a physical server to connect to from the list of replicas returned by the Dfs server. However, Dfs *does not*:

- Take into account the number of client sessions maintained by a replica.
- Take into account the length of sessions maintained by clients.
- Use a DNS-style round-robin selection criteria.

If a root or link appears to be overused, you can flush the PKT for a Windows 2000–based client by using Dfsutil. This forces the client to request a new referral. For clients with operating systems other than Windows 2000, you must restart the client to flush its cache.

For more information about how Windows 2000–based clients use the referrals from the Dfs server to randomize replica selection, see "How Dfs Works" earlier in this chapter.

Revision Levels and Load Sharing

Dfs clients and servers come in two versions. Clients and servers that are running Windows NT 4.0, Windows 98, and Windows 95 talk revision level 2. Clients and servers that are running Windows 2000 talk revision level 3. Clients and servers negotiate and converse at the highest common Dfs revision level. You can see Dfs revision levels in network packets by using Network Monitor.

This relates to load sharing in that Dfs servers send the referral list (the names of the physical shares underlying the Dfs namespace requested) to revision level 2 clients in a fixed order, relying on the client to randomize the list in a unique order.

When a revision level 3 client accesses the Dfs namespace, the Dfs server performs the randomization process. The referral list is split into two parts, one composed of replicas in the same site as the client and one containing out-of-site replicas. Both halves of the list are shuffled into random order by the server.

Dfs and File Replication Service

Domain-based Dfs uses FRS to replicate content between Windows 2000–based servers hosting Dfs roots or replica sets. This includes any shared files and folders located at the Dfs root itself. It is important to keep all replicas synchronized on all servers, either manually or automatically, using FRS.

Remember that with Dfs you can specify duplicate shared folders for the same link in the Dfs logical namespace to provide high availability. However, Dfs itself makes no assumption about whether the data in the underlying shared folders is synchronized. If the data is completely static, then it can be published to the shared folders once and be left alone. However, if the data in the underlying shared folders is dynamic, something must be done to synchronize that data in all replicas. FRS is used for precisely that purpose.

For more information about how FRS works, see "File Replication Service" in this book.

Dfs and Cluster Service

If you want to provide the high available functionality offered by the Cluster service for a Dfs tree, you can define a cluster file share resource in a cluster group containing a virtual server (network name and IP address) as a Dfs root. This automatically creates the Dfs root using the network name as the server name in the Dfs namespace. The server service in Windows 2000 still announces file and print shares on all Network names defined on the system; so although the cluster group is online on one of the nodes in a server cluster, the share is accessible by the virtual server name and the local computer name. If a user connects to the local computer name and the cluster group moves to another node in the cluster, the share becomes inaccessible to the user.

For more information about using Dfs on a server cluster, see the Microsoft TechNet link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. Search the Technical Support section of this site for Knowledge Base articles and other sources of technical information.

Supporting Dfs

This section focuses on supporting distributed file systems. Activities such as monitoring Dfs activity, maintaining the Dfs configuration, and troubleshooting Dfs problems are covered.

Because domain-based Dfs relies on FRS for file and root replication, you can use FRS support methods to resolve Dfs problems. For example, with FRS you can:

- Use Network Monitor to view counters associated with FRS performance objects.
- Use Event Viewer to filter and observe FRS activity.
- Use FRS logging to analyze Dfs problems.

For more information about the File Replication service, see "File Replication Service" in this book.

Monitoring Dfs Activity

You can use Network Monitor to examine Dfs activity. To do this, you have only to capture the network data and then create a display filter to analyze it.

▶ **To create a Dfs-oriented display filter in Network Monitor**

1. Start Network Monitor and view the captured file.
2. On the **Display** menu, click **Filter**.
3. In the Display Filter window, double-click the **Protocol == Any** expression.
4. In the Expression window, click the **Property** tab.
5. Locate and expand the **SMB** container in the left pane.

   Dfs-specific flags include DFS Server Function, DFS Server Function Flags, DFS Server Type, and DFS Servicesite GUID.
6. Again, under SMB, highlight **flags2** in the left pane and **includes** in the **Relation** fields. Values of interest include **No DFS namespace** and **Filenames use DFS namespace**.

For more information about using Network Monitor, see "Monitoring Network Performance" in *Server Operations Guide*.

Maintaining the Dfs Configuration

By using the Dfs administrative console, you can manage multiple domain-based and stand-alone Dfs roots from the same console. After you have established a console for the Dfs namespaces for which you are responsible, save the console for future use.

Checking Shared Folder Status

As you begin to build up an extensive, distributed Dfs namespace, be aware that, over time, some of the underlying shared folders might be retired or their server or folder names altered. If the Dfs link is not modified to reflect these changes, the result is references in the system to nonexistent shared folders.

With the Dfs administrative snap-in, you can check the status of individual shared folders as well as check the links that specify them in the Dfs namespace. Essentially, you can verify that Dfs can see both the server and the shared folders and that the shared folders are valid. Your operational plan for Dfs must include the periodic running of the **Check Status** function on frequently used shared folders.

Table 17.6 is a list of the possible status indicators for Dfs links and shared folders.

Table 17.6 Dfs Status Flags

Link	Description
Folder icon (yellow)	Default; no known status.
Folder icon with red "X"	Link cannot be negotiated because of a bad link or lack of transport.
Folder icon with green check mark	Link can be negotiated to all shared folders.
Folder icon with blue question mark (?)	Link can be negotiated but not to all shared folders.
Shared folder	**Description**
Folder icon (yellow)	Default; no known status.
Folder icon with red "X"	Shared folder cannot be found because the share or its server is offline or unreachable as a result of a bad link or no transport.
Folder icon with green check mark	Shared folder found.

Taking Resources Offline

By using the Dfs administrative tool, the administrator can temporarily take a replica offline, to perform maintenance, for example. When a replica is offline, the Dfs server does not hand out the shared folder during the referral process. It is a good idea to keep the replica offline until FRS or another means of replication has reproduced the changes in all linked replicas.

You might want to be aware of what is going on behind the scenes and how a client finds out about a replica in the first place. When a replica is taken offline, the Dfs knowledge in Active Directory is updated. There is some latency before this information has replicated to all of the domain controllers. The servers that host the Dfs namespace are then notified about the updates to the Dfs knowledge. Finally, the Dfs client itself might already have this Dfs link cached locally. The client does not go back to the Dfs server for a new referral (which includes the knowledge that a server is offline) until its referral cache time-out for that Dfs link has expired.

Depending on all these factors, by the time a client has learned that a replica has been brought offline, it might already have been brought back online. Therefore, in some situations, the offline feature might be useful only when a replica is taken offline for an extended period of time.

Note that, even if a shared folder was taken offline as described previously, but the client did not find out about it, the Dfs client still handles the situation gracefully. When the client next tries to gain access to the underlying shared folder, it times out and then selects another replica from the referral list.

Removing Dfs

Windows 2000–based servers and domain controllers store Dfs configuration information in the registry and Active Directory. In some situations, it might be useful to return to a known state. To do this, you must delete Dfs configuration data in the registry and Active Directory.

Caution Do not use a registry editor to edit the registry directly unless you have no alternate. The registry editors bypass the standard safeguards provided by administrative tools. These safeguards prevent you from entering conflicting settings that might degrade performance or damage your system. Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. To configure or customize Windows 2000, use the programs in Control Panel or Microsoft Management Console (MMC) whenever possible.

Stand-alone Dfs Configuration

If a stand-alone Dfs root configuration is damaged and you are unable to stop hosting a Dfs namespace using the Dfs administrator tool, you can delete the Dfs configuration on this computer with the following procedure:

▶ **To remove a stand-alone Dfs root**

1. On the **Start** menu, click **Run**.
2. Type **cmd** and then type **net stop dfs**.
3. Start a registry editor (either Regedit.exe or Regedt32.exe). (For more information about these editors, see Windows 2000 Server Help.)
4. In Regedit or Regedt32, go to HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft.
5. Delete the DFSHost subkey. (The subkeys it contains are also deleted.)
6. Go to HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet \Services\DFSDriver\LocalVolumes. Delete any subkey in the LocalVolumes subkey. *Do not* delete the LocalVolumes subkey.
7. Close the registry editor, and use Computer Management to restart the Dfs service.

Domain-based Dfs Configuration

If a domain-based Dfs root configuration is damaged and you are unable to stop hosting a Dfs namespace by using the Dfs administrative tool, you can delete the Dfs configuration on this computer and from Active Directory by using the following procedure:

▶ **To remove a domain-based Dfs root**

1. Perform the steps listed in the preceding procedure ("To remove a stand-alone Dfs root") without restarting Dfs.
2. In the **Active Directory Users and Computers** console, on the **View** menu, click **Advanced Features**.
3. Under the System folder, open the Dfs-configuration container.
4. Delete the Dfs root in the right pane.
5. Restart Dfs by using the Dfs administrative tool.

Configuration changes are immediately effective for the computer on which the changes are made. For another computer in the domain, you must wait for replication to occur to see the changes or force replication by restarting the other computer.

Troubleshooting Dfs Problems

Most Dfs problems can be divided into the following categories:

- Access to the Dfs namespace
- Finding shared folders
- Access to Dfs links and shared folders
- Security-related issues
- Replication latency

Note Dfs and FRS are closely intertwined. Problems with fault-tolerant Dfs functions are often FRS replication problems. For more information about solving FRS problems, see "File Replication Service" in this book.

Gaining Access to the Dfs Namespace

If a Dfs namespace is not accessible, check the following:

- Make sure that the Dfs service is running on all domain controllers.
- Make sure that the Dfs service is running on all servers that are hosting the Dfs namespace. (A server that "hosts" the Dfs namespace contains the Dfs root or a replica of it.)
- Make sure that the Net Logon Service is running on all servers that are hosting the Dfs namespace.
- Run the Active Directory administrative console to see if Active Directory is readable. Verify that an entry shows up in the Dfs-Configuration container under Users and Computers that corresponds to the Dfs namespace. The data is stored as a "blob," so you cannot obtain any further details by using the administrative console.
- Run the Dfs administrative console, specify **Connect to a Dfs Root**, and verify the configuration associated with a selected root. If you can connect to a Dfs root with the administrative tool, this also confirms that you are able to retrieve data from Active Directory.

Tracking Shared Folders

It is recommended that one of the first things that you determine when tracking an access-related issue with Dfs is the name of the underlying shared folder that the client has been referred to. In Windows 2000, there is a shell extension to Windows Explorer for precisely this purpose. When you right-click a folder that is in the Dfs namespace, there is a **Dfs** tab available in the Properties window. From the **Dfs** tab, you can see which shared folder you are referencing for the Dfs link. In addition, you can see the list of replicas that refer to the Dfs link, so you can disconnect from one replica and select another. Finally, you can also refresh the referral cache for the specified Dfs link. This makes the client obtain a new referral for the link from the Dfs server.

The **Dfs** tab is available only to Windows 2000–based clients. There are no client-side or server-side tools available for clients using earlier versions of Windows that readily provide the name of the shared folder to which they have been referred. To mount the drive containing the Dfs root on a local computer, enter the following on the command line:

net use * *domain_name**root_name*

Keep in mind that at the protocol level, on the wire, the client still connects to and communicates with the server hosting the shared folder as if the client had gained access to the folder directly. A network sniffing tool such as Network Monitor can still be used to capture communications between the client and other Dfs components. For more information, see "Monitoring Dfs" earlier in this chapter.

Gaining Access to Dfs Links and Shared Folders

If a client cannot gain access to a shared folder specified by a Dfs link, check the following:

- Use the Dfs administrative tool to identify the underlying shared folder.
- Check status to confirm that the Dfs link and the shared folder (or replica set) to which it points are valid. For more information, see "Checking Shared Folder Status" earlier in this chapter.
- The user should go to the Windows Explorer Dfs property page to determine the actual shared folder that he or she is attempting to connect to. For more information, see "Tracking Shared Folders" earlier in this chapter.
- The user should attempt to connect to the shared folder directly by way of the physical namespace. By using a command such as **ping**, **net view**, or **net use**, you can establish connectivity with the target computer and shared folder.
- If the Dfs link has a replica set configured, then be aware of the latency involved in content replication. Files and folders that have been modified on one replica might not yet have replicated to other replicas.

Security-Related Issues

If a client is experiencing security-related problems with a Dfs link, check the following:

- First, identify the actual shared folder to which the client is gaining access. See "Tracking Shared Folders" earlier in this chapter.
- Be aware that users might not see objects at intermediate points in the Dfs namespace if they do not have access to them.
- Because ACLs are applied to the actual shared folders, verify client connectivity to the physical shared folder and troubleshoot the security access problem as if the user were gaining access to the folder through the physical namespace (that is, confirm User account, check ACLs on the shared folder and files, check group membership, and so forth).
- If the Dfs link has a replica set configured, be aware of the latency involved in content replication. ACLs might have been modified on one replica but not yet been replicated to other replicas.

Replication Latency

Be aware of the latency issues already discussed regarding replication of Dfs namespace knowledge. Because the topology knowledge is stored in the domain's Active Directory, there is some latency before any modification to the Dfs namespace is replicated to all domain controllers.

From an administrator's perspective, remember that the Dfs administrative console connects directly to a domain controller. Therefore, the information that you see on one Dfs administrative console might not be identical with the information about another Dfs administrative console (which might be obtaining its information from a different domain controller).

From a client's perspective, you have the additional possibility that the client itself might have cached the information before it was modified. So, even though the information about the modification might have replicated to all the domain controllers, and even if the Dfs servers have obtained updates about the modification, the client might still be using an older cached copy. The ability to manually flush the cache before the referral time-out has expired, which is done from the **Dfs** tab in the **Properties** window in Windows Explorer, can be useful in this situation.

Dfs Utilities

You can use the Dfsutil tool on a Dfs server to do the following:

- List the Dfs roots in a domain.
- Check the validity of a Dfs topology in a domain.
- View a Dfs topology in a domain.
- Re-initialize a Dfs root (that is, stop and restart it).
- Identify the type of Dfs root (stand-alone or domain-based).
- Resolve the UNC path to a shared folder.

On a Dfs client, you can use the Dfsutil tool to do the following:

- Flush the local PKT.
- Flush a local PKT table entry.
- Flush one SPC table entry.
- Dump the local PKT.
- Dump the SPC table.
- Make the client reread the registry.

The syntax for Dfsutil is:

dfsutil [*/options*]

Substitute **/HELP** for *options* to see the Help shown in Figure 17.7.

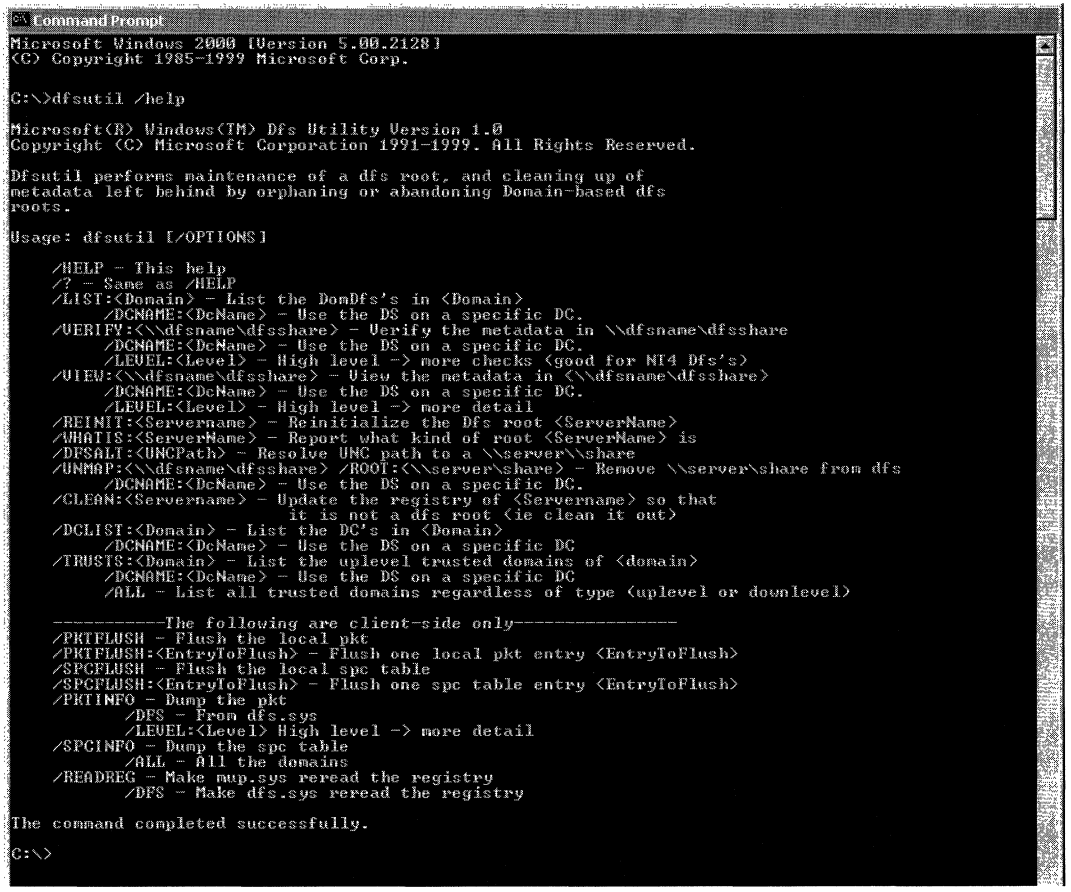


Figure 17.7 Dfsutil Help

To use Dfsutil, install the support tools that are located in the Support\Tools folder on the Windows 2000 Server installation CD. To install the tools, double-click the **Setup** icon in that folder.

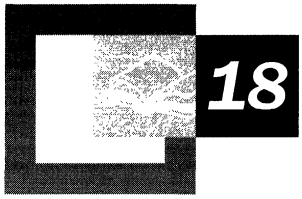
For more information about using Dfsutil, see Windows 2000 Support Tools Help.

Additional Information

For more information about the Dfs APIs, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

CHAPTER 18

File Replication Service



Microsoft® Windows® 2000 Server uses the File Replication service (FRS) to replicate system policies and logon scripts stored in System Volume (SYSVOL). Each domain controller keeps a copy of SYSVOL for network clients to access. FRS can also replicate data for Distributed file system (Dfs), synchronizing the content of each member in a replica set defined by Dfs. FRS can copy and maintain shared files and folders on multiple servers simultaneously. When changes occur, content is synchronized immediately within sites and by schedule between sites.

In This Chapter

Related Information in the Resource Kit

- For more information about distributed file systems, see "Distributed File System" in this book.
- For more information about directory replication, see "Active Directory Replication" in this book.

Introduction to FRS

File Replication service is a multithreaded replication engine that replaces the LMRepl service that is used in Microsoft® Windows NT®. Multithreaded means that several processes can run at the same time to handle multiple tasks. This allows FRS to replicate different files between different computers simultaneously.

FRS does not guarantee the order in which files arrive. Files begin replication in sequential order based on when the files are closed, but file size and link speed determine the order of completion. Because FRS replicates only whole files, the entire file is replicated even if you change only a single byte in the file.

FRS expands on the functionality provided by LMRepl with the following enhancements:

- Multimaster replication of files and folders for allowing updates to occur independently on any server in the domain.
- Site-aware clients (Microsoft® Windows® 2000, Microsoft® Windows NT® version 4.0, Microsoft® Windows® 95, and Microsoft® Windows 98 with the Active Directory add-on) for locating nearby servers hosting SYSVOL and Dfs content.
- Configurable schedules for replicating Dfs and SYSVOL content between sites.
- Automatic replication of folder and file attributes including ACLs.

Like LMRepl, FRS is automatically installed on Windows 2000 domain controllers and configured to start automatically. For member servers, the service start value is initially set to manual.

Note There is no administrative console for FRS. SYSVOL replication occurs automatically just like directory replication. Replication of Dfs files and folders is controlled by the Dfs administrative snap-in.

Although Active Directory replication and File Replication service are separate mechanisms, they are conceptually similar. Therefore, it can be useful to read about directory replication when you are learning about FRS. For information about directory replication, see "Active Directory Replication" in this book.

Key Terms

FRS terms that you need to know before reading further are:

Replication. The process of copying data, from one computer to another, that converges to an identical data set over time. Replication enhances availability and file sharing by duplicating shared files.

Replica. A member of a replica set that contains a copy of a shared folder or file.

Replica set. Two or more copies of a shared folder that participate in replication. Each copy must be located on a different computer.

Initial master. First member in a replica set that is the starting point for automatic replication. This means the files and folders in that replica are replicated to other replicas for the first replication cycle.

Replicating SYSVOL

The Windows 2000 System Volume, or SYSVOL, is built during the creation of a domain controller by Dcpromo.exe. It is a tree of folders containing files that need to be available and synchronized between domain controllers in a domain or forest, including:

- SYSVOL share.
- NETLOGON share.
- Windows 95, Windows 98, and Windows NT 4.0 system policies.
- Windows 2000 Group Policy settings.
- User logon and logoff scripts.

For example, the default folder structure contains the following folders for policies or scripts used by network clients:

\\Winnt\Sysvol\Sysvol*domain_name*\Policies

\\Winnt\Sysvol\Sysvol*domain_name*\Scripts

When you add, remove, or modify the contents of the Sysvol folder on a domain controller, those changes are replicated to the Sysvol folders on all other domain controllers in the domain.

FRS uses the same connection objects as the Active Directory™ directory service when it replicates SYSVOL content. Therefore, it uses the same schedule as Active Directory for intersite replication. However, unlike Active Directory, replicated content between sites is not compressed.

Tip A handy method for checking SYSVOL replication is to create a file named after the originating computer such as \\Winnt\Sysvol\Sysvol*domain_name**file_name_equals_server_name*. Then observe which other domain controllers receive the new file. Updates to servers in remote sites are governed by a schedule.

Replicating Dfs Replicas

Unlike SYSVOL replication, which is implicitly enabled, replication for Dfs replicas must be explicitly enabled by using the Dfs administrative console. Only domain-based Dfs can use FRS; stand-alone Dfs does not support automatic file replication. Remember that FRS is installed only on Windows 2000 servers and that the service starts automatically only on domain controllers. To start FRS on a member server, use Control Panel. Double-click **Administrative Tools**, double-click **Services**, and then use the menu options in the Services console.

▶ To enable or disable FRS replication

1. Open the Dfs administrative console.
2. In the left pane, right-click the Dfs link for the replica set, and then click **Replication Policy**.
3. In the **Replication Policy** dialog box, click **Enable** and **Disable** as needed.

   If replication is being enabled for the first time, the files and folders on the first server enabled (the initial master) become authoritative. This means those files and folders are duplicated to other replicas for the first replication cycle before multimaster replication takes effect.

Replication is not allowed, that is, the shared folder appears as N/A, under the following conditions:

- A shared folder on a computer where FRS is not installed.
- A shared folder that is not on the version of NTFS used in Windows 2000.
- A shared folder that uses a cluster name in its path name. For more information about using file replication with Cluster service, see "Distributed File System" in this book.

- A shared folder on a computer that does not belong to a Windows 2000 domain.
- A shared folder on a computer whose domain is inaccessible by the user who is currently logged on.

In addition, the following events do not trigger replication:

- Changes to a file or folder's last access time.
- Changes to a file or folder's archive bit.

For a complete description of Dfs, see "Distributed File System" in this book.

How FRS Works

FRS provides redundancy for the content of designated NTFS shares between Windows 2000 servers. The servers can be interconnected in any topology such as a ring or a star configuration. With an appropriate topology and sufficient network support, hundreds of computers can replicate the same set of files or folders. Conversely, one computer can be a member of multiple replica sets.

FRS also provides redundancy for SYSVOL and Dfs distribution by way of multiple distribution paths between the replicas in a replica set. If one replica is down, data flows using a different route. Dampening logic prevents a data file from being sent more than once to any particular replica.

Multimaster replication allows any domain controller or member server to propagate changes to replicated files and folders on any other domain controller or member server. There are no primary/secondary or master/slave relationships. When a replicated file is changed and closed, FRS submits that change to other members in the replica set. Those members decide whether to accept or reject the change according to certain criteria.

For example, suppose a replica set is composed of Computer A and Computer B. If File X on Computer A is updated and then closed, FRS notifies Computer B of the change.

- If the event time associated with Computer A is more than 30 minutes later than the event time for Computer B, the change is accepted immediately and Computer B is updated. If the event time associated with File X on Computer A is more than 30 minutes earlier than the event time for File X on Computer B, the change is rejected.

  An event time is associated with any change to a file. It denotes when the file was closed after being changed or was last replicated. The default event time window is 30 minutes.

- If the event time of the change in the file on Computer A is within 30 minutes of the event time for the version of the file on Computer B, FRS checks the version numbers of the file. If the version number of the file on Computer A is greater than that of the file on Computer B, the change is accepted and the file is updated. If the version number of the file on Computer A is less than that of the file on Computer B, the change is rejected.

  The version number is a numerical value FRS uses to track changes occurring to a replicated file. It is assigned by a counting mechanism similar to the update sequence number (USN) used by Active Directory. When a changed file is closed, its version number increments by one.
- If the version numbers for the files are equal, the event time is checked again, this time without the 30-minute window. In other words, if the event time associated with the file on Computer A is later than the event time for the file on Computer B, the change is accepted and Computer B is updated. If the event time associated with the file on Computer A is earlier than the event time for the file on Computer B, the change is rejected.

FRS uses a "last writer wins" algorithm, which means that the last update to a file or folder in a replica set becomes authoritative for replication, regardless of the document version number or file size. It does not merge changes; rather, the most recent version of a particular file overwrites all older versions. This makes FRS well suited to replicate files that are updated infrequently, such as product specifications, software distribution points, and Web content.

Files that contain information that is updated more frequently must accommodate two scenarios: concurrent users and replication latency.

- User A and User B open the same 100-page document on different replicas. User A adds 100 pages and saves the document first; user B deletes 80 pages and then saves the document. The 20-page document that was saved last becomes the authoritative file.

  FRS cannot deny file sharing or enforce file locking between two users who are writing to the same file on two different replicas.
- A user makes a change on a replicated Dfs share. Assume that the replication schedule for Dfs connection objects in Active Directory specifies that replication take place only at night. This means that updates originating on replicas in one site during the day are not available on replicas in other sites until the replication window opens in the evening.

FRS uses Active Directory to manage configuration information so you can administer computers containing replicated data offline. This means you can add or remove replicas, change connections between replicas, or modify replication schedules without needing to communicate with the affected computer.

Note The Dfs administrative console does not support full remote administration; that is, the affected computer must be online.

To provide secure communications, authenticated remote procedure call (RPC) with Kerberos encryption is used over TCP/IP as the protocol between members for replication. NTFS file permissions are also replicated. Files that are locked by their owners are not replicated until they are unlocked.

FRS works only with Windows 2000 because it relies on the NTFS change journal to provide a persistent (that is, logged) record of files that have changed on a member computer. Files are replicated only after they have been modified and closed. As a result, FRS does not lose track of a changed file even if the system shuts down abruptly.

Detailed Operation

FRS monitors the NTFS change journal for changes to its shares, applies a filter to exclude changes to nonreplicated files, places changes in a staging directory until they can be processed, and sends change notifications so that replication partners can pick up their changes. When the partners retrieve the staging file, it is considered replicated and deleted from the staging directory.

A list of partners is specified for each replica set. For SYSVOL, it is created automatically by the KCC (Knowledge Consistency Checker), which runs periodically on Windows 2000 domain controllers, to optimize and adjust the topology for failed computers or lost connections. When FRS is used to replicate Dfs shares, KCC is not involved. You must define replica sets by using the Dfs snap-in.

The fundamental objects in a replica set are computers and replication links. The replication links that connect computers are unidirectional. A change flows in the direction of the link between two partners. To replicate changes in both directions, a pair of links is necessary.

Before taking a detailed look at FRS operation, it is necessary to clarify the terms *inbound* and *outbound* partners in a replica set. Assume that two computers, A and B, host a replica set and that replication is enabled. As shown in Figure 18.1, if a file has changed on Computer A and needs to replicate to Computer B, then Computer B is Computer A's outbound partner and Computer A is Computer B's inbound partner. If a change occurs on Computer B that needs to be replicated to Computer A, then a second link is needed. In this case, Computer B is the inbound partner for Computer A, and Computer A is the outbound partner for Computer B.

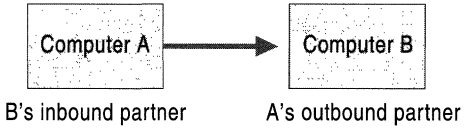


Figure 18.1 Inbound and Outbound Relationships

Figure 18.2 shows the detailed sequence of events that occurs when a file change introduced on Computer A replicates to Computer B.

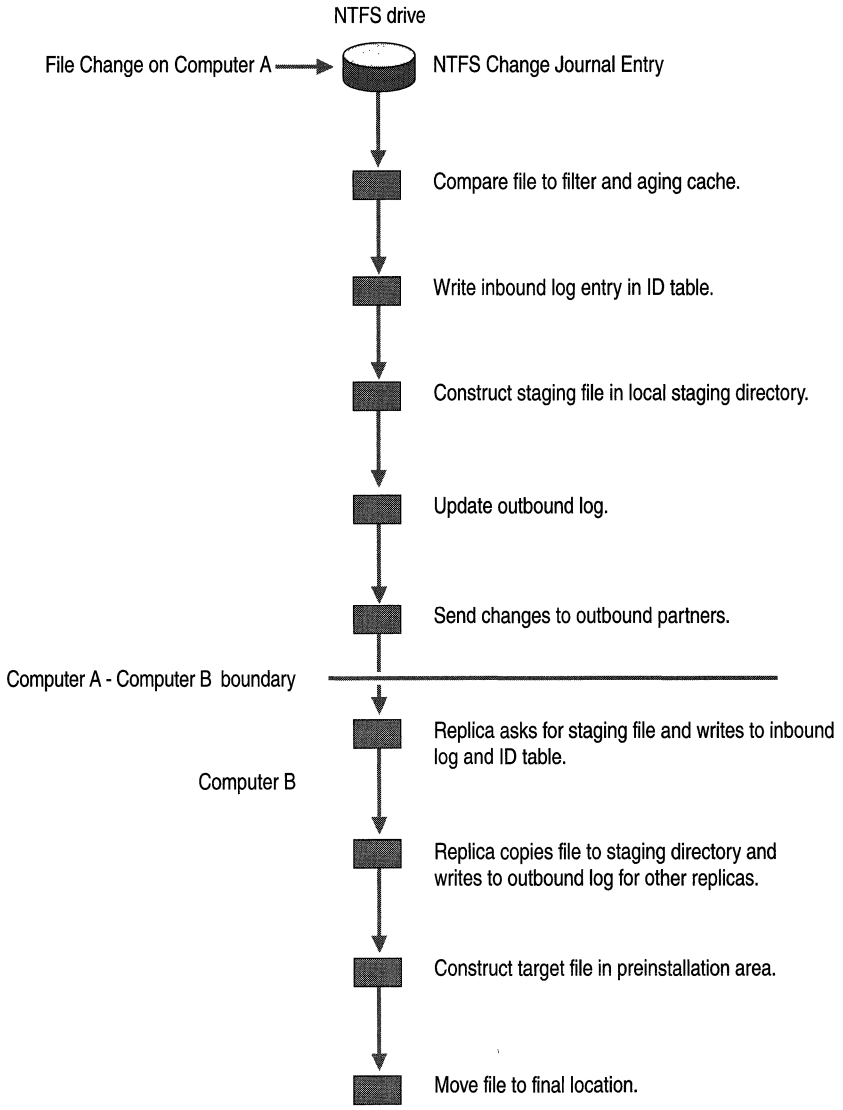


Figure 18.2 Detailed FRS Operation

1. When a file in a replica set changes and the user closes the file, NTFS makes an entry in the NTFS change journal.

   The NTFS change journal records changes to all files on the NTFS volume such as file creations, deletions, and modifications. The journal is limited in size, but it is persistent across restarts and crashes.

2. FRS monitors the NTFS change journal for changes that apply to the replicated shares. Only closed files are checked. File and folder filters are applied against changes in the folders of interest, notably domain-based Dfs and SYSVOL replica sets.
3. The aging cache, a three-second delay designed to catch additional changes to a file, expires. This prevents file replication when the file is undergoing rapid updates.
4. Computer A records the change as a change order in an inbound log. It also creates an entry in the ID table so that recovery can take place if a crash occurs.

   The inbound log contains change orders arriving from all inbound partners. The change orders are logged in the order that they arrive along with the file data. Each change order contains information about a change to a file or folder on a replica member, such as the name of the file or the time it was changed, that is used to construct a message about the change.
5. A copy of the changed file is constructed in a local staging directory.

   The staging directory is an area where modified files are stored temporarily prior to being propagated to other partners. A staging file is used by FRS to encapsulate the data and attributes associated with a replicated file (or directory) object. It ensures that the file data can be supplied to partners regardless of any file activity that might prevent access to the original file.
6. Computer A updates the outbound log.

   The outbound log contains change orders generated for a specified replica. The changes can originate locally or come from an inbound partner. These change orders are eventually sent to all outbound partners.
7. Computer A sends a change notification to Computer B.
8. If it decides to accept the change order, Computer B asks for the modified file. Computer B writes to its inbound and ID logs.
9. Computer B copies the staging file to its staging directory. It then writes to its outbound log so other outbound partners can pick up the change.

   The staging directory is an area where propagated files are stored temporarily prior to being installed locally on the partner. This is done so that users do not see a file locked for an extended period of time while FRS is moving the file over a slow or congested link. In addition, if the link fails in the middle of the transfer, users do not see a partial file.
10. The altered file is constructed in a preinstallation area and moved to its final location on Computer B.

FRS Tables

FRS transactions are stored in a Microsoft Jet database that defaults to *systemroot*\\Ntfrs\Jet\Ntfrs.jdb. Each replica set hosted by a computer has a set of tables stored in the Ntfrs.jdb file. These five tables are:

- Connection table. Contains one record per link or inbound/outbound partner connection.
- Inbound log. Stores pending change orders to be processed. As entries are processed, acknowledgments are sent to the inbound partners. Data stored in the inbound log includes change order globally unique identifier (GUID), file name, object ID, parent object ID, version number, and event time.

  During a planned shutdown, all new change orders since the last update are written to the inbound log. If an unplanned shutdown or network interruption occurs, the inbound partner resends all the change orders in its outbound log for which acknowledgments have not been received.
- Outbound log. Stores pending change orders to be sent to outbound partners. Change orders remain in the outbound log until all outbound partners receive and acknowledge the change. Data stored in the outbound log is the same as that stored in the inbound log. Also in the outbound log is the leading (next change) and trailing (last acknowledged) index for each partner.

  The outbound log or logs can become quite large, particularly when replicas are down, links between replicas are slow, replication hours are restricted, or a large number of changes occur. For example, suppose one of four replicas is down. Snapshots of the file image and log entries are maintained until this server becomes available. When the changes are finally sent, the inbound partner sends all changes in log file order.

  If the trailing location for a replica partner is overwritten, a complete synchronization must be completed for the replica. This involves the outbound partner sending its version vector — an array of numbers for each originator — with the changes it has received to the inbound partner. The inbound partner checks its ID table by using this state to determine what changes occurred afterward and sends them.
- Version vector. Measures the up-to-dateness of a replica compared to another replica. Each replica member in a replica set is assigned a number. Version vector joins occur the first time an FRS context is replicated or when the outbound log wraps.
- ID table. Lists all files in the replica set of which FRS is aware. Data stored in the ID table includes globally unique identifier (GUID), file name ID, parent file ID, file object ID, parent object ID, version number, and event time.

Changes are always logged in the Jet database for recovery purposes before any disk files are moved in case of a system failure.

FRS Startup

When FRS starts up, either when the host computer or the service starts (by using **net start ntfrs**, for example), it first polls Active Directory for configuration changes. Then FRS determines the inbound and outbound partners for each replica set.

FRS also polls Active Directory at regular intervals for configuration changes that affect its partner relationships. Events that reset the polling interval are:

- Adding a replica.
- Deleting a replica.
- Adding a connection.
- Deleting a connection.
- Changing a schedule.
- Changing a file or folder filter.

Upgrading LMRepl to FRS

Microsoft® Windows NT® Server version 4.0 and earlier provided a single-master file replication facility known as LAN Manager Replication (LMRepl). LMRepl was often used to replicate logon scripts and other database information to all domain controllers in the domain. FRS replaces LMRepl in Windows 2000 Server.

Windows 2000 Server does not support LMRepl in mixed or native mode, so if you have been using LMRepl, you need to develop a strategy for using FRS to provide the same functionality. If you used LMRepl to replicate logon scripts, you need to transition to using SYSVOL. If you used LMRepl for other purposes, investigate using Dfs for replication.

LMRepl Process

LMRepl uses the concept of import directories and export directories. You configure LMRepl by selecting a server to host an export directory and a number of servers to host import directories. The servers hosting the directories do not need to be domain controllers; they can be ordinary member servers. Figure 18.3 illustrates the LMRepl process.

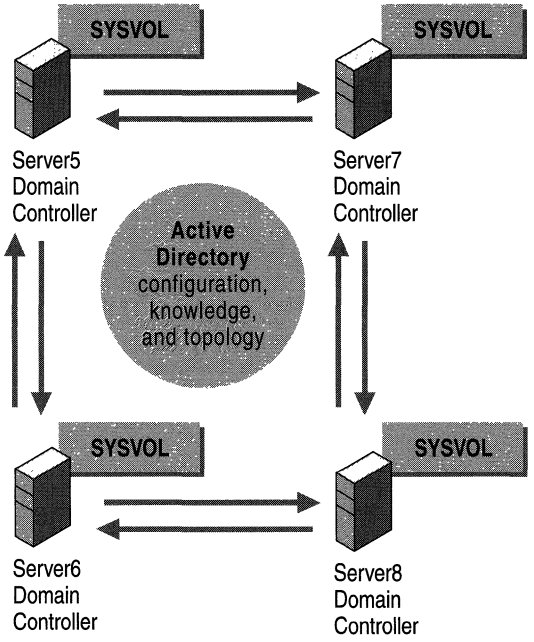


Figure 18.3 LMRepl Process

FRS Process

FRS in Windows 2000 Server is automatically configured so that every domain controller has a replicated SYSVOL. Any change you make to a logon script stored in the SYSVOL of any domain controller is replicated in multimaster fashion to other domain controllers. Unlike LMRepl and the hosting of import and export directories, only domain controllers can host SYSVOL. Figure 18.4 illustrates the FRS process.

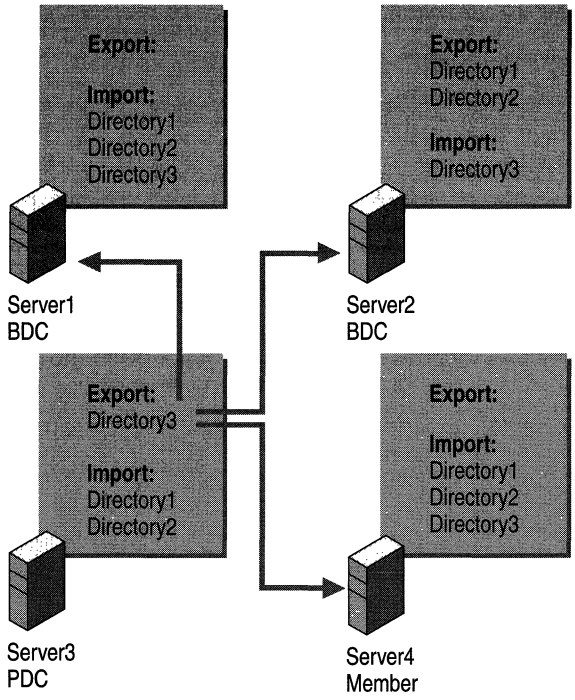


Figure 18.4 FRS Process

Maintaining a Mixed Environment

During an upgrade you can have a mixed environment of Windows NT 4.0 backup domain controllers and member servers operating with Windows 2000 domain controllers. However, because Windows 2000 Server does not support LMRepl, maintaining LMRepl services in a mixed environment can be an issue. To provide this support, you need to create a bridge between LMRepl and FRS so that both services can operate autonomously. Do this by selecting one Windows 2000 domain controller to copy the files that you want replicated to the Windows NT 4.0 export directory. The copying can be accomplished by a regularly scheduled script.

Note The term *mixed environment* is not to be confused with *mixed mode*, which refers to Windows NT 4.0 and Windows 2000 domain controllers existing within a domain. A mixed environment is a looser term that describes Windows NT 4.0 backup domain controllers (BDCs) and member servers operating with Windows 2000 domain controllers.

To maintain availability of LMRepl during an upgrade, make sure that the server hosting the export directory is upgraded only after all the other servers hosting import directories have been upgraded. If the server hosting the export directory is the primary domain controller (PDC), you should select a new server to host the export directory and then reconfigure LMRepl.

Customizing FRS

You can customize what FRS replicates and when by specifying a filter that excludes certain types of files or folders and by scheduling replication between sites.

Setting File and Folder Filters

File and folder filters are maintained for each FRS replica set including:

- SYSVOL
- Domain-based Dfs root shares
- Domain-based Dfs links

By default, the following files and folders are excluded from FRS replication:

- Encrypting File System (EFS)–encrypted files and folders that are computer specific
- File names starting with a tilde (~) character
- Files with .bak or .tmp extensions
- NTFS mount points
- All reparse points

Filters act as exclusion filters only for new files and folders added to a replica set. They have no effect on existing files in the replica set. For example, if you change the file filter from "*.tmp, *.bak" to "*.tmp, *.old," FRS does not go through the replica set and exclude all files that match *.old. Nor does it go through the replica set and begin to replicate all files that match *.bak.

After the filter change, new files that enter the replica set matching *.old are not replicated. New files entering the replica set matching *.bak are replicated. In addition, any preexisting file in the replica set that matched the old file filter (such as Test.bak, created when the old filter was in force) is not automatically replicated when the filter changed. You must explicitly modify such files before they begin replicating. Likewise, you must explicitly delete any preexisting files in the replica set that match *.old. Until that happens, changes to those files continue to replicate.

These rules apply in the same manner to the directory exclusion filter. In addition, for directories, if a directory is excluded, all subdirectories and files under that directory are also excluded.

The rationale for these rules is as follows. If you accidentally change a filter to exclude a file like *.doc, FRS does not go through and delete every Microsoft® Word file in the replica set. Similarly, if you unintentionally omit *.tmp from the filter, FRS does not go through each replica and begin replicating every temporary file it finds.

Use the Active Directory Users and Computers console to modify a file or folder filter.

▶ **To change the file or folder filter**

1. Open Active Directory Users and Computers, click **View**, and then click **Advanced Features**.
2. In the console tree, expand **System**, expand **File Replication Service**, and then expand **DFS Volumes**.
3. Expand the Dfs root and link that has the filter you want to change.
4. Right-click the link, and then click **Properties**.
5. On the **Replica Set** tab, enter your changes in the **File Filter** box, and then click **OK**.

Scheduling Replication

Schedules govern intersite replication between servers or domain controllers and also intrasite replication for Dfs replica sets. Intrasite replication for SYSVOL occurs automatically.

On SYSVOL

Site-to-site replication for SYSVOL is determined by the schedule for its connection object in Active Directory. It is treated as a trigger schedule, which means that when the connection schedule enables replication at a specified time, all pending file changes that have accumulated on an inbound partner are now replicated to the outbound partner. This can take minutes or hours depending on how much file data needs to be replicated.

Because trigger schedules are used with SYSVOL, its replication behavior is similar to that for Active Directory objects. Remember that replication of directory objects is completely separate from replication of SYSVOL files by FRS. As shown in Figure 18.5, the default interval for intersite replication of SYSVOL files is once per hour.

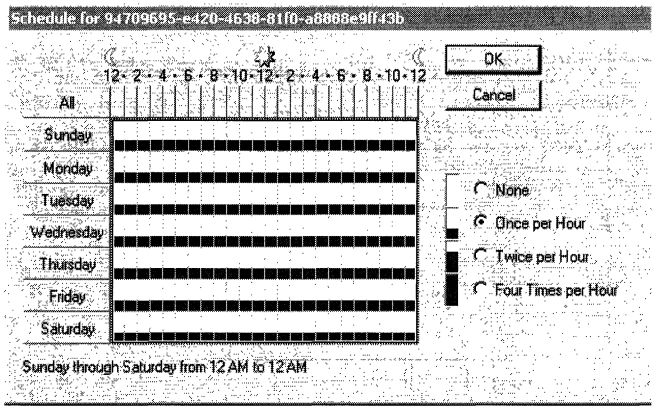


Figure 18.5 Default Replication Schedule for SYSVOL

▶ **To check the schedule on the connection object for SYSVOL**

1. Open the Sites and Services administrative console.
2. In the console tree, expand **Sites**, and then expand the site of interest.
3. Expand **Servers**, expand the server of interest, and then expand **NTDS_Settings**.
3. Right-click the connection of interest, and then click **Properties**.
4. Click **Change Schedule** to view the schedule for that connection.

On Dfs Replicas

A schedule for a replica set can be assigned to a connection object or to the replica set itself. Normally, a connection object is preferred because a schedule assigned to a connection object overrides a schedule assigned to a replica set. However, assigning a schedule to the replica set might be easier in certain circumstances. For example, if a replica set had a large number of replicas, say 100, it is a tedious process to configure all the connection objects.

The schedule for the replication of a replica set is interpreted as on or off. For example, the schedule shown in Figure 18.6 enables replication between 5 A.M. and 8 A.M. FRS starts replicating to the outbound partners at 5 A.M. and stops replicating at 8 A.M. even if not done sending all the files. This allows you to allocate replication when network bandwidth is available. For example, you can schedule replication during nonpeak hours in case a user dumps a 1 gigabyte (GB) file into a replica set.

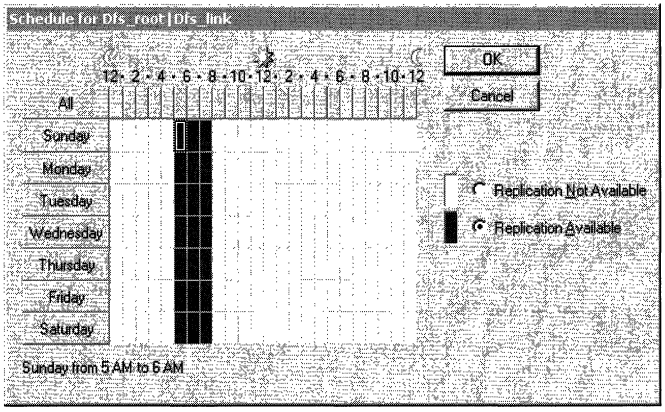


Figure 18.6 Replication Schedule for Dfs Replicas

▶ **To change the schedule on the connection object for a replica set**

1. Open Active Directory Users and Computers, click **View**, and then click **Advanced Features**.
2. In the console tree, expand **System**, expand **File Replication Service**, and then expand **DFS Volumes**.
3. Expand the Dfs root and link of interest.
4. Right-click the link, and then click **Properties**.
5. Click **Change Schedule** to view the schedule for that connection.

▶ **To change the schedule on the entire replica set**

1. Open Active Directory Users and Computers, click **View**, and then click **Advanced Features**.
2. In the console tree, expand **System**, expand **File Replication Service**, and then expand **DFS Volumes**.
3. Continue expanding until you reach the replica set of interest.
4. Click **Change Schedule** to view the schedule for that replica set.

   Remember that if a schedule is assigned to a connection object within this replica set, it overrides the schedule assigned to the replica set for that connection.

Forcing Replication Between Sites

The only way to force replication by FRS is to enable its schedule (assuming that it is off). The **Replicate Now** button in the Active Directory Sites and Services console pertains to directory replication only.

Tuning Recommendations

Use the following guidelines to help you optimize the performance of FRS.

Setting up the Dfs Topology

The size and shape of a replica set, that is, the depth and breadth of its tree, can affect performance significantly. For example, if the replica set is fully meshed, with each member connected to every other member, the propagation of change orders for replicating files can impose a heavy burden on the network. To reduce unnecessary traffic, delete connections you do not actually need.

By default, Dfs creates a full-mesh topology. When you add a replica, Dfs creates a link between the new replica and every other member in the replica set and vice versa. To remove unwanted connections, use the Users and Computers administrative console. For guidelines on designing a Dfs topology, see "Distributed File System" in this book.

Distributing Disk Usage

To spread out disk traffic, locate the FRS logs on a separate disk from the staging directory, the working directory, and the replicas themselves. (The working directory is the Ntfrs.jdb file.) This is especially important when logging with a high severity level. In fact, putting each of these areas on a separate disk drive gives the best replication performance because it distributes disk input/output (I/O).

To change the location where trace log files are stored, go to HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NtFrs\Parameters, and edit the value of the **Debug Log File** entry. Before this change becomes effective, you must stop and restart FRS.

Caution Do not use a registry editor to edit the registry directly unless you have no alternative. The registry editors bypass the standard safeguards provided by administrative tools. These safeguards prevent you from entering conflicting settings that might degrade performance or damage your system. Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. To configure or customize Windows 2000, use the programs in Control Panel or Microsoft Management Console (MMC) whenever possible.

Disabling Logging

If debugging replication problems is not a priority, you can disable logging to reduce disk traffic. Go to HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NtFrs\Parameters, and change the value of the **Debug Disable** entry to 1.

Maintaining Throughput

A slow outbound partner can cause the staging directory to fill up. To avoid an interruption to FRS replication because the staging directory is full, construct replica connections that have comparable bandwidth for all outbound partners. It is also a good idea to balance the bandwidth for inbound and outbound connections.

Adjusting the Size of the Staging Directory

Make sure the staging directory for replicated shares is large enough. The staging space limit governs the maximum amount of disk space that FRS can use to hold staging files. When this limit is reached, inbound replication pauses until space can be recovered by replicating one or more staging files to all outbound partners.

To adjust the size of the staging space, go to HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NtFrs\Parameters, and edit the value of the **Staging Space Limit in KB** entry.

Using FRS with Remote Storage

A replica set can include files that are stored offline in a tape library managed by Remote Storage. However, if you decide to deploy FRS and Remote Storage, remember that the overhead from replicating offline files can be substantial. For example, suppose a replica set has two members: Computer A and Computer B. Computer A uses Remote Storage to hold its replicated files whereas Computer B keeps its replicated files online. If Computer A is the initial master of the replica set, all of its files must be extracted from remote storage and replicated to Computer B when the latter is added to the replica set.

Frequent replication of data in the replicas can also be a burden for Computer A if the affected files must be retrieved from a tape. This is less of a concern when only a small subset of the replicated files, which are kept in disk storage on Computer A, are changed frequently by users.

Monitoring Performance

The two performance objects associated with File Replication service are listed in Table 18.1. The counters for these objects are available for each replica set managed by FRS while the service is running.

Table 18.1 FRS Performance Objects

Object	Description
FileReplicaConn	Performance statistics for the Replicaconn object that defines replica connections to Dfs roots.
FileReplicaSet	Performance statistics for the Replicaset object that defines a replica set.

Important FileReplicaSet counters that you can monitor to verify FRS performance are listed in Table 18.2.

Table 18.2 Key FileReplicaSet Counters

Counter	Description
Change Orders Received	Number of change notifications received from inbound partners.
Change Orders Sent	Number of change notifications sent out to outbound partners.
File Installed	Number of replicated files installed locally.
KB of Staging Space Free	Amount of free space in the staging directory used by FRS to temporarily store files before they are replicated. The default staging space is 660 megabytes (MB).

(continued)

Table 18.2 Key FileReplicaSet Counters *(continued)*

Counter	Description
KB of Staging Space In Use	Amount of space in the staging directory currently in use. If the staging directory runs out of space, replication stops.
Packets Received	Amount of data received locally. These packets can be change notifications, file data, or other command packets.
Packets Sent	Similar to packets received.
USN Records Accepted	Number of records that are accepted for replication. Replication is triggered by entries written to the NTFS change journal. FRS reads each file close record from the journal and determines whether to replicate the file. An accepted record generates a change order, which is then sent out. A high value on this counter (about one every five seconds) indicates a lot of replication traffic. This can cause replication latency.

You can also observe FRS activity using Event Viewer. Select **File Replication service**, and then filter events by time and date or by type of event such as information, warnings, or errors. Right-click the individual message, and then click **Properties** to display details about an event.

Restoring Replicated Files

Replicated files and folders can be backed up like any other share as long as you are aware of how replicated data is restored, in case a restore becomes necessary. In fact, a backup copy of SYSVOL or a Dfs replica is a good idea in case disaster recovery must be carried out.

Backup does not distinguish what type of data (replicated or otherwise) is being backed up or restored. If FRS replicates the restored files after a restore, this can be good or bad depending whether the files contain the latest or most valid data.

For example, when only one member in a replica set is lost, say, from the failure of a disk drive, restore its contents from a backup tape, and then let FRS restore the files that have changed since the backup tape was made. This minimizes network traffic by not restoring static files. This is called a *nonauthoritative* restore.

On the other hand, if the entire replica set or every copy of SYSVOL has been corrupted, restore one replica from the backup tape, and then replicate the restored files and folders to the other members in the replica set. This is called an *authoritative* restore.

Note Backup does not back up the FRS database (that is, the tables in Ntfrs.jdb). If the database becomes corrupted or lost, the server reconstructs it by comparing the files in its replica with the files in another replica on a different computer.

Nonauthoritative Restore Process

Version numbers are not retained with a file when it gets backed up. However, if the file was backed up from a replica set, it has a file object ID saved along with the other file attributes. The file object ID guides the nonauthoritative restore process.

Use a nonauthoritative restore to create a new replica member, which is the recommended course of action if an individual replica is lost. First, remove the failed member from the replica set. Then, restore the replicated share from the backup tape into what is to become the root directory for the new member. Next, add the member to the replica set, specifying the root path where the replicated share was just restored. Note that FRS has been running all this time, but because this member was removed from the replica set, none of the restored files has been replicated.

When FRS notices that the configuration has changed, it begins its initialization sequence to add the member. The first step is to move all the files just restored from the new replica's root directory to a temporary directory. Next, FRS joins with an inbound partner and requests information about every file in the replica share. The inbound partner supplies information such as the version number, file object ID, and so forth, for each file and folder. By using the file object ID, FRS locates the file or folder that was restored from tape and does a checksum-based comparison (using MD5) of the file contents with the corresponding checksum supplied by the inbound partner. If the checksums match, FRS places the file from tape into the replica share. Otherwise, FRS requests the file from the inbound partner. If the backup data is fairly current, few files are copied from the inbound partner.

At the conclusion of the nonauthoritative restore process, the file content and the version information on the new member match the content on the inbound partner. In essence, the files supplied from the backup tape are used only if they have file object IDs and their content matches the content of the corresponding file held by the inbound partner. This is especially valuable if the two members are linked by a low-speed network connection.

Authoritative Restore Process

With an authoritative restore process, the restored file and folders are given the newest version number. This means that the replicated share that was just restored is automatically replicated to other members in the replica set. Do an authoritative restore only if, for example, an entire replica set became corrupted.

Restoring Files on a Domain Controller

Restoration on a domain controller is FRS-aware. This means Backup recognizes that it is working with replicated data. Therefore, a nonauthoritative restore is always performed, because it is assumed that there is another replica in the domain with valid data.

If you want to force an authoritative restore, in the **Advanced Restore** dialog box, select **When restoring replicated data sets, mark the restored data as the primary data for all replicas**. This ensures that the restored data is replicated to your other servers. If you do not choose this option, the data you are restoring cannot be replicated to other servers because in a nonauthoritative restore the version of the file currently residing on the other replica set members always takes precedence. This causes other servers to overwrite the restored data, thereby preventing you from restoring the data. Forcing an authoritative restore assigns the newest version number to the replicated data to guarantee its replication to other servers.

Use the **When restoring replicated data sets, mark the restored data as the primary data for all replicas** switch if and only if this is the last member of the replica set. This switch is intended for disaster recovery cases when the whole replica set is lost. Setting a member as initial master when it has other members from which to synchronize can result in name collision.

▶ **To perform an authoritative restore of SYSVOL**

1. Restore the data from tape to an alternate location.
2. Perform a nonauthoritative restore.
3. After the SYSVOL share is published, copy the restored data from the alternate location into the actual location.

Restoring Files on a Member Server

Restoration on a member server is always authoritative because it is not FRS-aware. In other words, it assumes that there are no other copies of the restored files on other servers. As a result, the replica being restored replicates its data to other members of the replica set.

Note that an authoritative restore is simply the restoration of a file onto a member that is actively replicating files. It does not produce a mirror image of the backup tape content in the replica share. As an example, any new files that were created in the replica share after the backup tape was created are not deleted from the replica share. To perform a true authoritative restore of a replica share so that it mirrors the content from the backup tape, the user must first delete all files in the replica share and then restore the data from the backup tape.

▶ **To perform a nonauthoritative restore of a Dfs replica**

1. Remove the failed member from the replica set.
2. Disable replication on the host server.
3. Repair the faulty member. For example, replace a disk drive that has failed.
4. Add the member back as a new replica. Do not specify it as initial master when you enable replication unless you want to do an authoritative restore.

   Note that initial master is only relevant when this member is the only member (that is, the first member) in the replica set. In this case, FRS enumerates the replica share and preloads its database with information about each file and folder. In addition, FRS assigns file object IDs to every file and folder in the replica share. If this member is not the only member in the replica share, FRS always treats the addition of a new member as nonauthoritative as described previously.

Troubleshooting FRS

When FRS stops replicating content, the first thing to do is check the event log to see if the staging directory is full. If it is, replication stops. See step 7 in the following procedure for ways to correct this problem.

A general procedure for troubleshooting FRS problems consists of the following steps:

1. Verify that both Computer A and Computer B are available on the network. Because FRS uses the fully qualified domain name (FQDN) of the replica members, a good first check is to use a **ping** command specifying the fully qualified name of the problem replicas. From Computer A, send a **ping** command with Computer B's FQDN. From Computer B, send a **ping** command to Computer A's FQDN. Verify that the addresses returned by the **ping** command are the same as the addresses returned by an **ipconfig /all** command carried out on the command line of the destination computer.
2. Use the Services administrative console to confirm that FRS is running on the remote computer.
3. If the service is not running, review the File Replication service container of Event Viewer (Eventvwr.msc) on the problem computer. If the service has asserted, troubleshoot the assertion. Otherwise, restart the service.
4. Verify RPC connectivity between Computer A and Computer B. Check FRS event logs on both computers. If Event ID 13508 is present, there might be a problem with the RPC service on either computer or with creating a secure connection between Computer A and Computer B.
5. Use the Active Directory Sites and Services console to verify the replication schedule on the Connection object. Make sure that replication is enabled between Computer A and Computer B and that the connection is enabled. The Connection object is the inbound connection under Computer A's NTFRS_MEMBER object from Computer B. For SYSVOL, the connection object resides under Sites*site_name*\Servers*server_name*\NTDS Settings*connection_name*.
6. Create a test file on Computer B, and verify its replication to Computer A.

7. Check for free disk space on Computer A (source directory, staging directory, and database partition) and Computer B (destination partition, preinstall partition, and database partition). Look for the following events in Event Viewer:
   - 13511: Database is out of disk space.

     For more information about how to move the database to a larger volume, see "FRS Tables" earlier in this chapter.
   - 13522: Staging directory is full.

     This can be caused by an outbound partner that has not connected for a while. Delete the connection and stop and restart FRS to force deletion of the staging files. You can also follow the procedure described in "Tuning Recommendations" earlier in this chapter to increase the size of the staging directory.
8. Check for files that are larger than the amount of free space on the source or destination server or larger than the size of the staging directory limit in the registry. Resolve the disk space problem or increase the maximum staging file space. See Error 13522 for details.
9. Check whether the file on the originating server is locked on either computer. If the file is locked on Computer B so that FRS cannot read the file, FRS cannot generate the staging file, which delays replication. If the file is locked on Computer A so that FRS cannot update the file, FRS continues to retry the update until it succeeds. The retry interval is 30 to 60 seconds.
10. Check whether the source file was excluded from replication. Confirm that the file is not EFS encrypted, a NTFS junction, or excluded by a file or folder filter on the originating replica member. If any of these are true, FRS does not replicate the file or directory.
11. If all of the previous conditions check out, try to replicate the file again.

FRS Logs

FRS creates text-based logs in the *systemroot*\Debug directory to help you debug problems. The Ntfrsapi.log file contains events that take place during promotion and demotion—namely, creating the subkeys in the NTFRS registry key.

To observe a particular event, take a snapshot of the log files as close to the occurrence of the event as possible. Save the log files in a different location so they can be examined afterward.

Log Settings

The Ntfrs log files store transaction and event detail in sequentially numbered files: Ntfrs_0001 through Ntfrs_0005. Transactions and events are written to the log with the highest version number in existence at that time. The characteristics of the log files are determined by the values of several registry entries in the HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NtFrs\Parameters subkey. After the number of logs specified by the value of the **Debug Log Files** registry entry have been filled, the lowest log version is deleted and the remaining log file names are decremented by *n*–1 to make room for a new log file.

Log detail is controlled by the value of the **Debug Log Severity** registry entry, ranging from 0 to 5, with 5 providing the most detail. Log size is determined by the value of the **Debug Maximum Log Messages** registry entry. The default value of 10,000 lines for **Debug Maximum Log Messages** results in a ~ 1-MB log file for a total of 10 MB of logs (**Debug Log Files** * (**Debug Maximum Log Message** * 100)). Setting **Debug Log Messages** to 50,000 results in a 5-MB log file and 25 MB of log detail with default settings.

To change the quantity, size, or level of detail of FRS log files, edit the values of the registry entries. Before you increase either the size or quantity of log files, make sure sufficient disk space is available. In general, budget 1 MB for each 10,000 messages.

Caution Do not use a registry editor to edit the registry directly unless you have no alternative. The registry editors bypass the standard safeguards provided by administrative tools. These safeguards prevent you from entering conflicting settings that might degrade performance or damage your system. Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. To configure or customize Windows 2000, use the programs in Control Panel or Microsoft Management Console (MMC) whenever possible.

▶ **To change the characteristics of Ntfrs log files**

1. From the **Start** menu, click **Run**.
2. Type **cmd**, and then type **net stop ntfrs**.
3. Start a registry editor (either Regedit.exe or Regedt32.exe).

4. Go to HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NtFrs\Parameters, and edit the values of the following entries:
   - **Debug Log Severity**: Determines the level of detail in the Ntfrs_000n.log files. Severity levels are assigned to different debug print statements in the FRS code. If the value of this entry is 0, only the most severe events are recorded in the log. If the value of this entry is 5, almost all events are recorded in the log. Higher values are inclusive.
   - **Debug Log Files**: Determines how many FRS service transaction and event log files can be active simultaneously. Logs are written on a first in–first out basis with the highest number containing the most recently logged entries. Logs 1 through 5 are created in sequential order. When the number of messages written to Ntfrs_0005.log reaches the value of **Debug Maximum Log Messages**, Ntfrs_0001.log is deleted, Ntfrs_0002.log becomes 1, 3 becomes 2, and so on.
   - **Debug Maximum Log Messages**: Determines how many entries can be stored in a debug log file. 10,000 entries use approximately 1 MB of space on disk.

   These entries are not visible by default. Use the **Edit** menu to add them, if necessary.
5. Close the registry editor and restart FRS. FRS creates the specified log files automatically.

To capture a random or intermittent event, you might want to expand FRS logging capability. For example, you can increase the number of log files to 50 and then archive the files when they become full. This accumulates the history needed to respond to overnight queries from users, for example.

Depending on the problem that is being investigated, it might be necessary to review logs on both the inbound and outbound replicas. System clocks must be synchronized so that events can be correlated between replication partners.

Finally, the recovery setting for the FRS service in service control manager (SCM) can be critical to locating and keeping important log events on the system. If the service is asserting but SCM is configured to automatically start FRS upon error, enough log traffic might be generated to cause events in Ntfrs_0005.log to decrement and be deleted from the drive. Stop the service on both the inbound and outbound replicas close to the time when an error occurs, and then copy the logs to a safe place.

Analyzing Log Files

The first step to solving problems with the logs is to make sure the **Debug Log Severity** entry in the registry is set high enough to capture the events needed to identify the problem. Severity settings range from 0 to 5 and are cumulative, meaning that a setting of 4 includes log events with a severity of 0 to 3.

Next, identify errors, warning messages, and milestone events in the log files. A good practice is to start at the bottom of the last log file and work your way up. Focus on keywords such as "install," "success," and "fail." If you do not find the error that you are looking for, start at the bottom of the previous log (Ntfrs_0005, then Ntfrs_0004, and so on). Use the **find** command to isolate errors in the log files as follows:

```
find /in "error warn fail" ntfrs.* >err.tmp
```

Depending on the context, some errors (such as, "jet attach db – 1811. Db not found") can be ignored because the Ntfrs.jdb file does not exist the first time FRS starts. Until the service creates the file, expect to see this immediately after Dcpromo or when you delete the Ntfrs.jdb file manually. Sharing violations, designated by the SHARING_VIOLATION status code, occur when a user or process has a lock on a file. Because FRS tracks only closed files, locked files and directories do not replicate.

If failure errors are encountered, look at the thread number and follow up all events in the log that have matching thread identifiers until you see the associated change order.

To determine why a file on Computer A has not replicated to a second or third replica, locate the ":: COG" number in the Ntfrs_00*n*.log files on the originating server. Search for the same globally unique identifier (GUID) in the logs on the second and third replicas.

Ntfrsutl Tool

You can use the Ntfrsutl tool to do the following:

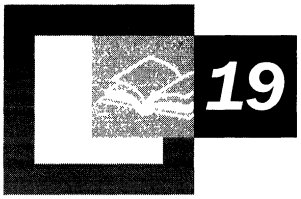
- Show the ID table, inbound log, or outbound log for a computer hosting FRS.
- Examine memory usage by FRS.
- Show the FRS configuration in Active Directory.
- List the active replica sets in a domain.
- List the application programming interface (API) and version number for FRS.
- Poll immediately, quickly, or slowly for changes to the FRS configuration.

The syntax for Ntfrsutl is shown in Figure 18.7.

Figure 18.7 Ntfrsutl Tool

CHAPTER 19

Network Load Balancing



Network Load Balancing is one of the Windows Clustering features of Microsoft® Windows® 2000 Advanced Server. Network Load Balancing can enhance the availability and scalability of Internet server programs such as those used on Web servers, FTP servers, and other mission-critical servers. The material in this chapter will help you configure Network Load Balancing clusters to deliver the reliability and performance that high-volume Web servers and other mission-critical servers need.

In This Chapter

Network Load Balancing Overview

Network Load Balancing clusters distribute client connections over multiple servers, providing scalability and high availability for client requests for TCP/IP-based services and applications.

The heart of Network Load Balancing is the driver Wlbs.sys, which is loaded into each member server, or *host*, in the cluster. Wlbs.sys includes the statistical mapping algorithm that the cluster hosts collectively use to determine which host handles each incoming request.

Load-balanced applications need to carefully manage state on the server. State that is persistent across multiple requests or that is shared among clients needs to be shared in locations that are transparently accessible from all cluster hosts. Updates to state that is shared among the hosts needs to be synchronized—for example, by using a back-end database server.

Network Load Balancing provides the following:

- Load balancing

  Load balancing is static (unless the cluster set changes). The port rules that you create for the cluster's hosts specify the division of the incoming client connections.

- High availability

  If a host fails, the cluster detects the failure and redistributes subsequent client requests to hosts that are still viable members of the cluster. Any client connections that were open when the host failed are ended. On retry, the client connection is routed to a viable host. The downtime for client connections is less than 10 seconds.

- Scalability

  Network Load Balancing accomplishes scalability by allowing you to add as many as 32 hosts to the cluster. You can add hosts without shutting down the cluster.

How Network Load Balancing Works

Briefly, when Network Load Balancing is installed as a network driver on each of the cluster hosts, the cluster presents a virtual IP address to client requests. The client requests go to all the hosts in the cluster, but only the host to which a given client request is mapped accepts and handles the request. All the other hosts drop the request. Depending on configuration of port rules and on affinity, the statistical mapping algorithm, which is present on all the cluster hosts, maps the client requests to particular hosts for processing.

The hosts exchange heartbeat messages to maintain consistent data about the cluster's membership. If a host fails to send or does not respond to heartbeat messages, the remaining hosts perform convergence, a process in which they determine which hosts are still active members of the cluster. If a new host attempts to join the cluster, it sends heartbeat messages that trigger convergence. After all cluster hosts agree on the current cluster membership, the client load is repartitioned, and convergence completes.

Discussion of Network Load Balancing clusters requires clarification of two kinds of client states, application data state and session state:

- In terms of application data, you must consider whether the server application makes changes to the data store and whether the changes are synchronized across instances of the application (the instances that are running on the Network Load Balancing cluster hosts). An example of an application that does not make changes to the data store is a static Web page supported by an IIS server.

  Means must be provided to synchronize updates to data state that need to be shared across servers. One such means is use of a back-end database server that is shared by all instances of the application. An example would be an Active Server Pages (ASP) page that is supported by an IIS server and that can access a shared back-end database server, such as a SQL Server.
- Session state (or intraclient state) refers to client data that is visible to a particular client for the duration of a session. Session state can span multiple TCP connections, which can be either simultaneous or sequential. Network Load Balancing assists in preserving session state through client affinity settings. These settings direct all TCP connections from a given client address or class of client addresses to the same cluster host. This allows session state to be maintained by the server application in the host memory.

  Client/server applications that embed session state within "cookies" or push it to a back-end database do not need client affinity to be maintained.

  An example of an application that requires maintaining session state is an e-commerce application that maintains a shopping cart for each client.

By setting port rules, cluster parameters, and host parameters, you gain great flexibility in configuring the cluster, which enables you to customize the cluster according to the various hosts' capacities and sources of client requests. You can:

- Divide the load of incoming client requests among the hosts according to a given load partitioning, expressed as percentages of the incoming client connections. You can optionally route all requests of a given client to the host that handled the client's first request (single affinity).

  Network Load Balancing normalizes the load percentage based on the sum of assigned load percentages for all active hosts. In other words, if one host fails, the remaining hosts increase the number of client requests they handle, proportionally to their original load percentages. For example, assume each host in a four-host cluster is assigned 25 percent of the load. If one of these hosts fails, the three remaining active hosts would each handle 33 percent of the load.
- Specify that one host handle all client requests, with the others serving as failover alternatives.

You can combine the preceding capabilities by setting cluster and host parameters and creating port rules for your particular scenario. For guidelines on setting parameters and port rules for various scenarios, see "Scenarios" later in this chapter.

Before specific scenarios are discussed, the following sections explore the basic concepts of Network Load Balancing:

- System Requirements

  This section includes caveats and recommendations.
- Components
- Network Load Balancing Design

  This section covers basic concepts, such as the parameters and port rules, heartbeats and convergence, how Network Load Balancing maps client requests to hosts, and maintaining client connections.
- Implementing Network Load Balancing

  This section discusses the cluster and host parameters and the port rules in more depth.

System Requirements

The following are Network Load Balancing requirements:

- Windows 2000 Advanced Server
- TCP/IP protocol
- FDDI, Ethernet, or Gigabit Ethernet
- Cluster hosts that reside on the same physical subnet
- 1.5-megabyte (MB) hard disk space
- Between 250 KB and 4 MB of RAM, using the default parameters and depending on the network load

The following are some additional considerations in creating an environment for Network Load Balancing:

- It is not supported for a given server to be a member of both a Network Load Balancing cluster and a server cluster.
- Network Load Balancing does not support load-balancing on a token ring network.
- Network Load Balancing can operate on a mixed-version cluster—that is, on a cluster in which some hosts run Windows Load Balancing Service under Microsoft® Windows® NT version 4.0 and some run Network Load Balancing under Microsoft® Windows® 2000.
- Although only one network adapter is necessary per host, an additional network adapter is recommended for separating, on each host, client requests from other network traffic that is not related to Network Load Balancing, such as content replication or access to a back-end database.

For information about installing or upgrading Network Load Balancing, including rolling upgrades, see Windows 2000 Network Load Balancing Help.

Components

The following are the principal Network Load Balancing components. They are installed to each Network Load Balancing cluster host ("Wlbs" remains from a previous version of the software):

- Wlbs.sys

  The Network Load Balancing networking device driver.

- Wlbs.exe

  The Network Load Balancing control program. Except for changing registry parameters, you can use Wlbs.exe from the command line to start, stop, and administer Network Load Balancing, as well as to enable and disable ports and to query cluster status.

  For information about command-line syntax and arguments that Wlbs.exe carries out, see Windows 2000 Network Load Balancing Help.

- Wlbs.chm

  Network Load Balancing Help.

Network Load Balancing Design

Rather than routing incoming client requests through a central host for redistribution, every Network Load Balancing cluster host receives each client request. A statistical mapping algorithm determines which host processes each incoming client request. The distribution is affected by host priorities, whether the cluster is in multicast or unicast mode, port rules, and the affinity set.

This design has the following advantages:

- Because filtering packets is faster than modifying them in one host and then retransmitting them to their destination hosts, which then must receive them, Network Load Balancing provides significantly higher throughput than do load-balancing solutions that route packets through a central host.
- Network Load Balancing avoids a single point of failure and provides redundancy equal to the number of servers in a cluster.
- Because Network Load Balancing is a software solution, it scales with the technology of the servers where it is installed.

The trade-off for these advantages is that sending all the client traffic to all the hosts means that the network adapter(s) in each host must handle all the incoming client traffic (which is usually a small percentage of overall traffic).

Requests That Require Synchronized Change in Data State

When a Network Load Balancing host processes a client request that requires changing state information that is visible to all application instances, the change in data must be synchronized across all the hosts in the cluster. To accomplish this synchronization, the application can maintain shared state information in a back-end database and generate an update to the back-end database server. If the target application is managed as a server-cluster resource, the back-end servers can be members of a server cluster. The application can also provide other methods of its own design, such as cookies, for managing shared state information.

Heartbeats and Convergence

Network Load Balancing hosts maintain membership in the cluster through heartbeats. By default, when a host fails to send out heartbeat messages within about five seconds, it is deemed to have failed, and the remaining hosts in the cluster perform convergence, in order to do the following:

- Establish which hosts are still active members of the cluster.
- Elect the host with the highest priority as the new default host.

  Note that the lowest value for the Priority ID host parameter indicates the highest priority among hosts.
- Redistribute the failed host's client requests to the surviving hosts.

In convergence, surviving hosts look for consistent heartbeats; if the host that failed to send heartbeats once again provides heartbeats consistently, it rejoins the cluster in the course of convergence. The other consistency that active hosts establish during convergence is that all the hosts have a consistent view of which hosts are active.

Convergence generally takes less than 10 seconds, so interruption in client service by the cluster is minimal.

By editing the registry, you can change both the number of missed messages required to start convergence and the period between heartbeats. However, making the period between heartbeats too short increases network overhead on the system.

During convergence, hosts that are still up continue handling client requests.

Statistical Mapping Algorithm

The assignment of a given client request to a server occurs on all the hosts; there is not a single host that centrally distributes the requests among the hosts. The hosts jointly use a statistical algorithm that maps incoming client requests to active hosts in the cluster.

Apart from the influence of cluster and host parameter settings, it is possible for two successive client requests to be assigned to the same host during normal operation. However, as more client requests come into the cluster, distribution of client requests by the algorithm statistically approaches the load division specified by the Load Weight parameter of the relevant port rule.

The distribution of client requests that the statistical mapping function effects is influenced by the following:

- Host priorities
- Multicast or unicast mode
- Port rules
- Affinity
- Load percentage distribution
- Client IP address
- Client port number
- Other internal load information

The statistical mapping function does not change the existing distribution of requests unless the membership of the cluster changes or you adjust the load percentage.

Affinity

Affinity defines a relationship between client requests from a single client address or from a Class C network of clients and one of the cluster hosts. Affinity ensures that requests from the specified clients are always handled by the same host. The relationship lasts until convergence occurs (namely, until the membership of the cluster changes) or until you change the affinity setting. There is no time-out—the relationship is based only on the client IP address.

There are three types of affinity, which you choose with the Affinity setting. The Affinity setting determines which bits of the source IP and IP port number affect the choice of a host to handle traffic for a particular client's request. The Affinity settings are as follows:

- None

  Setting Affinity to None distributes client requests more evenly; when maintaining session state is not an issue, you can use this setting to speed up response time to requests. For example, because multiple requests from a particular client can go to more than one cluster host, clients that access Web pages can get different parts of a page or different pages from different hosts.

With Affinity set to None, the Network Load Balancing statistical mapping algorithm uses both the port number and entire IP address of the client to influence the distribution of client requests.

In certain circumstances, setting Affinity to None is suitable when the Network Load Balancing cluster sits behind a reverse proxy server. All the client requests have the same source IP address, so the port number creates an even distribution of requests among the cluster hosts.

- Single

  When Affinity is set to Single, the entire source IP address (but not the port number) is used to determine the distribution of client requests.

  You typically set Affinity to Single for intranet sites that need to maintain session state. Single Affinity always returns each client's traffic to the same server, thus assisting the application in maintaining client sessions and their associated session state.

  Note that client sessions that span multiple TCP connections (such as ASP sessions) are maintained as long as the Network Load Balancing cluster membership does not change. If the membership changes by adding a new host, the distribution of client requests is recomputed, and you cannot depend on new TCP connections from existing client sessions ending up at the same server. If a host leaves the cluster, its clients are partitioned among the remaining cluster hosts when convergence completes, and other clients are unaffected.

- Class C

  When Affinity is set to Class C, only the upper 24 bits of the client's IP address are used by the statistical-mapping algorithm. This option is appropriate for server farms that serve the Internet. Client requests coming over the Internet might come from clients sitting behind proxy farms. In this case, during a single client session, client requests can come into the Network Load Balancing cluster from several source IP addresses during a session.

  Class C Affinity addresses this issue by directing all the client requests from a particular Class C network to a single Network Load Balancing host.

  There is no guarantee, however, that all of the servers in a proxy farm are on the same Class C network. If the client's proxy servers are on different Class C networks, then the affinity relationship between a client and the server ends when the client sends successive requests from different Class C network addresses.

Implementing Network Load Balancing

The following are required for applications to work with Network Load Balancing:

- They must use TCP connections or UDP data streams.
- If client data changes, applications must provide a means of synchronizing updates to client data that is shared on multiple instances across the cluster.
- If session state is an issue, applications must use single or Class C affinity or provide a means (such as a client cookie or reference to a back-end database) of maintaining session state in order to be uniformly accessible across the cluster.

Applications that are not compatible with Network Load Balancing have one or more of the following characteristics:

- They bind to actual computer names (examples of such applications are Exchange Server and Distributed file system).
- They have files that must be continuously open for writing (examples of such applications are Exchange Server and SMTP servers).

  In a Network Load Balancing cluster, multiple instances of an application (on different cluster hosts) should not have a file simultaneously opened for writing unless the application was designed to synchronize file updates. This is generally not the case.

Configuring Network Load Balancing

You define how the cluster load-balances client requests (and its other behaviors) by using the following Network Load Balancing parameters:

- Cluster parameters (primary IP address, subnet mask, full Internet name, multicast support, remote control password, and confirm password), which specify the behavior of the cluster.
- Host parameters (host priority, initial value, dedicated IP address, and subnet mask parameters), which define how each host functions within the cluster and in load-balancing.

  Host parameters are unrelated to the scenario for which you configure the Network Load Balancing cluster.

- Port rules (port range, protocols, filtering mode, affinity, load percentage, equal load distribution, and handling priority parameters), which define how the hosts distribute the incoming requests on a port or range of ports.

  Port rules define distribution of client requests for each scenario and must match on every cluster host. If a server attempts to join the cluster with a port rule that is inconsistent with the rest of the cluster or is incorrectly specified, the server is not accepted into the cluster, and the current load distribution is unchanged. The cluster does not complete convergence while there is a host with a port-rule mismatch.

  For information about ports and a useful list of port assignments, see the appendix "TCP and UDP Port Assignments" in the *Microsoft® Windows® 2000 Server Resource Kit TCP/IP Core Networking Guide*.

For information about the basic concepts of Network Load Balancing parameters and about configuring cluster parameters, host parameters, and port rules, see Windows 2000 Network Load Balancing Help.

For information about scenario-driven guidelines to setting values, see "Implementing Network Load Balancing" earlier in this chapter.

Client requests that you do not want to load-balance are a special case. For information about how to prevent load-balancing of a class of client requests, see "Default Handling of Client Requests" later in this chapter.

Cluster Parameters

By default, Network Load Balancing operates in unicast mode to ensure full compatibility with all makes of routers. In some cases, you might want to switch to multicast mode in order to avoid the use of a second network adapter for those communications between cluster hosts that are unrelated to cluster operations (for example, for content replication). If you switch to multicast mode, be sure that your router is compatible with this mode. For more information about the unicast and multicast modes, see Windows 2000 Network Load Balancing Help.

In unicast mode, Network Load Balancing on each host causes the network adapter's media access control address to be replaced with the cluster MAC address. (The media access control address is the hardware address, as distinct from the IP address.)

In multicast mode, the Network Load Balancing driver on each host retains the network adapter's media access control address and adds a multicast media access control address.

Note Network Load Balancing multicast mode is level-2 multicast. Do not confuse it with IP multicast. The virtual IP address must not be an IP multicast address.

For more information about multicast and unicast modes in Network Load Balancing, see Windows 2000 Network Load Balancing Help.

Host Parameters

Although you do not need to change the host parameters according to scenario, you should be aware of the following considerations.

Host Priority ID

This parameter defines the host's priority in being assigned client requests. The possible values are 1 to 32, inclusive; the lower the integer, the higher the host's priority.

Gaps in the numerical sequence of Host Priority IDs are allowed. This means that if a host from the middle of the sequence of Host Priority IDs goes out of service, the cluster can continue responding to clients.

The Host Priority ID does the following:

- Gives each host a unique identifier within the cluster.
- Establishes the default handling priority among hosts for traffic that is not load-balanced by port rules.

Initial State

Usually, you set the Initial parameter so that Network Load Balancing starts when the host boots. However, if the startup of the service receiving the client traffic is delayed after boot and there are many client requests for Network Load Balancing to distribute as soon as Network Load Balancing starts, a backlog of client requests would form before startup of the service for which Network Load Balancing is handling client requests. An example would be some earlier Web servers. In this case, you can filter traffic by starting Network Load Balancing manually or from a script after starting the service that receives the client requests.

Dedicated IP Address and Subnet Mask

Because each host's dedicated IP address is for network traffic that is not related to client requests to the cluster, Network Load Balancing never filters traffic or applies port rules to this address.

The dedicated IP address is normally the first in the list of IP addresses used by TCP/IP for the host's network connection. This ensures that outbound connections from the host use the dedicated address instead of a virtual IP address for their source address. Otherwise, replies for outbound connections could be load-balanced and delivered to another host.

In certain scenarios (such as load-balancing for virtual private networks), you do not set a dedicated IP address in TCP/IP for the host's network connection. These applications require that the cluster's primary IP address be used for outbound connections. Therefore, in these scenarios you do not need to set a value for this parameter. These scenarios are covered in "Scenarios," later in this chapter.

Port Rules

Each port rule configures load-balancing for client requests that use the port or ports covered by the Port Range parameter. How you load-balance your applications is mostly defined by how you add or modify port rules, which you create on each host for any particular port range.

For information about port assignments for common applications, such as HTTP and FTP, see Windows 2000 Network Load Balancing Help. For a complete list of TCP and UDP port assignments, see "TCP and UDP Port Assignments" in the *Microsoft® Windows® 2000 Server Resource Kit TCP/IP Core Networking Guide*.

Port Range

To load-balance all client requests with a single port rule, use the default port range (0-65535). By using the default port range, you do not have to worry about which port or ports are associated with the application whose client requests you are load-balancing.

Note For Windows Load Balancing Service on clusters that are running Windows NT version 4.0, the default port range was 1-65535. Make sure to check default port-range values when you build mixed Windows NT 4.0/Windows 2000 clusters or when you perform rolling upgrades from Windows NT 4.0 to Windows 2000.

To specify a single port, enter the same port number for both the start and the end of the range—for example, 80-80.

You might need to use multiple port rules if you load-balance multiple applications with multiple policies. For example, you might set client affinity for one application but not for another.

If you use multiple port rules, make sure that for a specific port rule, the port range covers all the ports that the application uses; protocols such as FTP use more than one port.

Each port rule configures port ranges only for contiguous port numbers. Therefore, in rare circumstances (for example, if one application is associated with two noncontiguous sets of ports, between which there is an intervening port that another application uses) you might have to define more than one port rule for the application that uses the noncontiguous ports. For example, HTTP requests (and therefore most Web requests) use port 80, and Secure Sockets Layer (SSL) requests use port 443.

Protocols

Some applications (such as streaming media applications) use both TCP and UDP ports. In most scenarios, set the Protocols parameter to **Both**.

Network Load Balancing with Network Hardware Switches

As explained in Windows 2000 Network Load Balancing Help, network adapters for Network Load Balancing hosts in a single cluster must all be on the same broadcast subnet and connected to each other through either a hub or a switch.

Network hardware switches mediate between a network and computers or other switches, routing packets from the network to the correct computer.

If you connect Network Load Balancing hosts with a switch, the switch must be level-2 rather than level-3 or higher, because all the hosts share the same IP address (the cluster IP address), and level-3 switches direct network packets (incoming client requests) according to the IP address of the destination computer.

In unicast mode, each host's unique media access control address is replaced with the same cluster media access control address. Identifying all the hosts with one media access control address makes it possible to distribute incoming client requests (network packets) to all the hosts.

However, most level-2 switches require that each port be associated with a unique source media access control address. Network Load Balancing addresses this requirement in unicast mode by enabling the MaskSourceMAC feature by default.

When MaskSourceMAC is enabled, Network Load Balancing masks the source media access control address for outgoing packets so that for each port the switch continues to see a unique source media access control address. This satisfies the switch's requirement that each port be associated with a unique media access control address. Figure 19.1 shows a representative configuration of a Network Load Balancing cluster in unicast mode, with MaskSourceMAC enabled, and attached to a level-2 switch.

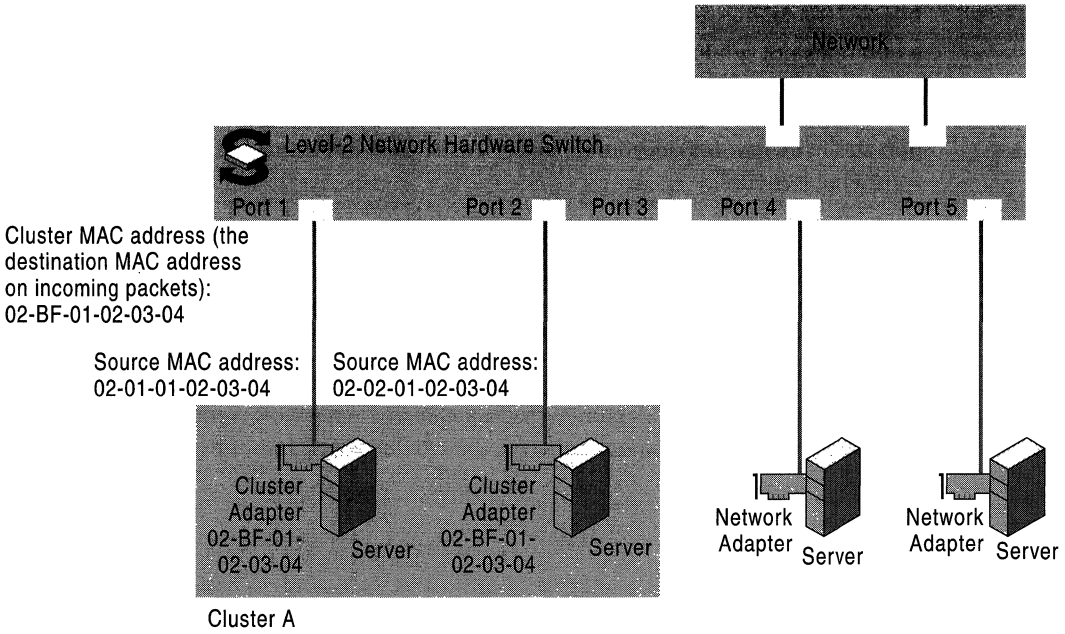


Figure 19.1 Network Load Balancing cluster running in unicast mode with MaskSourceMAC enabled

Masking the cluster media access control address on outgoing packets prevents the switch from associating the cluster media access control address with a single port. When a client request (which contains the cluster media access control address) enters the switch, the switch does not recognize the media access control address in the packet and so sends the packet to all ports. This is called "switch flooding."

In unicast mode, Network Load Balancing induces switch flooding by design, so that packets sent to the cluster's virtual IP address go to all the cluster hosts. Switch flooding is part of the Network Load Balancing strategy of obtaining the best throughput for any specific load of client requests.

If, however, the cluster shares the switch with other (noncluster) computers or other clusters, switch flooding can add to the other computers' network overhead by including them in the flooding.

You can avoid flooding noncluster computers by putting a network hub between the switch and the Network Load Balancing cluster hosts, and then disabling the MaskSourceMAC feature. The hub delivers each packet to every host, and the switch associates the cluster media access control address with a single port, satisfying the switch's requirement that each port be associated with a unique media access control address.

Placing the Network Load Balancing hosts downstream (toward the cluster) from a hub does not reduce the bandwidth for downstream packets. However, all upstream (from the cluster) traffic must flow through the hub. To optimize use of the hub, you can also connect each host's second network adapter back to another port in the switch for outbound packets, as shown in Figure 19.2. This has the following benefits:

- Routing outbound packets through network adapters that are not attached to the hub improves use of the hub's capacity.
- Use of the capacity for multiple upstream pipes from the switch to the network is improved, because multiple cluster hosts can simultaneously send traffic to different upstream pipes.
- Using two network adapters to separate each cluster host's inbound and outbound network traffic improves the cluster hosts' handling of network traffic.

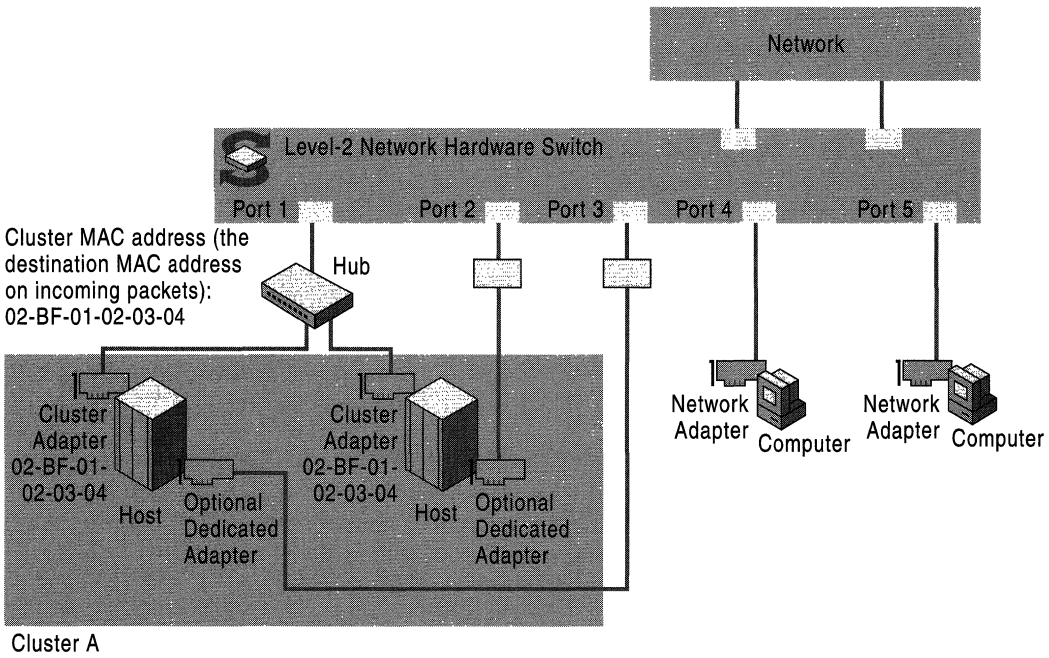


Figure 19.2 Network Load Balancing cluster running in unicast mode, with MaskSourceMAC disabled

Finally, if you choose not to use a hub as described here (for example, if the Network Load Balancing cluster does not share the level-2 switch with any other computers), you can put a level-3 switch upstream from the level-2 switch to prevent switch flooding of other interconnected level-2 switches.

Scenarios

The scenarios in this section are some representative configurations for which you might use a Network Load Balancing cluster. Each scenario includes information about configuring the cluster.

IIS Server (Web Farm)

An IIS Server Web farm is the most common scenario for Network Load Balancing. The scenarios "Servicing Multiple Web Sites (Multihoming)" and "Servicing a Web Site with Active Server Pages," which are discussed later in this chapter, are variations on the theme.

Port-rule Settings

Filtering Mode: Multiple Hosts.

Affinity: None, unless session state must be maintained; to maintain session state, either Single or Class C.

Load Weight/Equal load distribution: Equal. However, if one host has a greater capacity than the others, you can use this parameter to increase its share of the client requests.

Servicing Multiple Web Sites (Multihoming)

This scenario is a variation on the IIS Server Web farm scenario. The port-rule settings are the same; the scenarios differ in how you configure additional IP addresses for the cluster in the **Advanced TCP/IP Settings** dialog box.

For more information about port-rule settings, see "IIS Server (Web Farm)," earlier in this chapter.

Note that you enter additional virtual IP addresses in the **Advanced TCP/IP Settings** dialog box, not in the **Network Load Balancing Properties** dialog box.

Before performing the following procedure, be sure to enter the host's dedicated IP address and the cluster's primary IP address in the **Network Load Balancing Properties** dialog box and in the **Advanced TCP/IP Settings** dialog box, as described in Windows 2000 Advanced Server Help.

▶ **To host multiple Web sites with different IP addresses on a Network Load Balancing cluster**

1. Click the **Start** menu, point to **Settings**, and then open **Network and Dial-up Connections**.
2. Click the **Local Area Connection** for which Network Load balancing is enabled, and then, in the **Local Area Connection Status** dialog box, click **Properties**.
3. In the **Local Area Connection Properties** dialog box, click **Internet Protocol (TCP/IP)**, and then click **Properties**.
4. Click **Advanced**.

   The list under **IP Addresses** should already contain the host's dedicated IP address and the cluster's primary IP address. This cluster IP address corresponds to the cluster's primary IP address that you entered in the **Network Load Balancing Properties** dialog box under **Cluster parameters**.
5. For each additional virtual IP address that your cluster needs in order to run a multihomed server, click **Add**, and then enter the cluster IP address, followed by the appropriate subnet mask.

Servicing a Web Site with Active Server Pages

Web sites that use Active Server Pages (ASP) can maintain session state across client connections. Network Load Balancing helps preserve client access to session information by ensuring that all TCP/IP connections from a single client are directed to the same cluster host. To do so, set Affinity to either Single or Class C.

There are, however, situations in which a client can connect with one cluster host, and then have subsequent connections load-balanced to different hosts. Such situations include the following:

- A host is added to the cluster, and Network Load Balancing load-balances new connections from this client to the host.

  Note that existing connections are unaffected.
- Multiple client-side proxy servers cause multiple connections from the same client to originate from different IP addresses.

If either of the preceding situations arises, ASP applications must provide a means to retrieve and manage session state even if a client connects to multiple cluster hosts as part of a single session. The following are two strategies for addressing this issue:

- Use a means at the ASP level, such as a cookie, to retrieve the ASP client state across the Network Load Balancing cluster nodes.
- Encapsulate in a client-side cookie the state from a specific client request. The cookie gives the server the context for subsequent client requests. This solution works only if there is a relatively small amount of state associated with each client transaction. As state grows larger, it becomes increasingly difficult to have the client forward a large cookie to the server with every request.

For more information about port-rule settings, see "IIS Server (Web Farm)," earlier in this chapter.

Servicing a Web Site That Uses Secure Sockets Layer

If you service a Web site that uses Secure Sockets Layer (SSL) to build secure connections with clients, whether in conjunction with unsecured connections or not, create the following port rules.

Port-rule Settings

Port Range: 443, or use the default (0-65535).

Filtering Mode: Multiple Hosts

Affinity: Single

If you are load-balancing for a particular Class C address space, such as a corporate proxy array, or firewall, set Affinity to Class C.

Load Weight/Equal load distribution: Use the default.

Creating a Virtual Private Network

This scenario's discussion applies to setting up a virtual private network (VPN) by using the Point-to-Point Tunneling Protocol (PPTP).

When using Network Load Balancing with VPN servers to load-balance PPTP clients, it is important to configure the TCP/IP properties correctly to ensure compatibility with clients running earlier versions of Windows (such as Microsoft® Windows® 98 and Windows NT 4.0). To do this, assign only a single virtual IP address to the network adapter used by Network Load Balancing, and do not assign another IP address on any network adapter on this subnet. This restriction does not apply for Windows 2000 clients. Assigning only a single virtual IP address to the network adapter used by Network Load Balancing ensures that network traffic returning from the host to the client originates from the virtual IP address to which the client sent the request.

Set bindings so that Network Load Balancing is enabled for the cluster network adapter (the network adapter with the cluster's virtual address).

Note If a particular host fails, client sessions handled by that host handle will also break. Clients are prompted to log on again; their new session is handled by one of the remaining hosts.

Port-rules Settings

To provide load-balancing for virtual private network clients, use the default port rule for all hosts, as follows:

Port Range: Set the range to 0-65535, (the default). Setting the range to the default covers all the ports, so the port rule remains valid even if there is a change in the port numbers you want to cover.

Filtering Mode: Accept the default.

Affinity: Single (default).

Load Weight/Equal load distribution: Accept the default.

Streaming Media

You can use Network Load Balancing to distribute client requests among several streaming media servers.

If you add a host to the cluster, the statistical-mapping algorithm in Network Load Balancing maps some clients to the new host. Because Network Load Balancing cannot detect the start and termination of streams that use the UDP protocol, active streams for clients that are mapped to the new host mid-stream are interrupted. (This behavior does not occur for streams that use the TCP protocol.) As much as possible, add hosts to the cluster only at times that minimize possible disruptions to clients.

Port-rules Settings

Port Range: 0–65535

Filtering Mode: Multiple Hosts

Affinity: Single

Load Weight/Equal load distribution: Equal

Single-Server Failover Support

Although you can use Network Load Balancing to provide failover support for applications, managing the application as a resource in a server cluster is the preferred solution. However, if you choose to achieve failover support with Network Load Balancing, this section describes how.

In this scenario, start the application on every host to which the cluster traffic can fail over.

In all scenarios, Network Load Balancing does not restart the application on failover. It assumes that an instance of the application is running on each host in the cluster.

For Network Load Balancing to provide single-server failover support for a specific application, the files that the application uses must be simultaneously accessible to all hosts that run the application. These files normally reside on a back-end file server. Some applicationsrequire that these files be continuously open exclusively by one instance of the group; in a Network Load Balancing cluster, you cannot have two instances of a single file open for writing. These failover issues are addressed by server clusters, which run the Cluster service.

Other applications open files only on client request. For these applications, providing single-server failover support in a Network Load Balancing cluster works well. Again, the files must be visible to all cluster hosts. You can accomplish this by placing the files on a back-end file server or by replicating them across the Network Load Balancing cluster.

There are two alternatives for configuring the port rules for single-server failover support:

- Use no port rules.

  All the traffic goes to the host with the highest priority (the Host Priority ID with the lowest value). If that host fails, all the traffic switches to the host with the next-highest priority.

- For each application for which you're configuring single-server failover support, create a different port rule for the application's port range, in which:
  - Filtering Mode is set to Single.
  - Handling priorities are set according to the desired failover priority across the cluster hosts.

  This option overrides the Host Priority IDs with handling priorities for each application's port range. With this configuration, you can run two single-server applications on separate hosts and fail in opposite directions.

  For example, if applications Red and Blue are assigned Handling Priority IDs as indicated in Table 19.1, the applications will run on different hosts and fail over to different secondary hosts.

Table 19.1 Hypothetical Assignment of Handling Priority IDs

Host	Application Red's Port	Application Blue's Port
Host A	Handling Priority 1	Handling Priority 2
Host B	Handling Priority 2	Handling Priority 1

Port-rule Settings

Filtering Mode: Single host.

Affinity: Not available when filtering mode is single host.

Load Weight/Equal load distribution: Not available when filtering mode is single host.

Handling Priority: See the application issues discussion for this scenario.

Default Handling of Client Requests

Network Load Balancing is designed so that network traffic is not affected for the virtual IP address(es) of applications that are not being load-balanced. All traffic not explicitly load-balanced with port rules is sent to the default host. For example, incoming Telnet requests for the Virtual IP addresses are handled by the default host.

Therefore, if you do not want to load-balance some traffic for the virtual IP address, you do not define a port rule for it; the default host then handles all the traffic for that address. However, the default port range (0-65535) covers all ports, so you need to make sure that the port range for any port rules you define does not include ports associated with applications whose client requests you do not want to load-balance.

Note Undesired load-balancing is not an issue for dedicated IP addresses. Incoming network traffic for the dedicated IP address on each host is never affected by Network Load Balancing.

Wlbs Display Command

The **wlbs display** command-line command provides much valuable information about the Network Load Balancing host on which it is carried out. It is intended for use in diagnosing problems with NLB configurations. The information comprises the following areas:

- Details of the current cluster configuration.

  This section includes the current values for the cluster parameters, host parameters, and port rules, and other registry values for the host.

  For more information about the cluster parameters, host parameters, and port rules, see Windows 2000 Network Load Balancing Help.

- Last 10 event messages.

  The output for each message is the output of the event log. Each message is followed by two lines of hexadecimal numbers.

  For the sake of brevity, all the event messages but one were deleted from the following sample output.

- IP configuration of the host operating system and network adapter.

  This is the output of the **ipconfig** command.

- Current state of the cluster.

  This is the membership of the cluster as of the last convergence.

The following is an example of the output of the **wlbs display** command:

```
D:\>wlbs display
WLBS Cluster Control Utility V2.3. (c) 1997-99 Microsoft Corporation

=== Configuration: ===

Current time              = Thu Jul 01 13:02:23 1999
ParametersVersion         = 4
VirtualNICName            = \Device\{31270FF0-11FD-11D3-8B19-
    02BFAC1FF0AB}
AliveMsgPeriod            = 1000
AliveMsgTolerance         = 5
NumActions                = 50
NumPackets                = 100
NumAliveMsgs              = 66
ClusterNetworkAddress     = 02-bf-00-00-00-00
ClusterName               = cluster.reskit.com
ClusterIPAddress          = 0.0.0.0
ClusterNetworkMask        = 255.255.248.0
DedicatedIPAddress        = 172.31.240.170
DedicatedNetworkMask      = 255.255.248.0
HostPriority              = 1
ClusterModeOnStart        = ENABLED
LicenseKey                =
DescriptorsPerAlloc       = 512
MaxDescriptorAllocs       = 512
ScaleSingleClient         = 0
NBTSupportEnable          = 1
MulticastSupportEnable    = 0
MulticastARPEnable        = 1
MaskSourceMAC             = 1
IPToMACEnable             = 1
ConnectionCleanupDelay    = 300000
RemoteControlEnabled      = 1
RemoteControlUDPPort      = 2504
RemoteControlCode         = 0x0
RemoteMaintenanceEnabled  = 0x0
CurrentVersion            = V2.3
InstallDate               = 0x36D75CDA
VerifyDate                = 0x1C9E7553
NumberOfRules             = 1
PortRules
Start    End      Prot    Mode            Pri     Load    Affinity
   1     65535    Both    Multiple                Equal   S

00000001 0000FFFF 6FFFF001 00000002 00000003 00000001 00010001 00000000
```

```
=== Event messages: ===

#3852 ID: 0x40070024 Type: 4 Category: 0 Time: Fri Jun 11 15:29:16 1999
WLBS : registry parameters successfully reloaded.
00000070 00420042 00490041 002D004E 00310057 004E002D 00350054 00000000
00370031 002E0032 00310033 0032002E 00300034 0031002E 00310037

=== IP configuration: ===

Windows NT IP Configuration

        Host Name . . . . . . . . . . : NLB-HOST-1
        Primary Domain Name . . . . . : testclus.reskit.com
        Node Type . . . . . . . . . . : Hybrid
        IP Routing Enabled. . . . . . : Yes
        WINS Proxy Enabled. . . . . . : No

Ethernet adapter Local Area Connection:

        Adapter Domain Name . . . . . :
        DNS servers . . . . . . . . . :
        Description . . . . . . . . . : 3Com EtherLink XL 10/100 PCI TX
NIC (3C905
    B-TX)
        Physical Address. . . . . . . : 00-C0-4F-57-0E-34
        DHCP Enabled. . . . . . . . . : No
        IP Address. . . . . . . . . . : 0.0.0.0
        Subnet Mask . . . . . . . . . : 0.0.0.0
        IP Address. . . . . . . . . . : 172.31.240.170
        Subnet Mask . . . . . . . . . : 255.255.248.0
        Default Gateway . . . . . . . :

=== Current state: ===

Host 1 is stopped and does not know convergence state of the cluster.
```

Changing Network Load Balancing Resource Limits in the Registry

You can tune Network Load Balancing performance by manually modifying the following entries in the registry in the HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WLBS\Parameters subkey.

- **AliveMsgPeriod**

  Determines the period (in milliseconds) between Network Load Balancing heartbeat messages broadcast by each host.

  Default value: 1000 (one second)

  Possible range: 100–10000

- **AliveMsgTolerance**

  Determines the number of **AliveMsgPeriod** periods to wait after the last message from a host before the host is declared offline and the cluster performs convergence.

  Default value: 5

  Possible range: 5–100

- **DescriptorsPerAlloc**

  Determines the number of connection descriptors allocated at a time. Connection descriptors are used to track TCP connections.

  Default value: 512

  Possible range: 16–1024

- **MaxDescriptorAllocs**

  Determines the maximum number of times that connection descriptors can be allocated (this value limits the maximum memory footprint of Network Load Balancing).

  Default value: 512

  Possible range: 1–1024

- **NumActions**

  An internal Network Load Balancing entry. Increase the value of this entry only if you encounter an event log message that advises you to do so.

  Default value: 50

  Possible range: 5–500

- **NumPackets**

  An internal Network Load Balancing entry. Increase the value of this entry only if you encounter an event log message that advises you to do so.

  Default value: 100

  Possible range: 5–500

- **NumAliveMsgs**

  An internal Network Load Balancing entry. Increase the value of this entry only if you encounter an event log message that advises you to do so.

  Default value: 10

  Possible range: 5–500

- **MaskSourceMAC**

  Enables masking of the Source media access control address.

  If the host is connected to a switch when Network Load Balancing is running in unicast mode, set the value of **MaskSourceMAC** to 1 (the default). If the Network Load Balancing host is running in unicast mode and is attached to a hub that is connected to a switch, set the value of this entry to 0. If Network Load Balancing is running in multicast mode, this setting has no effect.

  Default value: 1

  Possible range: 0–1

- **RemoteControlUDPPort**

  Determines the UDP port that is used by Network Load Balancing to receive remote control messages. Note that for backwards compatibility, Network Load Balancing (and, on Windows NT 4.0, Windows Load Balancing Service) automatically listens to port 1717. If you decide to firewall port 2504 to block remote control messages, you also need to firewall port 1717.

  Default value: 2504

- **NetmonAliveMsgs**

  Determines whether Network Monitor (NetMon) captures Network Load Balancing heartbeat messages on the local host.

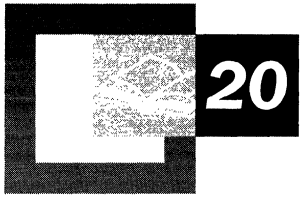
  To allow NetMon to capture Network Load Balancing heartbeat messages on the local host, set the value of this entry to 1. To get the best performance, leave the value of this entry at its default value of 0.

Additional Resources

- For information about setting up, installing, and operating a server cluster, see Windows 2000 Advanced Server Help.
- For more information about the Windows Clustering API, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

CHAPTER 20

Interpreting the Cluster Log



The cluster log is a record of the *Cluster service* (Clussvc.exe and associated processes) activity in a server cluster member. Its completeness as a diagnostic log makes it a powerful troubleshooting tool. The cluster log, also known as the diagnostic log, is enabled by default in Windows 2000. Its output is printed to a .log file in %SystemRoot%\Cluster.

Although cluster log entries might seem opaque, they open windows to a wealth of information about your cluster's inner workings when you understand the following:

- The parts of a cluster log entry.
- The different types of cluster log entries and what they describe.
- How the entries describe the detailed activity of cluster operations.

To better help you understand these things, this chapter interprets typical cluster log entries for the form and join operations.

In This Chapter

Related Information in the Resource Kit

For more information about the Cluster service, see the following:

- Microsoft® Windows® 2000 Advanced Server Help.
- *Microsoft® Windows® 2000 Server Resource Kit Deployment Planning Guide.*

Cluster Log Basics

In this chapter, the term *cluster* always refers to a server cluster, which is a group of servers linked by Cluster service that provides high availability and application failover support. The Cluster service comprises Clussvc.exe and associated dynamic-link libraries (DLLs) and executable files.

Cluster activity is the sum of any and all events that take place in a cluster. Cluster activity includes major blocks of activity, such as the initialization, joining, and forming operations.

The cluster log is a diagnostic log that is a more complete record of cluster activity than the Microsoft Windows 2000 event log; the cluster log records the Cluster service activity that leads up to the events recorded in the event log. Although the event log can point you to a problem, the cluster log helps you get at its root. So, for diagnosis, check the event log first, then the cluster log. For more information about correlating the entries in the event log and the cluster log, see "Correlating the Windows 2000 Event Log and the Cluster Log" later in this chapter.

Note The cluster log records only the events of a single member in a server cluster. It does not record events for network load balancing. To record all the events in a Cluster service cluster, you must enable logging for each member of the cluster.

To successfully interpret the cluster log, you need to be familiar with the following areas:

- Anatomy of a cluster log entry.
- Meanings of abbreviations such as [DM], [RGP], and [JOIN].
- Meanings of state codes.
- Techniques for tracking the source of a problem.

Anatomy of a Cluster Log Entry

Though cluster log entries might seem impenetrable at first glance, they are actually fairly easy to parse. Consider the following entry, the first line in the body of a typical cluster log:

```
378.32c::1999/06/09-18:00:18.874 Cluster service started - Cluster Node
Version 3.2051
```

Its main elements, common to every line of the log, include the following, starting from the beginning of the entry:

- The IDs of the process and thread issuing the log entry. These two IDs are concatenated, separated by a period. In the preceding example, the Process ID is 378, and the thread ID is 32c.
- The timestamp in the following format, in Greenwich Mean Time (GMT):

  yyyy/mm/dd-hh:mm:ss.sss

  where:

  - yyyy/mm/dd represents the year, month, and day.
  - hh:mm:ss.sss represents the time of day on a 24-hour clock, carried out to the thousandths of a second.

  In the preceding example, the time is 18.874 seconds past 6 o'clock in the evening, GMT.
- The event description, such as "Cluster service started."

There are two types of cluster log entries: component event log entries and resource DLL log entries.

Component Event Log Entries

The Cluster service comprises a number of components, such as the Database Manager and the Global Update Manager. The logging of their interactions is what makes the cluster log such a powerful diagnostic tool.

Here is a typical example of a cluster log entry for a component event:

```
378.380::1999/06/09-18:00:50.881 [NM] Forming cluster membership.
```

Entries describing component events follow the process/thread ID and timestamp with the following:

- One of two types of abbreviations, enclosed in square brackets, such as [NM] or [JOIN]. The two types of abbreviations are:
  - The component that wrote the event to the cluster log (in this entry, it is [NM], the Node Manager).
  - The state of the node at the time the entry was written to the cluster log.

    In an abbreviation, the node state reflects the operation that is in progress, such as [INIT] or [JOIN].

  A cluster log abbreviation can combine a component and state, as in [NMJOIN], which combines the Node Manager abbreviation and the join operation:

  ```
  388.55c::1999/06/09-18:08:25.621 [NMJOIN] Processing request by node
   2 to begin joining.
  ```

- The event description (for example, "Found the quorum resource 254ef0e8-1937-11d3-b3fe-00a0c986aa14."). In the example, "254ef0e8-1937-11d3-b3fe-00a0c986aa14" is the globally unique identifier (GUID) for the resource that the component found. When the Cluster service creates a resource, one of the entries typically names the resource, which makes it easy to subsequently identify the resource in event descriptions by its GUID. The following is one such entry:

  ```
  378.380::1999/06/09-18:00:51.193 [FM] Name for Resource
    254ef0e5-1937-11d3-b3fe-00a0c986aa14 is 'Cluster IP Address'.
  ```

Meanings of Abbreviations

Cluster log abbreviations for components and node states are shown in Table 20.1.

Table 20.1 Cluster Log Abbreviations for Components and Node States

Abbreviation	Node state or component
[API]	API support. These entries come from the Cluster service component that provides support for the Server Cluster API.
[ClMsg]	Cluster messaging. The component that Regroup (also known as Membership Manager — see later in this table) uses to send and receive its messages.
[ClNet]	Cluster network engine. Generic code to determine a node's network configuration.
[CP]	Checkpoint Manager. If a resource has its registry key registered for checkpointing, the Checkpoint Manager monitors any changes to the key while the resource is online and writes a checkpoint to the quorum disk whenever there is a change to the registered key. On the node to which the resource is being failed over, the resource key in the registry is updated with the resource key's checkpoint before the resource is brought online.
[CS]	Cluster service. This abbreviation is assigned to messages that come out of the Cluster service rather than one of its components.
[DM]	Database Manager. The agent through which other components read or make changes to the cluster configuration database.
[EP]	Event Processor. Components of the Cluster service register with the Event Processor to receive internal cluster events, such as a node's going up or down.
[FM]	Failover Manager. Coordinates the moving of a group from one node to another based on failure criteria specified by the group's properties.
[GUM]	Global Update Manager. A cluster-wide, broadcast-like remote procedure call (RPC) mechanism used to distribute information to all nodes in the cluster.
[INIT]	The initial state of a node prior to joining or forming a cluster.
[JOIN]	The node state that follows [INIT] when the node attempts to join a cluster. If the join operation succeeds, the state of the node then moves to cluster member.
[LM]	Log Manager. Maintains the quorum log.
[MM]	Membership Manager, also known and written to the cluster log as Regroup ([RGP]). See [RGP] in this table.

(continued)

Table 20.1 Cluster Log Abbreviations for Components and Node States *(continued)*

Abbreviation	Node state or component
[NM]	Node Manager. Keeps track of the state of other nodes in the cluster as well as maintaining the cluster-wide network configuration.
[OM]	Object Manager. Maintains an in-memory database of entities, or objects (nodes, networks, groups, and so on). Each object has an associated type and a set of methods with which other components can manipulate it. Each cluster object is represented in the Object Manager space. The Object Manager does not differentiate between types of objects.
[RGP]	Regroup, also known and written to the cluster log as Membership Manager ([MM]). Tracks which nodes are members of the cluster. Regroup writes entries to the log during initialization, form operations, and join operations, and when cluster membership changes.
[RM]	Resource Monitor. Any of the processes (instances of Resrcmon.exe) of the Cluster service that actually monitor individual resources.

Resource DLL Log Entries

Because resource groups are the basic unit of failover, resource DLL entries are key to understanding cluster activity. The following entry is a cluster log entry for a resource DLL event, in this case one of the entries from the disk arbitration process.

```
15c.458::1999/06/09-18:00:47.897 Physical Disk <Disk D:>: [DISKARB]
 Arbitration Parameters (1 9999).
```

Instead of listing an abbreviated component name between the timestamp and event description as component log entries do, entries describing resource DLL events list the following information:

- Resource type ("Physical Disk")
- Resource name ("<Disk I:>")

The event description in this example is "[DISKARB] Arbitration Parameters (1 9999)."

Meanings of State Codes and Status Codes

Interpreting state and status codes is crucial to deciphering the cluster log. Doing so is not difficult. The following two procedures tell you how.

▶ To find the meaning of the status codes

- At the command prompt, type:

  ```
  net helpmsg [error_number]
  ```

  For example, in the following entry, "error 5" is the error number. Using **net hlpmsg** returns "Access is denied." The error indicates that the problem has to do with permissions.

  ```
  388.4e8::1999/06/09-20:20:57.281 [NM] Received advice that node 2 has
   failed with error 5.
  ```

Remember that you still need to study the context of the error to discover its cause.

▶ To find the meaning of state codes

1. From the event description in the cluster log, note the type of object — group, resource, node state, network, or net interface — associated with the entry.

   Note In Windows 2000, the resource name is logged with its GUID when the Cluster service is started. You can expect to find the GUID and the resource name associated in a single entry during resource creation if these log entries have not been overwritten.

   You can also find the resource name in the registry, in the resource's subkey, which is identified by the resource's GUID. For more information about finding the resource name in the registry, see "Identifying GUIDs in the Registry" later in this chapter.

2. In "State Codes" later in this chapter, find the meaning in the appropriate table.

For example, in the following entry, "state 129" means ClusterResourceOnlinePending.

```
388.398::1999/06/09-18:07:45.295 [FM] FmpRmOnlineResource: Returning.
 Resource 254ef0e8-1937-11d3-b3fe-00a0c986aa14, state 129, status
  997.
```

Techniques for Tracking the Source of a Problem

It's often helpful to go to the end of the log and work your way backward to find the oldest failure that caused your problem. However, the true cause of the failure might not be the first apparent cause you find when working backward. The initial failure can cause side effects that trigger other failures.

You can also use the following for tracking operations in the cluster log:

- Timestamps
- GUIDs
- Resources
- Groups
- Process and thread Ids
- GUM sequence numbers
- The gdwQuoBlockingResources global variable counts of shared locks

Timestamps

Scan the timestamps for breaks of one second or more to find periods of inactivity in the Cluster service. A break might indicate that all threads of activity have completed.

For example, consider the following entries from the logging of a cluster form operation and cluster join operation.

```
378.328::1999/06/09-18:00:52.006 [FM] FmpPropagateResourceState:
 resource 76d0bf88-5e2c-44b9-9bdb-3eb677487afb online event.
378.328::1999/06/09-18:00:52.006 [FM] FmpRmWorkItemHandler: Exit
378.3a8::1999/06/09-18:00:52.006 [GUM] GumSendUpdate:  Locker waiting
 type 0 context 11
378.3a8::1999/06/09-18:00:52.006 [GUM] Thread 0x3a8 UpdateLock wait on
 Type 0
378.3a8::1999/06/09-18:00:52.006 [GUM] DoLockingUpdate successful, lock
 granted to 1
378.3a8::1999/06/09-18:00:52.006 [GUM] GumSendUpdate: Locker dispatching
 seq 1673 type 0 context 11
378.3a8::1999/06/09-18:00:52.006 [GUM] GumpDoUnlockingUpdate releasing
 lock ownership
378.3a8::1999/06/09-18:00:52.006 [GUM] GumSendUpdate: completed update
 seq 1673 type 0 context 11
378.3c8::1999/06/09-18:00:53.053 [GUM] GumSendUpdate:  Locker waiting
 type 1 context 4098
```

The first two entries report the Failover Manager's initiation of global update, and the second-to-last entry reports the completion of the update. Note the break in time between this entry and the last entry.

Occasionally, timestamps violate chronological order. For example, note that in the following example from a Microsoft® Windows NT® 4.0 cluster, the entry with the timestamp 01-20:43:09.257 ("[GUM] Thread 0x7f UpdateLock wait on Type 1") appears out of order.

```
07b::01-20:43:09.304 [GUM] GumSendUpdate: completed update seq 1191
 type 1 context 0
07f::01-20:43:09.257 [GUM] Thread 0x7f UpdateLock wait on Type 1
07f::01-20:43:09.304 [GUM] DoLockingUpdate successful, lock granted to 2
```

This happened because the cluster log obtains the system time and then tries to acquire access to the log file (also known as acquiring the print lock). In this case, thread 07b acquired the print lock before thread 07f, which had the time but not the print lock.

To find why 07f didn't acquire the print lock when it got the time, scan for the 07f thread ID. The following two entries show that 07f was interrupted by 0ae in the middle of a GUM update:

```
07f::01-20:43:09.257 [GUM] GumSendUpdate:  Locker waiting type 1
 context 4098
0ae::01-20:43:09.265 [GUM] GumSendUpdate:  Locker waiting type 0
 context 8
```

GUIDs, Resources, and Groups

Because the Cluster service often doesn't include the resource's display name in entries written to the cluster log, you need to track resource GUIDs as well as display names. The entry reporting an offline call for a given resource reliably includes both the display name and the GUID, so it is a good resource for correlating a resource's GUID and display name.

For example, the following entry tells you that resource "254ef0e5-1937-11d3-b3fe-00a0c986aa14" is the Cluster IP Address resource:

```
378.130::1999/06/09-18:02:24.189 [FM] FmpOfflineResource: Offline
 resource <Cluster IP Address> <254ef0e5-1937-11d3-b3fe-00a0c986aa14>
```

Another dependable correlation of a resource's display name and GUID is the entry the Failover Manager [FM] writes during resource creation, which names the resource. The following is an example:

```
378.380::1999/06/09-18:00:51.209 [FM] Name for Resource
  191c151b-a8e5-45a6-8330-d38669845e66 is 'testwinfile'.
```

The Cluster service's model for bringing resources online is to bring everything online as quickly as possible. If there are hundreds of resources, the Cluster service tries to bring them all online simultaneously.

Creating Groups and Resources

The following set of entries log successful creation of a group and a resource at startup of the Cluster service. In both cases, you must track the object by its GUID. Note, however, the entry that gives the display "Name for" the group or resource. Note also that the Failover Manager is the component responsible for creating and initializing groups and resources at startup.

In the first two entries, the Failover Manager creates the group in its memory and initializes it with the values for the group in the cluster registry key.

```
378.380::1999/06/09-18:00:51.209 [FM] Creating group

d103ed17-f7bc-4ff9-8a64-e692f4ecb663
378.380::1999/06/09-18:00:51.209 [FM] Initializing group
  d103ed17-f7bc-4ff9-8a64-e692f4ecb663 from the registry.
```

The following is the "Name for" entry for the group.

```
378.380::1999/06/09-18:00:51.209 [FM] Name for Group
  d103ed17-f7bc-4ff9-8a64-e692f4ecb663 is 'test group'.
```

The Failover Manager creates the list of the nodes that are the group's preferred owners.

```
378.380::1999/06/09-18:00:51.209 [FM] Group
  d103ed17-f7bc-4ff9-8a64-e692f4ecb663 preferred owner 1.
378.380::1999/06/09-18:00:51.209 [FM] Group
  d103ed17-f7bc-4ff9-8a64-e692f4ecb663 preferred owner 2.
```

The Failover Manager creates the list of the resources that the group owns.

```
378.380::1999/06/09-18:00:51.209 [FM] Group
  d103ed17-f7bc-4ff9-8a64-e692f4ecb663 contains Resource
 191c151b-a8e5-45a6-8330-d38669845e66.
```

Having created the group, the Failover Manager creates resource 191c151b-a8e5-45a6-8330-d38669845e66 and then initializes it with its values from the cluster database (the cluster key in the registry).

```
378.380::1999/06/09-18:00:51.209 [FM] Creating resource

191c151b-a8e5-45a6-8330-d38669845e66
378.380::1999/06/09-18:00:51.209 [FM] Initializing resource
  191c151b-a8e5-45a6-8330-d38669845e66 from the registry.
```

The Failover Manager creates the list of possible owners for the resource.

```
378.380::1999/06/09-18:00:51.209 [FM] FmpAddPossibleEntry: adding node 1
 as possible host for resource 191c151b-a8e5-45a6-8330-d38669845e66.
378.380::1999/06/09-18:00:51.209 [FM] FmpAddPossibleEntry: adding node 2
 as possible host for resource 191c151b-a8e5-45a6-8330-d38669845e66.
```

The Failover Manager notes that the resource's dependencies have already been created.

```
378.380::1999/06/09-18:00:51.209 [FM] All dependencies for resource
 191c151b-a8e5-45a6-8330-d38669845e66 created.
```

Process and Thread Ids

Process and thread Ids help you trace activities and interactions. For example, in the section "Joining a Cluster (Unsuccessful Attempt)" later in the chapter, each thread spawned to connect to the sponsoring node has a distinct thread ID.

When an instance of an executable file (such as the Cluster service and Resource Monitor) starts, it starts a new process and, therefore, gets a new process ID. Knowing this, you can examine the activity with which a process ID is associated in order to deduce which executable file's process it represents.

Tracing process and thread Ids is especially helpful for finding out why a particular process is taking too long.

One caveat, however: Related events can have different thread Ids in the cluster log, even though they report activity that occurs as a single logical operation.

GUM Updates and Sequence Numbers

One of the most frequently repeated patterns you'll find in cluster logs is that of a global update, performed by the Global Update Manager (GUM) when there is a change to the cluster database. The following group of entries is a typical global update.

The GUM uses a unique sequence number for each transaction, incrementing the sequence number by one for each new update. In this group of entries, "1657" is the sequence number. Note that this update propagates two changes: a membership change (type 2 context 17) and a resource-state change (type 0 context 8). For context numbers of GUM updates, see "Context Numbers" later in this chapter.

```
378.380::1999/06/09-18:00:50.897 [GUM] GumSendUpdate:  Locker waiting
 type 2 context 17
378.380::1999/06/09-18:00:50.897 [GUM] Thread 0x380 UpdateLock wait on
 Type 2
378.380::1999/06/09-18:00:50.897 [GUM] DoLockingUpdate successful, lock
 granted to 1
378.380::1999/06/09-18:00:50.897 [GUM] GumSendUpdate: Locker dispatching
 seq 1657    type 2 context 17
378.380::1999/06/09-18:00:50.897 [dm] DmBeginLocalUpdate Entry
378.380::1999/06/09-18:00:50.897 [dm] DmBeginLocalUpdate Exit,
 pLocalXsaction=0x000a0dd8 dwError=0x00000000
```

```
378.380::1999/06/09-18:00:47.928 [GUM] GumSendUpdate: Locker dispatching
 seq 1657    type 0 context 8
378.380::1999/06/09-18:00:47.928 [GUM] GumpDoUnlockingUpdate releasing
 lock ownership
378.380::1999/06/09-18:00:47.928 [GUM] GumSendUpdate: completed update
 seq 1657    type 0 context 8
```

GUM sequence numbers are useful for correlating the activities of two or more nodes in a cluster. For instance, a "Locker dispatching seq *xxx*" message has a corresponding "s_GumQueueLockingUpdate: dispatching seq *xxx*" in the other node's log, where "*xxx*" represents the same sequence number. This can help you synchronize time periods between two logs. In the following pair of examples, note the continuity of the sequence number ("seq 1732"). Note also that the timestamps for the two nodes are not exactly synchronized: The timestamp for the "GumSendUpdate: Dispatching" entry on the sending node is later than the timestamp for the "GumQueueLockingUpdate: Dispatching" entry on the receiving node. Thus, you need the sequence number to correlate the update between the two nodes.

The following entries were logged on the node issuing a GUM update:

```
378.390::1999/06/09-18:16:29.671 [GUM] GumSendUpdate: Dispatching

seq 1732    type 2 context 17 to node 2
.
.
.
378.390::1999/06/09-18:16:29.718 [GUM] GumSendUpdate: completed update
  seq 1732    type 2 context 17
.
.
.
378.390::1999/06/09-18:16:29.671 [GUM] GumSendUpdate: Dispatching
    seq 1732    type 2 context 17 to node 2
```

The following entries were logged on the node receiving the same GUM update:

```
388.55c::1999/06/09-18:16:29.600 s_GumQueueLockingUpdate: dispatching
 seq 1732    type 2 context 17
.
.
.
388.55c::1999/06/09-18:16:29.662 s_GumQueueLockingUpdate: completed
 update seq 1732    type 2 context 17 result 0
```

Shared Locks and gdwQuoBlockingResources

When something in your cluster is preventing the Quorum resource from being taken offline (as in a failover or move of the Quorum resource from one node to the other), you can use the gdwQuoBlockingResources count of shared locks to identify the root cause of the problem.

A lock (called the quorum lock) prevents access to the Quorum resource by components other than the holder of the lock; there are two modes for the lock on the Quorum resource:

- Shared lock

  Dependent resources acquire the quorum lock in shared mode when they need to access the quorum device and therefore need the quorum device to stay online. A shared lock prevents the Cluster service from taking the Quorum resource offline.

- Exclusive lock

  The Cluster service acquires the quorum lock in exclusive mode when the Quorum resource is going to go offline. The information in the Quorum resource can only be modified by a thread that acquired the quorum lock in exclusive mode.

Clearly, acquisition of an exclusive lock must not supersede existing shared locks. That is, the Cluster service cannot acquire an exclusive lock and take the Quorum resource offline while components are in the middle of operations that require the Quorum resource to be online.

On the other hand, after existing operations that require access to the quorum device have completed their tasks, the Cluster service must have an opportunity to take the Quorum resource offline.

The Cluster service solves this problem with the gdwQuoBlockingResources global variable and a rule. The gdwQuoBlockingResources global variable is a count of the number of existing shared locks, which functions as follows:

- When a process acquires a shared lock, the gdwQuoBlockingResources count increases by one.
- When the process releases the shared lock, the gdwQuoBlockingResources count decreases by one.
- When the gdwQuoBlockingResources count reaches zero, the Cluster service acquires the exclusive lock and is free to take the Quorum resource offline.

The rule is that a request for an exclusive lock is queued behind existing shared locks.

If there is a request to take the Quorum resource offline and the count is not zero, the InterlockedIncrement and InterlockedDecrement entries include the count of currently shared locks.

The following "InterlockedIncrement/InterlockedDecrement" pair of entries bracket the acquisition of shared locks by the Network Name resource as the resource is taken offline. If the resource could not be taken offline, preventing the gdwQuoBlockingResources count from reaching zero, you would find the problem with the resource here, between the "InterlockedIncrement/InterlockedDecrement" pair of entries.

```
388.4e8::1999/06/09-20:20:56.640 [FM] FmpRmOfflineResource:
 InterlockedIncrement on gdwQuoBlockingResources for resource
 254ef0e6-1937-11d3-b3fe-00a0c986aa14
4b4.450::1999/06/09-20:20:56.906 Network Name <Cluster Name>: Taking
 resource offline...
4b4.450::1999/06/09-20:20:56.906 Network Name <Cluster Name>: Offline of
 resource continuing...
4b4.450::1999/06/09-20:20:57.078 Network Name <Cluster Name>: Failed to
 delete server name CLUSTER1, status 2114.
4b4.450::1999/06/09-20:20:57.078 Network Name <Cluster Name>: Failed to
 delete server name CLUSTER1, status 2114.
4b4.450::1999/06/09-20:20:57.078 Network Name <Cluster Name>: Deleted
 workstation name CLUSTER1 from transport 0.
4b4.450::1999/06/09-20:20:57.187 Network Name <Cluster Name>: Resource
 is now offline
388.4e8::1999/06/09-20:20:57.187 [FM] FmpRmOfflineResource:
 InterlockedDecrement on gdwQuoBlockingResources for resource
 254ef0e6-1937-11d3-b3fe-00a0c986aa14
```

The next section discusses what the cluster log reveals about cluster activities.

Cluster Form and Join Operation Entries

As you read entries in the cluster log, you notice the beginnings and endings of operations, even though there are no divisions between the entries — it goes right from the last entry of an unsuccessful join to the first of a successful form, if that's the next loggable event.

The following operations span large groups of cluster log entries:

- Initializing the node
- Joining a cluster (unsuccessful attempt)
- An attempt by a node to form a cluster
- Activity after a successful join or form operation

The following sections look in detail at the entries for each of those operations.

Initializing the Node

The following entries represent initialization of the local node.

```
378.32c::1999/06/09-18:00:18.874 Cluster Service started - Cluster Node
 Version 3.2051
378.32c::1999/06/09-18:00:18.874                          OS Version
 5.0.2051
```

Note that the preceding entries include, with the time the Cluster service started, the version number of the Cluster service and of the node's operating system.

```
378.380::1999/06/09-18:00:18.874 [CS] Service Starting...
378.380::1999/06/09-18:00:19.210 [EP] Initialization...
378.380::1999/06/09-18:00:19.218 [DM]: Initialization
378.380::1999/06/09-18:00:19.226 [DM]: Loading cluster database from
 C:\WINNT\cluster\CLUSDB
```

In the preceding entry, the Database Manager loads the cluster database into the local registry. Later, the Database Manager updates the cluster's registry data with any cluster database checkpoints or quorum log change records that are more recent than the version of the cluster database that it just loaded into the cluster registry key.

```
378.380::1999/06/09-18:00:19.382 [DM] DmpStartFlusher: Entry
378.380::1999/06/09-18:00:19.382 [DM] DmpStartFlusher: thread created
378.380::1999/06/09-18:00:19.406 [NM] Initializing...
378.380::1999/06/09-18:00:19.429 [NM] Local node name = NODE1.
378.380::1999/06/09-18:00:19.429 [NM] Local node ID = 1.
```

The last two, preceding entries identify the node the name and ID of the node whose activity this log tracks. This identity is important for tracking interactions in the various nodes' cluster logs.

```
378.380::1999/06/09-18:00:19.429 [NM] Creating object for node 1 (NODE1)
378.380::1999/06/09-18:00:19.429 [NM] Initializing networks.
378.380::1999/06/09-18:00:19.437 [NM] Initializing network interfaces.
378.380::1999/06/09-18:00:19.609 [NM] Initialization complete.
378.380::1999/06/09-18:00:19.632 [FM] Starting worker thread...
378.3a8::1999/06/09-18:00:19.632 [FM] Worker thread running
378.380::1999/06/09-18:00:19.632 [API] Initializing
378.380::1999/06/09-18:00:19.632 [lm] :LmInitialize Entry.
378.380::1999/06/09-18:00:19.640 [lm] :TimerActInitialize Entry.
378.380::1999/06/09-18:00:19.640 [CS] Service Domain Account =
 ITRESKIT\administrator
378.380::1999/06/09-18:00:19.640 [CS] Initializing RPC server.
378.380::1999/06/09-18:00:19.734 [INIT] Attempting to join cluster
 CLUSTER1
```

After it is initialized, the Cluster service immediately tries to join a cluster.

Joining a Cluster (Unsuccessful Attempt)

The entries in this section are the record of a failed JOIN operation. They record the attempts of the joining node to find a sponsor from the available addresses and node names.

The first entry is a state entry, reporting the [INIT] state of the node attempting to join.

```
378.380::1999/06/09-18:00:19.734 [INIT] Attempting to join cluster
 CLUSTER1
```

In all the following entries, the state of the node is [JOIN].

The following nine entries record the spawning of threads that try to connect to the sponsor node and record the node's requests to join the cluster. The service always tries to join a cluster through the following:

- Cluster's IP address (172.16.64.254).
- Sponsor's name (NODE2).
- Network interfaces that are marked for internal cluster communication on the sponsor node. In this example, that includes the sponsor node's IP addresses 172.16.64.2 and 10.10.0.2.

```
378.380::1999/06/09-18:00:19.742 [JOIN] Spawning thread to connect to
 sponsor 172.16.64.254
378.3c4::1999/06/09-18:00:19.742 [JOIN] Asking 172.16.64.254 to sponsor
 us.
378.380::1999/06/09-18:00:19.742 [JOIN] Spawning thread to connect to
 sponsor 172.16.64.2
378.3c8::1999/06/09-18:00:19.742 [JOIN] Asking 172.16.64.2 to sponsor
 us.
378.380::1999/06/09-18:00:19.742 [JOIN] Spawning thread to connect to
 sponsor 10.10.0.2
378.3cc::1999/06/09-18:00:19.742 [JOIN] Asking 10.10.0.2 to sponsor us.
378.380::1999/06/09-18:00:19.742 [JOIN] Spawning thread to connect to
 sponsor NODE2
378.380::1999/06/09-18:00:19.742 [JOIN] Waiting for all connect threads
 to terminate.
378.3d0::1999/06/09-18:00:19.742 [JOIN] Asking NODE2 to sponsor us.
```

The following entries report the failures (or successes) of the requests to join. Using the **net helpmsg** *error number* command, you can find out that "status=1722" means "The RPC server is unavailable." In other words, the node attempting to join cannot contact the Cluster service on the target node.

Note the consistency of the process ID in both the request entry and response entries. Note also the consistency of the thread ID for each thread generated to a given sponsor.

```
378.3c8::1999/06/09-18:00:46.600 [JOIN] Sponsor 172.16.64.2 is not
 available (JoinVersion), status=1722.
378.3c4::1999/06/09-18:00:46.600 [JOIN] Sponsor 172.16.64.254 is not
 available (JoinVersion), status=1722.
378.3cc::1999/06/09-18:00:46.600 [JOIN] Sponsor 10.10.0.2 is not
 available (JoinVersion), status=1722.
378.3c8::1999/06/09-18:00:46.600 [JOIN] JoinVersion data for sponsor
 172.16.64.2 is invalid, status 1722.
378.3c4::1999/06/09-18:00:46.600 [JOIN] JoinVersion data for sponsor
 172.16.64.254 is invalid, status 1722.
```

```
378.3cc::1999/06/09-18:00:46.600 [JOIN] JoinVersion data for sponsor
 10.10.0.2 is invalid, status 1722.
378.3d0::1999/06/09-18:00:46.615 [JOIN] Sponsor NODE2 is not available
 (JoinVersion), status=1722.
378.3d0::1999/06/09-18:00:46.615 [JOIN] JoinVersion data for sponsor
 NODE2 is invalid, status 1722.
```

All the threads are terminated; failure to find a sponsor for joining a cluster is reported; and, consequently, failure of the attempted join is reported:

```
378.380::1999/06/09-18:00:46.615 [JOIN] All connect threads have
 terminated.
378.380::1999/06/09-18:00:46.615 [JOIN] Unable to connect to any sponsor
 node.
378.380::1999/06/09-18:00:46.615 [INIT] Failed to join cluster, status
 53
```

Next, the Cluster service on this node tries to form a cluster.

Forming a Cluster

Forming a cluster involves the following stages:

- Starting an instance of the Resource Monitor (Resrcmon.exe).
- Bringing the Quorum resource online, including the following:
  - Applying the quorum log changes to the cluster database.
  - Recreating groups and resources in the cluster database.

    The Cluster service might have used stale object information at startup; now it needs to destroy and then recreate all the group and resource objects in order to refresh their information.
- Configuring the networks.
- Bringing resources online, which might involve updating resources' registry keys.

  The cluster can be successfully formed before all the resources have been brought online.

The form operation starts at the following entry:

```
378.380::1999/06/09-18:00:46.615 [INIT] Attempting to form cluster
  CLUSTER1
```

The form operation ends at the following entry:

```
378.380::1999/06/09-18:00:51.787 [INIT] Successfully formed a cluster.
```

The following sections discuss some of the significant entries in a form operation.

Starting Resrcmon.exe

The following entry describes starting the Resource Monitor:

```
15c.0d4::1999/06/09-18:00:46.694 [RM] Main: Initializing.
```

Bringing the Quorum Resource Online

The entries in this section describe the following:

- Creating the group that contains the Quorum resource.
- Creating the Physical Disk resource that is the Quorum resource.
- Initializing the Quorum resource.
- Bringing the Quorum resource online.

The following entries describe bringing online the group that is going to contain the Quorum resource. Note the following:

- The component that creates and initializes the group is the Failover Manager.
- In the second entry, the group is initialized using the values from the local registry when the node forms the cluster.
- The third entry identifies the group.
- In the fourth entry, the Failover Manager reports learning that the group contains resource 254ef0e8-1937-11d3-b3fe-00a0c986aa14.

```
378.380::1999/06/09-18:00:47.069 [FM] Creating group
  254ef0e7-1937-11d3-b3fe-00a0c986aa14
378.380::1999/06/09-18:00:47.084 [FM] Initializing group
  254ef0e7-1937-11d3-b3fe-00a0c986aa14 from the registry.
378.380::1999/06/09-18:00:47.100 [FM] Name for Group
  254ef0e7-1937-11d3-b3fe-00a0c986aa14 is 'Disk Group 1'.
378.380::1999/06/09-18:00:47.100 [FM] Group
  254ef0e7-1937-11d3-b3fe-00a0c986aa14 contains Resource
  254ef0e8-1937-11d3-b3fe-00a0c986aa14.
```

In the following entries, resource 254ef0e8-1937-11d3-b3fe-00a0c986aa14 (the Physical Disk resource, which is the Quorum resource) is created and initialized using the values from the registry. (Note the entry that gives the display name of the resource.) The cluster key configuration for this resource specifies the following:

- Which nodes are added as possible owners for failover support.
- Any resource dependencies that need to be created.

```
378.380::1999/06/09-18:00:47.100 [FM] Creating resource
  254ef0e8-1937-11d3-b3fe-00a0c986aa14
378.380::1999/06/09-18:00:47.100 [FM] Initializing resource
  254ef0e8-1937-11d3-b3fe-00a0c986aa14 from the registry.
378.380::1999/06/09-18:00:47.100 [FM] Name for Resource
  254ef0e8-1937-11d3-b3fe-00a0c986aa14 is 'Disk D:'.
378.380::1999/06/09-18:00:47.100 [FM] FmpAddPossibleEntry: adding node 1
  as possible host for resource 254ef0e8-1937-11d3-b3fe-00a0c986aa14.
```

Using the cluster configuration data in the registry, the Failover Manager creates the resources on which the Physical Disk resource (D☺ depends.

```
378.380::1999/06/09-18:00:47.100 [FM] All dependencies for resource
  254ef0e8-1937-11d3-b3fe-00a0c986aa14 created.
```

In the following entry, the node arbitrates for the ownership of the Quorum resource.

```
378.380::1999/06/09-18:00:47.115 [FM] arbitrate for quorum resource id
  254ef0e8-1937-11d3-b3fe-00a0c986aa14.
```

Note that in adding possible owners for the Quorum resource, there are no other nodes presently in the cluster. This fact explains the following two entries.

```
378.380::1999/06/09-18:00:47.115 [FM] FmpAddPossibleNode: adding node 1
  to resource type's possible node list
378.380::1999/06/09-18:00:47.115 [FM] FmpAddPossibleNode: Warning, node
  2 not found
```

The following entry reports adding the Quorum resource to the Resource Monitor's list of resources to watch and help manage. This entry occurs for each resource as it is added to the list.

```
378.380::1999/06/09-18:00:47.115 [FM] FmpRmCreateResource: creating
  resource 254ef0e8-1937-11d3-b3fe-00a0c986aa14 in shared resource
  monitor
```

In the following entry, the PnP window is the Plug and Play window, which gets messages that indicate changes in the disk configuration.

```
15c.46c::1999/06/09-18:00:47.740 Physical Disk: PnP window created
  successfully.
```

In the following entry, the Cluster service makes node 1 the Quorum resource owner.

```
378.380::1999/06/09-18:00:47.897 [MM] MmSetQuorumOwner(1,1), old owner
  0.
```

Of the entries that report disk arbitration, the significant ones are the following. In the second entry, the node's ownership of the Quorum resource has been confirmed.

```
15c.458::1999/06/09-18:00:47.912 Physical Disk <Disk D:>:
  [DISKARB]Assume ownership of a quorum device.
15c.458::1999/06/09-18:00:47.912 Physical Disk <Disk D:>:
  [DISKARB]Arbitrate returned status 0.
```

As the Quorum resource is brought online, the following [GUM] entries report a successful GUM update. GUM relies on one node to be the master when issuing updates to the other nodes. This master is called the "locker."

Note that the sequence number is 1657. You can use this sequence number to track the update.

```
378.380::1999/06/09-18:00:47.928 [GUM] GumSendUpdate:  Locker waiting
  type 0 context 8
378.380::1999/06/09-18:00:47.928 [GUM] Thread 0x380 UpdateLock wait on
  Type 0
378.380::1999/06/09-18:00:47.928 [GUM] DoLockingUpdate successful, lock
  granted to 1
378.380::1999/06/09-18:00:47.928 [GUM] GumSendUpdate: Locker dispatching
  seq 1657    type 0 context 8
378.380::1999/06/09-18:00:47.928 [GUM] GumpDoUnlockingUpdate releasing
  lock ownership
378.380::1999/06/09-18:00:47.928 [GUM] GumSendUpdate: completed update
  seq 1657    type 0 context 8
```

The description "type 0 context 8" occurs frequently in cluster logs. "Type 0 context 8" means that GUM is performing an update for the Failover Manager and that the update concerns a change in the state of a resource (in this case, the Quorum resource). Here's why:

- In the string, "type" refers to update type. The three update types are shown in Table 20.2.

Table 20.2 Update Types

Type Code	Type
0	GumUpdateFailoverManager
1	GumUpdateRegistry
2	GumUpdateMembership

- Each update type contains a certain number of routines. The context number indicates which routine is called. Context numbers and their meanings are listed in tables per update type in "Context Numbers" later in this chapter.

In the following entries, the state of the Quorum resource is propagated to the cluster nodes; then the Quorum resource is brought online:

```
378.380::1999/06/09-18:00:47.928 [FM] FmpPropagateResourceState:
  resource 254ef0e8-1937-11d3-b3fe-00a0c986aa14 pending event.
378.380::1999/06/09-18:00:47.928 [FM] FmpRmOnlineResource: Resource
  254ef0e8-1937-11d3-b3fe-00a0c986aa14 pending
378.380::1999/06/09-18:00:47.944 [FM] FmpRmOnlineResource: Returning.
  Resource 254ef0e8-1937-11d3-b3fe-00a0c986aa14, state 129, status
  997.
```

Next, the Cluster service makes sure the drive-letter assignment data is consistent across the cluster.

In the following entry, the Cluster service lists the partition table data associated with the drive letter D:

```
15c.470::1999/06/09-18:00:47.944 Physical Disk <Disk D:>: Mountie[0]: 1,
  let=?, start=104000, len=FF9FC000.
```

In the following entries, the Cluster service synchronizes drive-letter information on all the sources (HKEY_LOCAL_MACHINE\SYSTEM\Disk, \DiskInfo, \MountMgr, and \MountVolumeInfo). The information in the HKEY_LOCAL_MACHINE\SYSTEM\Disk subkey is out-of-date, and the fourth and fifth entries report the updating of the subkey.

```
15c.470::1999/06/09-18:00:48.037 Physical Disk <Disk D:>: MountieVerify:
  Registry-System\DISK.GetInfo returned 0 [0:0].
15c.470::1999/06/09-18:00:48.037 Physical Disk <Disk D:>: MountieVerify:
  ClusReg-DiskInfo selected.
15c.470::1999/06/09-18:00:48.037 Physical Disk <Disk D:>: MountieVerify:
  DriveLetters mask is now 00000008.
15c.470::1999/06/09-18:00:48.037 Physical Disk <Disk D:>: MountieVerify:
  Update needed for 08.
15c.470::1999/06/09-18:00:48.053 Physical Disk <Disk D:>: FtInfo_Set:
  Update successful.
```

The following entry means that all the sources of drive letter information have been updated and are synchronized. In the description "Update needed for nn," where "nn" is a two-digit number; "00" means that nothing needs updating. Note that in the fourth entry of the preceding example, the description reads "Update needed for 08."

```
15c.470::1999/06/09-18:00:48.053 Physical Disk <Disk D:>: MountieUpdate:
  Update needed for 00.
```

In the following entry, the Resource Monitor notifies the Cluster service that the Disk D: resource is online.

```
15c.470::1999/06/09-18:00:48.240 [RM] RmpSetResourceStatus, Posting
  state 2 notification for resource <Disk D:>
```

The following entry indicates that the operation of bringing the Quorum resource online has succeeded.

```
378.3a4::1999/06/09-18:00:48.240 [DM] DmpQuoObjNotifyCb: Quorum resource
  is online
```

Applying Quorum Log Changes to the Cluster Database

The quorum log, also known as the *change log,* records changes to the cluster configuration.

At this point in the cluster formation operation, after the Quorum resource is online, the Cluster service does the following:

- Opens the quorum log.
- Validates all the log records for integrity.
- Reads log records and determines which records are newer than the local registry's current copy of the cluster database.
- Applies log changes that are newer to the configuration stored in the registry.

In the following entries, the Cluster service opens the quorum log:

```
378.3a4::1999/06/09-18:00:48.240 [DM] DmpQuoObjNotifyCb: Own quorum
  resource, try open the quorum log
378.3a4::1999/06/09-18:00:48.303 [DM] DmpQuoObjNotifyCb: the name of the
  quorum file is D:\MSCS\quolog.log
378.3a4::1999/06/09-18:00:48.303 [lm] LogCreate : Entry
  FileName=D:\MSCS\quolog.log MaxFileSize=0x00010000
378.3a4::1999/06/09-18:00:48.397 [lm] LogpCreate : Entry
```

The service mounts the log and prepares it so that the log can be read and written:

```
378.3a4::1999/06/09-18:00:48.443 [lm] LogpMountLog : Entry
  pLog=0x000a0650
378.3a4::1999/06/09-18:00:48.443 [lm]:LogpMountLog::Quorumlog File
  size=0x00008000
```

The following entries are typical of those in which the Cluster service checks for or reads log records at a particular offset in the file.

```
378.3a4::1999/06/09-18:00:48.443 [lm]:LogpMountLog::checking LSN
  0x00000408
.
.
.
378.3a4::1999/06/09-18:00:48.443 [lm]:LogpMountLog::checking LSN
  0x00000a48
```

In the cluster form operation, applying quorum log changes to the cluster database involves the following:

1. Checking for a tombstone file.
2. Unrolling the quorum log and propagating state changes to the local registry copy of the cluster database.

Checking for a Tombstone File

A tombstone file is created when a cluster node's quorum-device information is outdated; the tombstone file alerts the node's Cluster service of the fact that the quorum directory is not located where the node's quorum-device information currently says it is.

For example, if node A was not running when node B changed the location of the quorum device, node A has outdated quorum-device information and node B has the current quorum-device information. If node B now goes down and node A comes up, node A's outdated quorum-device information points it to a tombstone file. The tombstone file doesn't point to the new Quorum resource. The result is that node A can only join the cluster; it cannot form a cluster.

In the following entries, the Database Manager checks whether a valid quorum tombstone exists. The return value of zero in the second entry means the DmpChkQuoTombstone function encountered no errors in trying to validate the tombstone information and the form operation can continue.

```
378.380::1999/06/09-18:00:48.459 [DM] DmpChkQuoTombStone - Entry
378.380::1999/06/09-18:00:48.459 [DM] DmpChkQuoTombStone:
  Exit,dwError=0x00000000
```

If DmpChkQuoTombstone had found a valid quorum tombstone, the Database Manager would have logged an error in the event log, and the Cluster service would have exited immediately. In that case, you wouldn't see the Exit statement from DmpChkQuoTombstone in the cluster log.

Unrolling the Quorum Log and Propagating State Changes

In unrolling the quorum log, the Database Manager does the following:

- Finds the last cluster database checkpoint.
- Uploads it to the cluster registry key.
- Applies the changes recorded in the quorum log to the cluster database.

Each configuration change is updated to any cluster nodes. Even though in the cluster form operation there is no other node, each change that occurs in the form operation prompts an update by the GUM.

The following entries are part of the Database Manager's finding the last cluster database checkpoint (sequence number 1656). The first entry notes the sequence number for the cluster key in the local registry.

```
378.380::1999/06/09-18:00:48.459 [DM] DmpApplyChanges: The current
  registry sequence number 1656
378.380::1999/06/09-18:00:48.459 [lm] LogGetLastChkPoint:: Entry
378.328::1999/06/09-18:00:48.537 [FM] HandleResourceTransition: Resource
  Name = 254ef0e8-1937-11d3-b3fe-00a0c986aa14 old state=129 new
  state=2
```

After calling LogGetLastChkPoint, the Log Manager writes the sequence number. Be sure to track such activity by its thread. For example, the following entries are interspersed with entries from the 378.328 thread, which originated with the Failover Manager.

```
378.380::1999/06/09-18:00:48.553 [LM] LogGetLastChkPoint: ChkPt File
  D:\MSCS\chk678.tmp ChkPtSeq=1656 ChkPtLsn=0x00000678 Checksum=93996
378.328::1999/06/09-18:00:48.568 FmpPropagateResourceState: signalling
  the ghQuoOnlineEvent
378.380::1999/06/09-18:00:48.568 [LM] LogGetLastChkPoint exit, returning
  0x00000000
378.328::1999/06/09-18:00:48.568 [GUM] GumSendUpdate:  Locker waiting
  type 0 context 8
378.380::1999/06/09-18:00:48.568 [DM] DmpLogFindStartLsn:
  LogGetLastChkPt rets, Seq#=1656 ChkPtLsn=0x00000678
```

When a checkpoint is written to the quorum disk, the writing of the checkpoint is described in the quorum log by a begin-checkpoint record and an end-checkpoint record. In the following entry, the Database Manager finds the last begin-checkpoint and end-checkpoint records in the quorum log, which it then uses to find and identify the most recent checkpoint file. It then uploads the checkpoint file to the cluster registry key:

```
378.380::1999/06/09-18:00:48.568 [DM] DmpLogFindStartLsn: Uploading
  chkpt from quorum log
```

In the following entries, the Database Manager loads the cluster database and applies the changes in the quorum log to the loaded cluster database:

```
378.380::1999/06/09-18:00:49.365 [DM]: Loading cluster database from
  C:\WINNT\cluster\CLUSDB
15c.46c::1999/06/09-18:00:49.459 Physical Disk:
  AddVolume : \\?\Volume{89fc8c21-18ef-11d3-b1b2-806d6172696f}\
  'C', 7 (8419616)
378.380::1999/06/09-18:00:49.678 [DM] DmpLogFindStartLsn: chkpt uploaded
  from quorum log
378.380::1999/06/09-18:00:49.693 [DM] DmpLogFindStartLsn:
  LSN=0x00000678, dwError=0x00000000
378.380::1999/06/09-18:00:49.693 [DM] DmpApplyChanges: The LSN of the
  record to apply changes from 0x00000678
```

Now that the quorum log entries have been read, validated, and propagated to the cluster's database, the Cluster service is ready to recreate its groups and resources.

Recreating Groups and Resources

In order to start, the Cluster service uses the cluster database from the local registry, which might be out-of-date (for instance, if the cluster database was updated while the node was down).

To guarantee that the Cluster service always uses the most up-to-date copy of the cluster database, the Cluster service updates the cluster configuration by first destroying the local in-memory copies of all the current objects (groups, resources, and so on) and then recreating cluster objects from the updated registry information in the cluster key. The local registry's copy of the cluster database is updated by copying the latest version from the quorum disk or, if the node is joining a cluster, from the sponsoring node.

Destroying Groups and Resources

The following entries report the destruction of the following:

- Disk Group 1 (254ef0e7-1937-11d3-b3fe-00a0c986aa14).
- The Quorum resource (254ef0e8-1937-11d3-b3fe-00a0c986aa14).

```
378.380::1999/06/09-18:00:49.693 [FM] FmFormNewClusterPhase1, Entry.
  Quorum quorum will be deleted
378.380::1999/06/09-18:00:49.693 [FM] DestroyGroup: destroying
  254ef0e7-1937-11d3-b3fe-00a0c986aa14
378.380::1999/06/09-18:00:49.693 [FM] DestroyResource: destroying
  254ef0e8-1937-11d3-b3fe-00a0c986aa14
15c.46c::1999/06/09-18:00:49.693 Physical Disk: AddVolume:
  GetPartitionInfo(\??\Volume{89fc8c1b-18ef-11d3-b1b2-806d6172696f}),
  error 1
378.380::1999/06/09-18:00:49.693 [FM] FmpDestroyResource Exit.
378.380::1999/06/09-18:00:49.693 [FM] FmpDestroyGroup: Group
  254ef0e7-1937-11d3-b3fe-00a0c986aa14 destroyed.
378.380::1999/06/09-18:00:49.693 [OM] Deleting object Disk Group 1
  (254ef0e7-1937-11d3-b3fe-00a0c986aa14)
```

Creating Groups and Resources

The forming cluster builds up the memory database by recreating first the groups and then the resources on the basis of the new configuration.

Configuring the networks (which is covered in the next section) happens after the Cluster service has destroyed the groups and resources and before it recreates them.

Note that this set of operations actually follows the cluster's configuring of the network in time.

Recreating the groups and resources begins with the following:

```
378.380::1999/06/09-18:00:51.162 [FM] FmFormNewClusterPhase2, Entry.
```

In the subsequent entries, up to "378.380::1999/06/09-18:00:51.209 [FM] All groups created", the forming node does the following:

- Reads the list of groups from the registry:

  ```
  378.380::1999/06/09-18:00:51.193 [FM] Processing groups list.
  ```

- Creates and initializes the first group:

  ```
  378.380::1999/06/09-18:00:51.193 [FM] Creating group 254ef0e4-1937-
    11d3-b3fe-00a0c986aa14
  ```

- Initializes the group from the registry:

```
378.380::1999/06/09-18:00:51.193 [FM] Initializing group 254ef0e4-
  1937-11d3-b3fe-00a0c986aa14 from the registry.
378.380::1999/06/09-18:00:51.193 [FM] Name for Group 254ef0e4-1937-
  11d3-b3fe-00a0c986aa14 is 'Cluster Group'.
```

- Creates and initializes resources:

```
378.380::1999/06/09-18:00:51.193 [FM] Group
  254ef0e4-1937-11d3-b3fe-00a0c986aa14 contains Resource
  254ef0e5-1937-11d3-b3fe-00a0c986aa14.
378.380::1999/06/09-18:00:51.193 [FM] Creating resource
  254ef0e5-1937-11d3-b3fe-00a0c986aa14
378.380::1999/06/09-18:00:51.193 [FM] Initializing resource
  254ef0e5-1937-11d3-b3fe-00a0c986aa14 from the registry.
378.380::1999/06/09-18:00:51.193 [FM] Name for Resource
  254ef0e5-1937-11d3-b3fe-00a0c986aa14 is 'Cluster IP Address'.
.
.
.
378.380::1999/06/09-18:00:51.193 [FM] Group
  254ef0e4-1937-11d3-b3fe-00a0c986aa14 contains Resource
  254ef0e6-1937-11d3-b3fe-00a0c986aa14.
378.380::1999/06/09-18:00:51.193 [FM] Creating resource
  254ef0e6-1937-11d3-b3fe-00a0c986aa14
378.380::1999/06/09-18:00:51.193 [FM] Initializing resource
  254ef0e6-1937-11d3-b3fe-00a0c986aa14 from the registry.
378.380::1999/06/09-18:00:51.193 [FM] Name for Resource
  254ef0e6-1937-11d3-b3fe-00a0c986aa14 is 'Cluster Name'.
```

- Adds possible owners to the resources and groups:

```
378.380::1999/06/09-18:00:51.193 [FM] FmpAddPossibleEntry: adding
  node 1 as possible host for resource
  254ef0e5-1937-11d3-b3fe-00a0c986aa14.
378.380::1999/06/09-18:00:51.193 [FM] FmpAddPossibleEntry: adding
  node 2 as possible host for resource
  254ef0e5-1937-11d3-b3fe-00a0c986aa14.
378.380::1999/06/09-18:00:51.193 [FM] All dependencies for resource
  254ef0e5-1937-11d3-b3fe-00a0c986aa14 created.
.
.
.
378.380::1999/06/09-18:00:51.193 [FM] FmpAddPossibleEntry: adding
  node 1 as possible host for resource
  254ef0e6-1937-11d3-b3fe-00a0c986aa14.
378.380::1999/06/09-18:00:51.193 [FM] FmpAddPossibleEntry: adding
  node 2 as possible host for resource
  254ef0e6-1937-11d3-b3fe-00a0c986aa14.
```

- The Cluster service discovers that resource 254ef0e6-1937-11d3-b3fe-00a0c986aa14 (the Cluster Name resource) depends on resource 254ef0e5-1937-11d3-b3fe-00a0c986aa14 (the Cluster IP Address resource). The Cluster service tries to create the Cluster IP Address resource, discovers that it already exists, and so reports the final entry in this group:

```
378.380::1999/06/09-18:00:51.193 [FM] Resource
  254ef0e6-1937-11d3-b3fe-00a0c986aa14 depends on
  254ef0e5-1937-11d3-b3fe-00a0c986aa14. Creating...
378.380::1999/06/09-18:00:51.193 [FM] CreateResource, Opened existing
  resource 254ef0e5-1937-11d3-b3fe-00a0c986aa14
378.380::1999/06/09-18:00:51.193 [FM] All dependencies for resource
  254ef0e6-1937-11d3-b3fe-00a0c986aa14 created.
```

Among the groups and resources created are the Quorum resource and the group that contains the Quorum resource:

```
378.380::1999/06/09-18:00:51.209 [FM] Name for Group
  254ef0e7-1937-11d3-b3fe-00a0c986aa14 is 'Disk Group 1'.
378.380::1999/06/09-18:00:51.209 [FM] Group
  254ef0e7-1937-11d3-b3fe-00a0c986aa14 contains Resource
  254ef0e8-1937-11d3-b3fe-00a0c986aa14.
378.380::1999/06/09-18:00:51.209 [FM] CreateResource, Opened existing
  resource 254ef0e8-1937-11d3-b3fe-00a0c986aa14
378.380::1999/06/09-18:00:51.209 [FM] ReCreating quorum resource
  254ef0e8-1937-11d3-b3fe-00a0c986aa14
```

After the service has created all the groups and resources, this entry appears in the log:

```
378.380::1999/06/09-18:00:51.209 [FM] All groups created.
```

With the group and resource objects recreated and initialized and their possible owners added, the Cluster service is ready to configure the cluster's network information.

Configuring the Networks

In this phase of cluster formation, the Cluster service creates network and network interface objects and makes sure the information about them is accurate and consistent across the cluster. To do so, the network configuration engine does the following:

- Creates objects of the node, network, and network interface information in the registry (that is, the cluster database, as modified by the quorum log).
- Collects the system's current network configuration.
- Recreates the networks and interface objects.
- Registers the networks and interfaces with cluster transport.
- Brings the networks and interfaces online with the Cluster service and transport.
- Performs a connectivity report to detect network interface failures.
- Calculates NodeHighestVersion and NodeLowestVersion.
- Performs fixups.
- Notes the state changes of interfaces as they come online.

The following entry begins network-configuration:

```
378.380::1999/06/09-18:00:49.756 [NM] Synchronizing node information.
```

Creating Objects

The Cluster service creates node objects that represent the members of the cluster, according to the cluster database.

Note that this cluster is being formed with only one node.

```
378.380::1999/06/09-18:00:49.756 [NM] Creating node objects.
378.380::1999/06/09-18:00:49.756 [NM] Creating object for node 2 (NODE2)
```

Collecting System Information

The Cluster service collects information from the system about network adapters and network connection objects.

Interfaces are IP addresses. An adapter can have multiple IP addresses (interfaces) associated with it.

```
378.380::1999/06/09-18:00:49.772 [NM] Synchronizing network information.
378.380::1999/06/09-18:00:49.787 [NM] Synchronizing interface
  information.
```

In the following entry, the Cluster service compares the current configuration with the one recorded in the cluster database. It notes which network adapters have been removed or added or whether there have been no changes in the network configuration.

```
378.380::1999/06/09-18:00:49.818 [NM] Running network configuration
  engine.
```

In the following entries, the Node Manager reports any changes it detected. It updates the cluster database with this information. The second entry means the previously existing adapters (one, in this case) were matched to their previous networks.

```
378.380::1999/06/09-18:00:50.600 [NM] Processing network configuration
  changes.
378.380::1999/06/09-18:00:50.600 [NM] Matched 1 networks, created 0 new
  networks.
378.380::1999/06/09-18:00:50.600 [NM] Resynchronizing network
  information.
378.380::1999/06/09-18:00:50.600 [NM] Resynchronizing interface
  information.
```

Recreating Network and Interface Objects

The Cluster service creates the following network and interface objects on the basis of the current network configuration:

```
378.380::1999/06/09-18:00:50.600 [NM] Creating network objects.
378.380::1999/06/09-18:00:50.600 [NM] Creating object for network
  5da3c9ea-7c25-4129-ab70-2c058de6d1fd (CPQ Netintellegent 10.10.0.2).
378.380::1999/06/09-18:00:50.600 [NM] Creating object for network
  d561dbcb-1936-11d3-b3fe-00a0c986aa14 ( 100b 172.16.64.1).
378.380::1999/06/09-18:00:50.600 [NM] Creating interface objects.
378.380::1999/06/09-18:00:50.600 [NM] Creating object for interface
  2e12db05-2947-425f-835a-fb9e4f13f0b3 ( 100b 172.16.64.1 -     NODE2).
```

In the following entries, the Node Manager gathers information used to detect network interface failures:

```
378.380::1999/06/09-18:00:50.600 [NM] Assigned index 0 to interface
  2e12db05-2947-425f-835a-fb9e4f13f0b3.
378.380::1999/06/09-18:00:50.600 [NM] Creating object for interface
  a6fe4897-139a-4b84-815f-8892f674afb8 (CPQ Netintellegent 10.10.0.2 -
  NODE2).
378.380::1999/06/09-18:00:50.600 [NM] Assigned index 0 to interface
  a6fe4897-139a-4b84-815f-8892f674afb8.
378.380::1999/06/09-18:00:50.600 [NM] Creating object for interface
  d561dbcc-1936-11d3-b3fe-00a0c986aa14 ( 100b 172.16.64.1 -     NODE1).
378.380::1999/06/09-18:00:50.600 [NM] Assigned index 1 to interface
  d561dbcc-1936-11d3-b3fe-00a0c986aa14.
```

Registering Networks and Interfaces and Bringing Them Online

In the entries in this section, the Node Manager does the following:

- Registers the network d561dbcb-1936-11d3-b3fe-00a0c986aa14 and its network interfaces with cluster transport.

  The cluster transport is Clusnet.sys, a transport driver that uses the User Datagram Protocol (UDP) and that the Cluster service uses to send messages to other cluster members.

  A network is a link between network interfaces attached to the same subnet. If both network interfaces are online, the network state is up.
- Creates objects for the interfaces.
- Brings the network online.

Note the event entry description, "Registering interface 2e12db05-2947-425f-835a-fb9e4f13f0b3 with cluster transport, addr 172.16.64.2, endpoint 3343," in which

- "endpoint 3343" is the UDP socket number for clusnet's (the cluster transport's) socket, and
- "addr 172.16.64.1" is the relevant IP Address.

Thus, "Registering interface 2e12db05-2947-425f-835a-fb9e4f13f0b3 with cluster transport, addr 172.16.64.2, endpoint 3343" means the following:

- The node's network interface is registered.
- Socket 3343 on the node's IP address has been opened.

```
378.380::1999/06/09-18:00:50.600 [NM] Registering network
  d561dbcb-1936-11d3-b3fe-00a0c986aa14 with cluster transport.
378.380::1999/06/09-18:00:50.600 [NM] Bringing network
  d561dbcb-1936-11d3-b3fe-00a0c986aa14 online.
378.380::1999/06/09-18:00:50.600 [NM] Registering interfaces for network
  d561dbcb-1936-11d3-b3fe-00a0c986aa14 with cluster transport.
378.380::1999/06/09-18:00:50.600 [NM] Registering interface
  2e12db05-2947-425f-835a-fb9e4f13f0b3 with cluster transport, addr
  172.16.64.2, endpoint 3343.
378.380::1999/06/09-18:00:50.600 [NM] Registering interface
  d561dbcc-1936-11d3-b3fe-00a0c986aa14 with cluster transport, addr
  172.16.64.1, endpoint 3343.
378.380::1999/06/09-18:00:50.615 [NM] Initializing membership...
```

Performing the Connectivity Report

The following entries signal that the Node Manager is starting a connectivity report to detect any network interface failures.

```
378.328::1999/06/09-18:00:50.631 [NM] Scheduling connectivity report for
  network d561dbcb-1936-11d3-b3fe-00a0c986aa14 ( 100b     172.16.64.1)
378.328::1999/06/09-18:00:50.631 [NM] Updating local connectivity info
  for network d561dbcb-1936-11d3-b3fe-00a0c986aa14.
378.328::1999/06/09-18:00:50.631 [NM] Node for interface 0 is down
378.328::1999/06/09-18:00:50.631 [NM] Interface 1 is either up or
  unreachable
```

Note that in the following entry, the Node Manager discounts the connectivity report from interface 0 because the interface's node is down.

```
378.328::1999/06/09-18:00:50.631 [NM] The report from interface 0 is not
  valid
378.328::1999/06/09-18:00:50.662 [NM] Interface 1 is Up
378.328::1999/06/09-18:00:50.662 [NM] Beginning phase 2 of state
  computation for network d561dbcb-1936-11d3-b3fe-00a0c986aa14
```

In the following entry, the Node Manager summarizes the connectivity report. The unavailable interface is interface 0. (See the preceding entries.)

```
378.328::1999/06/09-18:00:50.662 [NM] Unavailable=1, Failed = 0,
  Unreachable=0, Reachable=1, Up=1,
```

For the meaning of state 3, see "State Codes" later in this chapter.

```
378.328::1999/06/09-18:00:50.662 [NM] Network d561dbcb-1936-11d3-b3fe-
  00a0c986aa14 is now in state 3
378.328::1999/06/09-18:00:50.662 [NM] Interface d561dbcc-1936-11d3-b3fe-
  00a0c986aa14 is up (node: NODE1, network:  100b 172.16.64.1).
378.328::1999/06/09-18:00:50.662 [NM] Network d561dbcb-1936-11d3-b3fe-
  00a0c986aa14 ( 100b 172.16.64.1) is up.
378.380::1999/06/09-18:00:50.631 [CLSMSG] Initializing.
378.380::1999/06/09-18:00:50.881 [NM] Membership initialization
  complete.
```

NodeHighestVersion and NodeLowestVersion

If the cluster nodes are running different versions of Windows NT or Windows 2000 Advanced Server and therefore different versions of the Cluster service, there can be an issue of whether the versions are compatible with all the cluster nodes. The Cluster service computes the Cluster service version associated with each node. It then computes the cluster version for the cluster as a whole from the versions of all the active members of the cluster. As other members join the cluster, the existing members compute a new version of the cluster.

In the following two pairs of entries, Cluster service computes a cluster version that is based on the versions of all the active members of the cluster, even when there is only one active member. The version number falls into two 16-bit numbers:

- The top 16 bits represent the cluster major version.
- The lower 16 bits represent the cluster build number.

First the Cluster service retrieves the NodeHighestVersion and NodeLowestVersion properties for each active member.

- NodeHighestVersion reflects the version of the Cluster service that is running on the member.
- NodeLowestVersion reflects the lowest version of Cluster service with which this active member can communicate.

```
378.380::1999/06/09-18:00:50.881 [NM] NmpValidateNodeVersion: Node=1,
  HighestVersion=0x00030803, LowestVersion=0x000200e0
```

In the second pair of entries, the Cluster service calculates the highest version of the Cluster service that can reliably run on all the nodes in the cluster and the lowest version with which every node can communicate. The calculation is biased toward the middle:

- ClusterHighestVersion is the minimum of all NodeHighestVersion properties. This indicates which versions of Cluster service are running on all active members if the cluster is a mixed-mode cluster.
- ClusterLowestVersion is the maximum of all NodeLowestVersion properties; this indicates which version of Cluster service must be running on a server for it to join this cluster.

```
378.380::1999/06/09-18:00:50.881 [NM] NmpCalcClusterVersion: status = 0
  ClusHighestVer=0x00030803, ClusLowestVer=0x000200e0
378.380::1999/06/09-18:00:50.881 [NM] [NmpResetClusterVersion]
  ClusterHighestVer=0x00030803 ClusterLowestVer=0x000200e0
378.380::1999/06/09-18:00:50.881 [NM] Disabling mixed NT4/NT5 operation.
```

Cluster version 1 corresponds to the Windows NT 4.0 Service Pack 3 release, and version 2 to the Windows NT 4.0 Service Pack 4 and Windows NT 4.0 Service Pack 5 releases.

Performing Fixups

In the following entry, the Node Manager performs fixups, which enables the cluster to run in mixed mode. (In this case, node 1 is running Windows 2000, and node 2 is running Windows NT 4.0.) Fixups are also run the first time a node that has been upgraded runs the Cluster service. After running fixups, the Cluster service performs a global update of the membership.

```
378.380::1999/06/09-18:00:50.881 [NM] NmPerformFixups Entry,
  dwFixupType=1
```

Updating the Local Cluster Database and Setting Database Entries

The following entries are typical of those that report beginning and committing local updates. Note the timestamps. Local updates of the cluster database can, but do not necessarily have to, occur at this point in cluster formation.

```
378.380::1999/06/09-18:00:50.897 [GUM] GumSendUpdate:  Locker waiting
  type 2 context 17
378.380::1999/06/09-18:00:50.897 [GUM] Thread 0x380 UpdateLock wait on
  Type 2
378.380::1999/06/09-18:00:50.897 [GUM] DoLockingUpdate successful, lock
  granted to 1
378.380::1999/06/09-18:00:50.897 [GUM] GumSendUpdate: Locker dispatching
  seq 1657 type 2 context 17
378.380::1999/06/09-18:00:50.897 [dm] DmBeginLocalUpdate Entry
378.380::1999/06/09-18:00:50.897 [dm] DmBeginLocalUpdate Exit,
  pLocalXsaction=0x000a0dd8 dwError=0x00000000
```

Bringing Resources Online

In the last stage of forming a cluster, the Cluster service creates resources and tries to bring them online. There are two parts to bringing resources online: in the cluster database, adding possible owners for resource types in the cluster database; and then creating the resources and bringing them online.

Adding Possible Owners for Resource Types

In the following entries, the Failover Manager adds possible owners for resource types.

In the log, there are many more events of this type, but they are all essentially identical to the two reproduced here:

```
378.380::1999/06/09-18:00:50.897 [FM] processing resource types.
378.380::1999/06/09-18:00:50.897 [FM] FmpAddPossibleNode: adding node 1
  to resource type's possible node list
378.380::1999/06/09-18:00:50.897 [FM] FmpAddPossibleNode: adding node 2
  to resource type's possible node list
```

After adding possible owners for resource types, the Cluster service updates local copies of the cluster registry key:

```
378.380::1999/06/09-18:00:50.928 [NM] NmpUpdatePerformJoinFixups2:
  called postfixup notifycb function with  status 0
378.380::1999/06/09-18:00:50.928 [DM] DmCommitLocalUpdate Entry
378.380::1999/06/09-18:00:50.928 [DM] DmCommitLocalUpdate Exit,
  dwError=0x00000000
378.380::1999/06/09-18:00:50.928 [GUM] GumpDoUnlockingUpdate releasing
  lock ownership
378.380::1999/06/09-18:00:50.928 [GUM] GumSendUpdate: completed update
  seq 1657    type 2 context 17
```

Creating Resources and Bringing Them Online

The following entries are representative of the process of creating and initializing resources. The pattern repeats, of course, for each resource. In this section, the Cluster service creates all groups from a group list in the cluster database, as indicated by the following line.

```
378.380::1999/06/09-18:00:51.193 [FM] Processing groups list.
```

In the following example, the Failover Manager creates and initializes the Cluster Group:

```
378.380::1999/06/09-18:00:51.193 [FM] Creating group 254ef0e4-1937-11d3-
  b3fe-00a0c986aa14
378.380::1999/06/09-18:00:51.193 [FM] Initializing group 254ef0e4-1937-
  11d3-b3fe-00a0c986aa14 from the registry.
378.380::1999/06/09-18:00:51.193 [FM] Name for Group 254ef0e4-1937-11d3-
  b3fe-00a0c986aa14 is 'Cluster Group'.
```

In the following line, the Failover Manager creates and brings online the first of the resources that the Cluster Group contains. The Failover Manager repeats this pattern for all the resources the Cluster Group contains.

The pattern for creating groups and resources is that the Failover Manager does the following:

- Notes that the containing group in fact contains the resource:

```
378.380::1999/06/09-18:00:51.193 [FM] Group 254ef0e4-1937-11d3-b3fe-
00a0c986aa14 contains Resource 254ef0e5-1937-11d3-b3fe-
00a0c986aa14.
```

- Creates the resource and initializes it from the cluster registry key:

```
378.380::1999/06/09-18:00:51.193 [FM] Creating resource 254ef0e5-
1937-11d3-b3fe-00a0c986aa14
378.380::1999/06/09-18:00:51.193 [FM] Initializing resource 254ef0e5-
1937-11d3-b3fe-00a0c986aa14 from the registry.
```

- Reports the name of the resource to the cluster log:

```
378.380::1999/06/09-18:00:51.193 [FM] Name for Resource 254ef0e5-
1937-11d3-b3fe-00a0c986aa14 is 'Cluster IP Address'.
```

- Adds possible owners of the resource in the cluster registry key:

```
378.380::1999/06/09-18:00:51.193 [FM] FmpAddPossibleEntry: adding
node 1 as possible host for resource 254ef0e5-1937-11d3-b3fe-
00a0c986aa14.
378.380::1999/06/09-18:00:51.193 [FM] FmpAddPossibleEntry: adding
node 2 as possible host for resource 254ef0e5-1937-11d3-b3fe-
00a0c986aa14.
```

- Creates and opens any dependencies for the resource:

```
378.380::1999/06/09-18:00:51.193 [FM] All dependencies for resource
254ef0e5-1937-11d3-b3fe-00a0c986aa14 created.
```

The following entry starts the creation of the next resource, which is the Cluster Name resource:

```
378.380::1999/06/09-18:00:51.193 [FM] Group 254ef0e4-1937-11d3-b3fe-
  00a0c986aa14 contains Resource 254ef0e6-1937-11d3-b3fe-00a0c986aa14.
```

In creating and bringing online the Cluster Name resource (Resource 254ef0e6-1937-11d3-b3fe-00a0c986aa14), the Failover Manager discovers that the Cluster Name resource depends on the Cluster IP Address resource and so opens the Cluster IP Address resource. Note that, in order for the Cluster service to perform such actions as bringing a resource online and changing its properties, it has to first open the resource to obtain a handle to it.

```
378.380::1999/06/09-18:00:51.193 [FM] Resource 254ef0e6-1937-11d3-b3fe-
  00a0c986aa14 depends on 254ef0e5-1937-11d3-b3fe-00a0c986aa14.
  Creating...
378.380::1999/06/09-18:00:51.193 [FM] CreateResource, Opened existing
  resource 254ef0e5-1937-11d3-b3fe-00a0c986aa14
378.380::1999/06/09-18:00:51.193 [FM] All dependencies for resource
  254ef0e6-1937-11d3-b3fe-00a0c986aa14 created.
```

The remainder of the events involved in bringing resource 254ef0e6-1937-11d3-b3fe-00a0c986aa14 (the Cluster Name resource) online are interspersed with entries reporting a great many other events, most of which, including global updates, involve creating and attempting to bring online other resources and groups.

In the following entries, the Cluster service starts creating the Quorum resource and the group to which it belongs. Notice that the Cluster service does not bring the Quorum resource online before the Cluster Name resource and that the pattern of events is the same for all resources and groups:

```
378.380::1999/06/09-18:00:51.193 [FM] Creating group 254ef0e7-1937-11d3-
  b3fe-00a0c986aa14
378.380::1999/06/09-18:00:51.209 [FM] Initializing group 254ef0e7-1937-
  11d3-b3fe-00a0c986aa14 from the registry.
378.380::1999/06/09-18:00:51.209 [FM] Name for Group 254ef0e7-1937-11d3-
  b3fe-00a0c986aa14 is 'Disk Group 1'.
378.380::1999/06/09-18:00:51.209 [FM] Group 254ef0e7-1937-11d3-b3fe-
  00a0c986aa14 contains Resource 254ef0e8-1937-11d3-b3fe-00a0c986aa14.
378.380::1999/06/09-18:00:51.209 [FM] CreateResource, Opened existing
  resource 254ef0e8-1937-11d3-b3fe-00a0c986aa14
378.380::1999/06/09-18:00:51.209 [FM] ReCreating quorum resource
  254ef0e8-1937-11d3-b3fe-00a0c986aa14
```

In a mixed-mode cluster, different versions of resource type DLLs can require fixups just as the Cluster service does. After all groups have been created and if the cluster is running in mixed-mode, fixups can be run on the resource type DLLs. This is described in the following entries.

```
378.380::1999/06/09-18:00:51.209 [FM] FmpFixupResourceTypes Entry.
378.380::1999/06/09-18:00:51.209 [FM] FmpFixupResourceTypes Exit
```

Then the Cluster service obtains and reports the state of all the groups in the cluster.

```
378.380::1999/06/09-18:00:51.209 [FMX] GetGroupListState, Group <Disk
  Group 1> state = 0
378.380::1999/06/09-18:00:51.209 [FMX] GetGroupListState, Group <Cluster
  Group> state = 1
378.380::1999/06/09-18:00:51.209 [FMX] GetGroupListState, Group <test
  group> state = 1
```

In the following entries, the Failover Manager initializes the Cluster IP Address resource by using the registry and then performs the subsequent global update and writes change records to the quorum log:

```
378.380::1999/06/09-18:00:51.209 [FM] Initializing resource 254ef0e5-
  1937-11d3-b3fe-00a0c986aa14 from the registry.
378.380::1999/06/09-18:00:51.225 [FM] Name for Resource 254ef0e5-1937-
  11d3-b3fe-00a0c986aa14 is 'Cluster IP Address'.
378.380::1999/06/09-18:00:51.225 [FM] FmpRmCreateResource: creating
  resource 254ef0e5-1937-11d3-b3fe-00a0c986aa14 in shared resource
  monitor
378.3c0::1999/06/09-18:00:51.240 [GUM] GumSendUpdate:  Locker waiting
  type 1 context 0
378.3c0::1999/06/09-18:00:51.240 [GUM] Thread 0x3c0 UpdateLock wait on
  Type 1
378.3c0::1999/06/09-18:00:51.240 [GUM] DoLockingUpdate successful, lock
  granted to 1
```

The following GUM update concerns creating a new registry key or entry:

```
378.3c0::1999/06/09-18:00:51.240 [GUM] GumSendUpdate: Locker dispatching
  seq 1659    type 1 context 0
378.3c0::1999/06/09-18:00:51.240 [dm] DmWriteToQuorumLog Entry Seq#=1659
  Type=0 Size=152
378.3c0::1999/06/09-18:00:51.240 [dm] DmWriteToQuorumLog Entry Seq#=1659
  Type=0 Size=152
378.3c0::1999/06/09-18:00:51.240 [GUM] GumpDoUnlockingUpdate releasing
  lock ownership
378.3c0::1999/06/09-18:00:51.240 [GUM] GumSendUpdate: completed update
  seq 1659    type 1 context 0
```

The following entry reports that the network interface is no longer valid; the interface is no longer available.

```
15c.458::1999/06/09-18:00:51.256 IP Address <Cluster IP Address>: NBT
  interface \Device\NetBt_If1 (instance 0x712F4D0A) is no longer
  valid.
```

In the following entry, the update concerns deleting an entry from a registry key, probably the one for the interface that is no longer valid:

```
378.3c0::1999/06/09-18:00:51.256 [GUM] GumSendUpdate:  Locker waiting
  type 1 context 4099
378.3c0::1999/06/09-18:00:51.256 [GUM] Thread 0x3c0 UpdateLock wait on
  Type 1
378.3c0::1999/06/09-18:00:51.256 [GUM] DoLockingUpdate successful, lock
  granted to 1
378.3c0::1999/06/09-18:00:51.256 [GUM] GumSendUpdate: Locker dispatching
  seq 1660    type 1 context 4099
```

The Cluster service attempts again to create and bring the Cluster Name resource (254ef0e6-1937-11d3-b3fe-00a0c986aa14) online, initializing the resource from the cluster registry key and creating it in the Resource Monitor:

```
378.380::1999/06/09-18:00:51.256 [FM] Initializing resource 254ef0e6-
  1937-11d3-b3fe-00a0c986aa14 from the registry.
378.380::1999/06/09-18:00:51.256 [FM] Name for Resource 254ef0e6-1937-
  11d3-b3fe-00a0c986aa14 is 'Cluster Name'.
378.380::1999/06/09-18:00:51.271 [FM] FmpRmCreateResource: creating
  resource 254ef0e6-1937-11d3-b3fe-00a0c986aa14 in shared resource
  monitor
```

In the following entries, the Database Manager has created a registry key to replace the one that was deleted in preceding the entries for 1999/06/09-18:00:51.256. This key might be a registry entry or subkey, but its creation is reported to the cluster log as a "key." The GUM performs a global update with this change, and the Database Manager writes the change to the quorum log. Note, again, that the type (0) in the write operation to the quorum log matches the context number for the global update. If the Cluster service on this node were to crash right after this operation and the other node were started, the other node would have access to the changes recorded in the quorum log and would apply them in its cluster formation process. As a result, the current configuration of the cluster would be preserved.

```
378.3c0::1999/06/09-18:00:51.271 [GUM] GumSendUpdate:  Locker waiting
  type 1 context 0
378.3c0::1999/06/09-18:00:51.271 [GUM] Thread 0x3c0 UpdateLock wait on
  Type 1
378.3c0::1999/06/09-18:00:51.271 [GUM] DoLockingUpdate successful, lock
  granted to 1
378.3c0::1999/06/09-18:00:51.271 [GUM] GumSendUpdate: Locker dispatching
  seq 1662    type 1 context 0
378.3c0::1999/06/09-18:00:51.271 [dm] DmWriteToQuorumLog Entry Seq#=1662
  Type=0 Size=152
378.3c0::1999/06/09-18:00:51.271 [dm] DmWriteToQuorumLog Entry Seq#=1662
  Type=0 Size=152
```

```
378.3c0::1999/06/09-18:00:51.271 [GUM] GumpDoUnlockingUpdate releasing
  lock ownership
378.3c0::1999/06/09-18:00:51.271 [GUM] GumSendUpdate: completed update
  seq 1662    type 1 context 0
15c.3cc::1999/06/09-18:00:51.287 Network Name <Cluster Name>: Successful
  open of resid 507752
378.380::1999/06/09-18:00:51.287 [FM] FmpRmCreateResource: created
  resource 254ef0e6-1937-11d3-b3fe-00a0c986aa14, resid 741136
```

After bringing other resources online, the Cluster service tries to bring the Cluster group, which contains the Cluster Name and Cluster IP Address resources, online:

```
378.380::1999/06/09-18:00:51.318 [FM] FmpOnlineGroupFromList: Previous
  group state for 254ef0e4-1937-11d3-b3fe-00a0c986aa14 is 1
378.380::1999/06/09-18:00:51.318 [FM] FmpOnlineGroup: trying to bring
  group 254ef0e4-1937-11d3-b3fe-00a0c986aa14 online
378.380::1999/06/09-18:00:51.318 [FM] OnlineGroup for 254ef0e4-1937-
  11d3-b3fe-00a0c986aa14 owner 651932
```

Next, the Cluster service brings the Cluster IP Address online. Note that the Failover Manager acquires a shared lock on the Quorum resource to do so. (For more information about shared locks, see "Shared Locks and gdwQuoBlockingResources" earlier in this chapter.)

```
378.380::1999/06/09-18:00:51.318 [FM] FmpRmOnlineResource: bringing
  resource 254ef0e5-1937-11d3-b3fe-00a0c986aa14 (resid 735200) online.
378.380::1999/06/09-18:00:51.318 [CP] CppResourceNotify for resource
  Cluster IP Address
378.380::1999/06/09-18:00:51.334 [FM] FmpRmOnlineResource: called
  InterlockedIncrement on gdwQuoBlockingResources for resource
  254ef0e5-1937-11d3-b3fe-00a0c986aa14
15c.458::1999/06/09-18:00:51.334 IP Address <Cluster IP Address>:
  Bringing resource online...
```

The GUM initiates a global update for the Cluster IP Address resource's change in state from Pending to Online while the Online thread is still running:

```
378.380::1999/06/09-18:00:51.334 [GUM] GumSendUpdate:  Locker waiting
  type 0 context 8
15c.488::1999/06/09-18:00:51.334 IP Address <Cluster IP Address>: Online
thread running.
```

The following entry is part of a group of entries that describe activity while both the Cluster IP Address and Cluster Name resources are still in the Pending state. In this entry, the IP Address resource is notified that the network interface to which it is bound has changed or been deleted.

```
15c.488::1999/06/09-18:00:51.334 IP Address <Cluster IP Address>:
  Created cluster notify port.
```

With the Cluster Group online, the Cluster service checks to make sure that there is still a network interface to which the IP Address can be bound on the same subnet. If there is not, the IP Address is orphaned.

```
15c.488::1999/06/09-18:00:51.350 IP Address <Cluster IP Address>:
  Checking if resource is orphaned: network masks 00FCFFFF=00FFFFFF
  and addresses FE4010AC^00000A0A.
```

After working on bringing another resource online, the Cluster service continues bringing the Cluster IP Address resource online. In these two entries, the Cluster service obtains a handle for the network interface and registers it so that the IP Address resource that is bound to it is notified if a change occurs to the network interface.

```
15c.488::1999/06/09-18:00:51.365 IP Address <Cluster IP Address>:
  Online: Opened object handle for netinterface d561dbcc-1936-11d3-
  b3fe-00a0c986aa14.
15c.488::1999/06/09-18:00:51.365 IP Address <Cluster IP Address>:
  Online: Registered notification for netinterface d561dbcc-1936-11d3-
  b3fe-00a0c986aa14.
```

In the following entries, the Database Manager takes a checkpoint of the cluster database (in this case, the cluster key of the registry for the local system), and the Log Manager writes it to the quorum disk:

```
378.380::1999/06/09-18:00:51.412 [dm]DmUpdateFormNewCluster - taking a
  checkpoint
378.380::1999/06/09-18:00:51.412 [LM] LogCheckPoint entry
378.380::1999/06/09-18:00:51.412 [LM] DmpGetSnapshotCb::checkpoint file
  name =D:\MSCS\chk67D.tmp Seq#=1661
378.380::1999/06/09-18:00:51.756 [dm] DmpGetSnapShotCb - DmpGetDatabase
  returned 0x00000000
378.380::1999/06/09-18:00:51.756 [LM] LogCheckPoint:
  ChkPtFile=D:\MSCS\chk67D.tmp Chkpt Trid=1661 CheckSum=86865
378.380::1999/06/09-18:00:51.756 [lm] LogpAppendPage : Writing 1024
  bytes to disk at offset 0x00000800
378.380::1999/06/09-18:00:51.756 [lm] LogFlush : pLog=0x000a0650 writing
  the 1024 bytes for active page at offset 0x00000c00
```

```
378.380::1999/06/09-18:00:51.771 [LM] LogCheckPoint: EndChkpt written.
  EndChkPtLsn =0x00000c08 ChkPt Seq=1661 ChkPt
  FileName=D:\MSCS\chk67D.tmp
378.380::1999/06/09-18:00:51.771 [lm] LogpCheckpoint : Writing 1024
  bytes to disk at offset 0x00000000
378.380::1999/06/09-18:00:51.771 [LM] LogCheckPoint Exit
```

The following entry reports that a node is down. Because a node is down, updates are made to the database and written to the quorum log — therefore, the Database Manager performs the update. (If all nodes were up, the GUM would perform the update, replicating any change across the cluster.)

```
378.380::1999/06/09-18:00:51.771 [dm] DmUpdateFormNewCluster - some node
  down
```

And now the cluster has successfully formed:

```
378.380::1999/06/09-18:00:51.787 [INIT] Successfully formed a cluster.
378.380::1999/06/09-18:00:51.803 [INIT] Cluster started.
```

With the cluster formed, the Cluster service continues bringing resources online, including the Cluster IP Address resource.

After making a number of changes to the cluster database and subsequently writing change records to the quorum log, the Database Manager changes one of the Parameters subkeys for the Cluster IP Address resource and writes the change to the quorum log. Then the GUM initiates a global update with the change to the registry (type 1 context 4098):

```
378.498::1999/06/09-18:00:53.162 DM: Setting value NbtDeviceInstance for
  key Resources\254ef0e5-1937-11d3-b3fe-00a0c986aa14\Parameters\1
378.498::1999/06/09-18:00:53.178 [dm] DmWriteToQuorumLog Entry Seq#=1677
  Type=4098 Size=180
378.498::1999/06/09-18:00:53.178 [lm] LogWrite : Entry TrId=1677 RmId=5
  RmType = 4098 Size=180
378.498::1999/06/09-18:00:53.178 [lm] LogWrite : Exit
  returning=0x000011f0
378.498::1999/06/09-18:00:53.178 [GUM] GumpDoUnlockingUpdate releasing
  lock ownership
378.498::1999/06/09-18:00:53.178 [GUM] GumSendUpdate: completed update
  seq 1677    type 1 context 4098
```

In the following entries, the Cluster service creates an NBT interface for the Cluster IP Address resource, and the Resource Monitor announces that the resource is online (state 2):

```
15c.488::1999/06/09-18:00:53.178 IP Address <Cluster IP Address>:
  Created NBT interface \Device\NetBt_If1 (instance 0x7A85A0AA).
15c.488::1999/06/09-18:00:59.396 [RM] RmpSetResourceStatus, Posting
  state 2 notification for resource <Cluster IP Address>
```

The following entry reports that the Cluster IP Address resource is online.

```
15c.488::1999/06/09-18:00:59.396 IP Address <Cluster IP Address>: IP
  Address 172.16.64.254 on adapter (R) PRO/100B PCI Adapter (TX)
  online
```

The following entry reports the Cluster IP Address resource as online. Note that the old state was 129 (ClusterResourceOnlinePending) and the new state is 2 (ClusterResourceOnline).

```
378.328::1999/06/09-18:00:59.396 [FM] HandleResourceTransition: Resource
  Name = 254ef0e5-1937-11d3-b3fe-00a0c986aa14 old state=129 new
  state=2
```

Bringing resource bf37b6a7-391d-11d2-a397-0000f803f3da (the Cluster Network Name resource) online takes longer. The entries that describe this process are the same as those that describe the process of bringing the Cluster IP Address online, except that there are some new entries.

In the first four entries, the Network Name resource is trying to come online. But it must wait for 254ef0e5-1937-11d3-b3fe-00a0c986aa14, the Cluster IP Address resource on which the Cluster Network Name resource depends, to come online first:

```
378.3a8::1999/06/09-18:00:59.396 [FM] OnlineWaitingTree, 254ef0e6-1937-
  11d3-b3fe-00a0c986aa14 (129) depends on 254ef0e5-1937-11d3-b3fe-
  00a0c986aa14 (2). Start now
378.3a8::1999/06/09-18:00:59.396 [FM] OnlineWaitingTree, Start resource
  254ef0e6-1937-11d3-b3fe-00a0c986aa14
378.3a8::1999/06/09-18:00:59.490 [FM] OnlineResource: 254ef0e6-1937-
  11d3-b3fe-00a0c986aa14 depends on 254ef0e5-1937-11d3-b3fe-
  00a0c986aa14. Bring online first.
378.3a8::1999/06/09-18:00:59.537 [FM] FmpRmOnlineResource: bringing
  resource 254ef0e6-1937-11d3-b3fe-00a0c986aa14 (resid 741136) online.
```

Bringing the Cluster Network Name resource online involves a number of activities that have been covered earlier in this chapter, including the following:

- Global updates.
- Changing the value of an entry for the resource in the cluster key of the registry.
- Writing to the quorum log a change record of the change in registry value.
- Acquiring and releasing shared locks on the Quorum resource.
- Disk arbitration.

The unique activities are reported in the following entries, at the end of which the Cluster Network Name resource has successfully come online:

```
15c.4d4::1999/06/09-18:01:01.177 Network Name <Cluster Name>: Registered
  server name CLUSTER1 on transport \Device\NetBt_If1.
15c.4d4::1999/06/09-18:01:01.177 Network Name <Cluster Name>: Registered
  workstation name CLUSTER1 on transport \Device\NetBt_If1.
.
.
.
15c.4d4::1999/06/09-18:01:01.615 Network Name <Cluster Name>: The zone
for 254.64.16.172.in-addr.arpa. does not accept dynamic DNS registration
updates.
15c.4d4::1999/06/09-18:01:01.630 Network Name <Cluster Name>: Registered
DNS name cluster1.itreskit.com on IP Address 172.16.64.254.
15c.4d4::1999/06/09-18:01:01.630 Network Name <Cluster Name>: Network
Name CLUSTER1 is now online
15c.4d4::1999/06/09-18:01:01.630 [RM] RmpSetResourceStatus, Posting
state 2 notification for resource <Cluster Name>
```

Cluster Successfully Formed

The Cluster service does the following:

- Begins replicating events from the event log to all the nodes in the cluster.

```
378.380::1999/06/09-18:00:51.381 [EVT] EvOnline
378.380::1999/06/09-18:00:51.381 [EVT] Set propagation state to 0001
378.380::1999/06/09-18:00:51.381 [EVT] EvOnline : calling
ElfRegisterClusterSvc
378.380::1999/06/09-18:00:51.412 [EVT] EvOnline: pPackedEventInfo-
>ulSize=3284 pPackedEventInfo->ulNulEventsForLogFile=2
```

- Takes a checkpoint (a snapshot) of the quorum database and replaces the old cluster database checkpoint on the quorum disk with the new one:

```
378.380::1999/06/09-18:00:51.412 [dm]DmUpdateFormNewCluster - taking
a checkpoint
378.380::1999/06/09-18:00:51.412 [LM] LogCheckPoint entry
378.380::1999/06/09-18:00:51.412 [LM] DmpGetSnapshotCb::checkpoint
file name =D:\MSCS\chk67D.tmp Seq#=1661
378.380::1999/06/09-18:00:51.756 [dm] DmpGetSnapShotCb -
DmpGetDatabase returned 0x00000000
378.380::1999/06/09-18:00:51.756 [LM] LogCheckPoint:
ChkPtFile=D:\MSCS\chk67D.tmp Chkpt Trid=1661 CheckSum=86865
378.380::1999/06/09-18:00:51.756 [lm] LogpAppendPage : Writing 1024
bytes to disk at offset 0x00000800
378.380::1999/06/09-18:00:51.756 [lm] LogFlush : pLog=0x000a0650
writing     the 1024 bytes for active page at offset 0x00000c00
378.380::1999/06/09-18:00:51.771 [LM] LogCheckPoint: EndChkpt
written. EndChkPtLsn =0x00000c08 ChkPt Seq=1661 ChkPt
FileName=D:\MSCS\chk67D.tmp
378.380::1999/06/09-18:00:51.771 [lm] LogpCheckpoint : Writing 1024
bytes to disk at offset 0x00000000
378.380::1999/06/09-18:00:51.771 [LM] LogCheckPoint Exit
```

The following entry reports that the Cluster service has successfully pinged its own IP address to make sure the IP address is unique:

```
15c.488::1999/06/09-18:00:59.396 IP Address <Cluster IP Address>: IP
  Address 172.16.64.254 on adapter (R) PRO/100B PCI Adapter (TX)
  online
```

Log Summary of Cluster Formation

When the Cluster service forms a cluster, it does the following:

- Starts the Resource Monitor, the manager of the cluster's resources.
- Brings the Quorum resource online.
- Updates local copies of the cluster database.
- Recreates groups and resources.
- Configures the networks, recreates network and interface objects, registers the networks and interfaces with cluster transport, and then brings them online.
- Brings all resources online.
- Takes a checkpoint of the cluster database.

And the cluster is formed. Resources continue to be brought online after the cluster formation is reported complete.

Failure Scenarios

This section contains selections from cluster log output that was produced when the cluster encountered the following causes of failure:

- Resource DLL is missing
- Intracluster network connection is broken
- Node cannot form cluster because quorum location changed

In the cluster log selections, only entries that are relevant or useful for orienting you within the sequence are included. Omitted entries within those presented are indicated by a vertical ellipsis, as follows:

.
.
.

Resource DLL Is Missing

A support engineer trying to start the Cluster service on a Windows NT 4.0 cluster receives the message, "Could not start the Cluster Server service on [server name] Error 127: The specified procedure could not be found."

The relevant entries in the cluster log are as follows. Note particularly the last two entries, which tell us the reason the procedure cannot be found. The problem is an error in loading Clusres.dll, which manages the resource in question. As a result, the support engineer is not able to create a resource to manage the service and so cannot start it:

```
0ab::11-18:35:52.546 [FM] arbitrate for quorum resource id e558e4a3-
320d-11d2-9b64-00805fa63b1b.
0ab::11-18:35:52.546 [FM] FmpQueryResourceInfo:initialize the resource
with the registry information
0ab::11-18:35:52.562 FmpAddPossibleNodeToList:: Warning, Node 2 not
found
0ab::11-18:35:52.562 FmpRmCreateResource: creating resource e558e4a3-
320d-11d2-9b64-00805fa63b1b in shared resource monitor
075::11-18:35:52.593 [RM] Error loading resource dll clusres.dll, error
127.
075::11-18:35:52.593 [RM] Failed creating resource e558e4a3-320d-11d2-
9b64-00805fa63b1b, error 127.
```

Intracluster Network Connection Is Broken

This section includes cluster log output from two nodes when the intracluster network connection is broken. In this cluster, the intracluster network connection is a single point of failure.

Log from Node 2

Node 2 is the Quorum owner when the intracluster network connection is broken.

The following entry indicates that there has been a loss of communications between nodes. The Cluster service initiates the holding I/O operation when there has been a loss of communications.

```
00000534.00000500::1999/10/21-23:09:01.999 [NM] Holding I/O.
.
.
.
00000534.0000057c::1999/10/21-23:09:02.108 [NM] Checking if we own the
  quorum resource.
.
.
.
```

```
00000534.0000057c::1999/10/21-23:09:02.124 [FM] Successfully arbitrated
  quorum resource a83b4084-3391-4618-890e-8794d4df923b.
.
.
.
00000534.00000500::1999/10/21-23:09:04.905 [ClMsg] Received interface
  unreachable event for node 1 network 1
00000534.00000500::1999/10/21-23:09:04.905 [ClMsg] Received interface
  unreachable event for node 1 network 2
00000534.0000052c::1999/10/21-23:09:04.905 [NM] Communication was lost
  with interface 0bd641f7-7d8c-4d94-9279-d461846b299b (node: NODE1,
  network: clients(1))
.
.
.
00000534.0000052c::1999/10/21-23:09:04.905 [NM] Communication was lost
  with interface ddda464e-7c6d-4439-b27b-cd0da7957162 (node: NODE1,
  network: interconnect)
.
.
.
00000534.0000057c::1999/10/21-23:09:09.123 [NM] Resuming I/O.
.
.
.
00000534.0000057c::1999/10/21-23:09:09.123 [EP] Nodes down event
  received
.
.
.
00000534.00000464::1999/10/21-23:09:09.139 [DM] DmpEventHandler - Node
is down, turn quorum logging on...
```

Log from Node 1

The following log entries are from node 1 and were generated for the same occurrence: the loss of the "interconnect" connection for cluster communications. The following entries, which establish that the network interface is unavailable, are the first indications. Declaring the other node to be down because the interface is unavailable triggers the regroup events noted below as RGP.

```
00000404.000004e4::1999/10/21-23:10:38.039 [ClMsg] Received interface
  unreachable event for node 2 network 2
00000404.00000590::1999/10/21-23:10:38.039 [NM] Communication was lost
  with interface 198ffe74-b7b9-41e5-b95a-25f618eb0c43 (node: NODE2,
  network: interconnect)
00000404.000004e4::1999/10/21-23:10:42.914 [ClMsg] Received node down
  event for node 2, epoch 0
.
.
.
00000404.00000374::1999/10/21-23:10:46.711 [NM] Checking if we own the
  quorum resource.
```

In the following entry, "error 1" means "Incorrect Function." The Cluster service on this node could not read the partition information from the quorum disk prior to asserting a reservation. This is because the other node had reserved the quorum disk after the successful bus reset noted several entries earlier.

```
00000388.0000059c::1999/10/21-23:10:50.351 Physical Disk <Disk E:>:
  [DiskArb]Failed to write (sector 12), error 1.
```

In the following entry, "status 1" means that arbitration for drive E:, the quorum disk, failed. That is, the other node successfully defended its reservation on the disk. The second and third entries following also report the failure:

```
00000388.0000059c::1999/10/21-23:10:50.351 Physical Disk <Disk E:>:
  [DiskArb]Arbitrate returned status 1.
.
.
.
00000404.00000374::1999/10/21-23:10:50.351 [FM] Failed to arbitrate
  quorum resource a83b4084-3391-4618-890e-8794d4df923b, error 1.
.
.
.
00000404.00000374::1999/10/21-23:10:50.351 [RGP] Node 1: REGROUP ERROR:
  arbitration failed.
```

Because arbitration has failed and the nodes are partitioned, the Cluster service on this node shuts down in order to quit participating in the cluster.

```
00000404.00000374::1999/10/21-23:10:50.351 [NM] Halting this node due to
  membership or communications error. Halt code = 1000
.
.
.
00000388.000003cc::1999/10/21-23:10:51.117 [RM] Going away, Status = 1,
  Shutdown = 0.
.
.
.
00000388.0000052c::1999/10/21-23:10:51.148 [RM] NotifyChanges shutting
down.
```

Node Cannot Form Cluster Because Quorum Location Changed

In the following log, the quorum location was changed while the Cluster service was turned off on node 1. After the quorum location was changed and before the Cluster service was restarted on node 1, the Cluster service was turned off on the other node. In this log, the initialization of the Cluster service and the formation of the cluster proceeds normally until the events described by the entries in this section. For the cluster log output of a normal cluster formation operation, see "Forming a Cluster," earlier in this chapter.

The portion of the cluster log reproduced here starts after the Cluster service has done the following:

- Was unable to find a node to sponsor this node's attempt to join the cluster.
- Started a form operation.
- Created the Cluster group and the resources in the Cluster group and then brought them online.
- Started arbitration for and obtained ownership of the Quorum resource and then brought it online.

The Cluster service now tries to open the quorum log:

```
00000460.00000598::1999/10/21-23:58:30.621 [INIT] Attempting to form
  cluster CLUSTER10
00000460.00000598::1999/10/21-23:58:30.746 [FMX] Found the quorum
  resource a83b4084-3391-4618-890e-8794d4df923b.
.
.
.
```

The following entry demonstrates that, according to the node's copy of the cluster configuration database, the quorum location is E:\MSCS\Quolog.log. However, while the node was offline, the quorum directory was moved to another location. Because the other node in the cluster went offline before the Cluster service was restarted on this node, this node cannot get the actual current quorum location.

```
00000460.0000042c::1999/10/21-23:58:31.152 [DM] DmpQuoObjNotifyCb: the
  name of the quorum file is E:\MSCS\quolog.log
.
.
.
```

Note in the following entries that until the Database Manager reports checking for a quorum tombstone and finding one, the node seems able to form a cluster.

```
00000460.0000042c::1999/10/21-23:58:31.168 [LM] LogpCreate : Exit with
  success
00000460.0000042c::1999/10/21-23:58:31.184 [DM] DmpQuoObjNotifyCb:
  Quorum log opened
00000460.0000042c::1999/10/21-23:58:31.184 [FM]
  HandleResourceTransition: Resource Name = a83b4084-3391-4618-890e-
  8794d4df923b old state=129 new state=2
00000460.0000042c::1999/10/21-23:58:31.184 [FM]
  FmpPropagateResourceState: signalling the ghQuoOnlineEvent
00000460.00000598::1999/10/21-23:58:31.184 [DM] DmpChkQuoTombStone -
  Entry
.
.
.
```

In the following two entries, the Database Manager reports that a tombstone file for an old location of the quorum directory was found. For more information about tombstones, see "Applying Quorum Log Changes to the Cluster Database" earlier in this chapter.

```
00000460.00000598::1999/10/21-23:58:31.199 [DM] DmpChkQuoTombStone:A
  tombstone for this resource, and quorum log file was found here.
00000460.00000598::1999/10/21-23:58:31.199 [DM] DmpChkQuoTombStone:This
  is node is only allowed to do a join, make sure another node forms
.
.
.
```

Finally, the following entry indicates that the Cluster service is shutting down.

```
000001bc.000004dc::1999/10/21-23:58:32.902 [RM] NotifyChanges shutting
down.
```

Tips

The following strategies can help you read the cluster log.

Reading the Log in Word or WordPad

Most cluster logs are too big to read in Notepad.

If you read the cluster log in either Microsoft® Word or Microsoft® WordPad, you can take advantage of search and other capabilities to follow thread Ids and resource and group Ids or names throughout the log.

Correlating the Windows 2000 Event Log and the Cluster Log

To home in on the cluster log entries relevant to your problem, use the GUM update numbers reported in the cluster log to correlate cluster log and Windows 2000 event log entries.

You can also use timestamps to correlate events between the two logs, but it is not likely they will align precisely.

Identifying GUIDs in the Registry

Although in Windows 2000 resource and group names as well as GUIDs are logged directly to the cluster log, the following procedure can still be valuable. Note that Regedt32.exe is required for this procedure.

Caution Do not use a registry editor to edit the registry directly unless you have no alternative. The registry editors bypass the standard safeguards provided by administrative tools. These safeguards prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your system. Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. To configure or customize Windows 2000, use the programs in Control Panel or Microsoft Management Console (MMC) whenever possible.

▶ To find a GUID in the cluster registry key

1. In a registry editor, do one of the following:
   - Click the HKEY_LOCAL_MACHINE subtree.

     –Or–
   - From the menu, choose HKEY_LOCAL_MACHINE.
2. Open the HKEY_LOCAL_MACHINE subtree.
   - If the Cluster key is present, go to step 5.
   - If the Cluster service has not started since the last reboot, load the Cluster registry key.
3. To load the Cluster hive (the Cluster key and its subkeys):
   - Click the HKEY_LOCAL_MACHINE subtree.
   - From the registry menu, select **Load Hive**.
   - Navigate to the cluster directory. (The cluster directory is probably %SystemRoot%\Cluster\).
   - Select CLUSDB (not Clusdb.log).

   The **Load Hive** dialog box appears and prompts you to supply a key name.
4. Type:

   Cluster
5. Open the Cluster key.

   GUIDs are located in the Groups, NetworkInterfaces, Networks, and Resources subkeys.
6. Open each of these subkeys to find the GUID of interest.
7. Open the GUID, and note the entry Name.

Logging When Running the Cluster Service With the –debug Option

If diagnostic logging is enabled and the Cluster service fails to start because of a logon error for the service account or another system-related error, the Cluster service might not have a chance to run, and the log file might be empty. If there was a previously existing log file, that file might persist.

An alternative is to run the Cluster service temporarily from a command prompt, with the **-debug** option set. Before running the Cluster service with the **-debug** option, consider the following:

- This method runs the Cluster service outside the normal environment provided by the Service Control Manager.
- To use these procedures, you must be logged on with administrative rights, so you must be logged on with the same account you usually use to run the Cluster service or with the computer's administrator account.
- Do not use the **-debug** option for normal use. The Cluster service does not run as efficiently when this option is set.

▶ To run the Cluster service from a command prompt

1. Open a new command prompt window.
2. Change directories to the cluster directory on the local system.

   The local cluster directory is %SystemRoot%\luster.
3. At the command prompt, type the following:

   CLUSSVC -debug

   In the command prompt window, the Cluster service prints output similar to entries written when running the Cluster service without the **-debug** option.

▶ To print the Cluster service output to a file

- At the command prompt, start the Cluster service by typing the following command:

  CLUSSVC -debug > c:\debug.log

Note When you run Clussvc.exe from a command window, the Resource Monitor does not output to that window. The Resource Monitor is a separate process and does not write to the window used by Clussvc.exe.

State Codes

The following state codes are derived from the Cluster service header files.

State Codes for Cluster Nodes

The state codes for cluster nodes are shown in Table 20.3.

Table 20.3 State Codes for Cluster Nodes

State Code	State
-1	ClusterNodeStateUnknown
0	ClusterNodeUp
1	ClusterNodeDown
2	ClusterNodePaused
3	ClusterNodeJoining

State Codes for Cluster Groups

The state codes for cluster groups are shown in Table 20.4.

Table 20.4 State Codes for Cluster Groups

State Code	State
-1	ClusterGroupStateUnknown
0	ClusterGroupOnline
1	ClusterGroupOffline
2	ClusterGroupFailed
3	ClusterGroupPartialOnline

State Codes for Cluster Resources

The state codes for cluster resources are shown in Table 20.5.

Table 20.5 State Codes for Cluster Resources

State Code	State
-1	ClusterResourceStateUnknown
0	ClusterResourceInherited
1	ClusterResourceInitializing
2	ClusterResourceOnline
3	ClusterResourceOffline
4	ClusterResourceFailed
128	ClusterResourcePending
129	ClusterResourceOnlinePending
130	ClusterResourceOfflinePending

State Codes for Network Interfaces

The state codes for network interfaces are shown in Table 20.6.

Table 20.6 State Codes for Network Interfaces

State Code	State
-1	ClusterNetInterfaceStateUnknown
0	ClusterNetInterfaceUnavailable
1	ClusterNetInterfaceFailed
2	ClusterNetInterfaceUnreachable
3	ClusterNetInterfaceUp

State Codes for Networks

The state codes for networks are shown in Table 20.7.

Table 20.7 State Codes for Networks

State Code	State
-1	ClusterNetworkStateUnknown
0	ClusterNetworkUnavailable
1	ClusterNetworkDown
2	ClusterNetworkPartitioned
3	ClusterNetworkUp

Context Numbers

The update functions for the GUM update type FailoverManager are shown in Table 20.8.

Table 20.8 Update Functions for FailoverManager (GUM update type 0)

Context Value	Update Name
0	ChangeResourceName
1	ChangeGroupName
2	DeleteResource
3	DeleteGroup
4	AddDependency
5	RemoveDependency
6	ChangeClusterName
7	ChangeQuorumResource
8	ResourceState
9	GroupState
10	PostClusterWideEvent
11	GroupNode
12	PossibleNodeForResType
13	GroupIntendedOwner
14	AssignOwnerToGroup
15	ApproveJoin
16	CompleteGroupMove

(continued)

Table 20.8 Update Functions for FailoverManager (GUM update type 0) ***(continued)***

Context Value	Update Name
17	CheckAndSetGroupOwner
65537	FailureCount
65538	GroupOwner
65539	CreateGroup
65540	CreateResource
65541	Join
65542	AddPossibleNode
65543	RemovePossibleNode
65544	CreateResourceType
65545	DeleteResourceType
65546	ChangeGroup

The update functions for the GUM update type GumUpdateRegistry are shown in Table 20.9.

Table 20.9 Update Functions for GumUpdateRegistry (GUM update type 1)

Context Value	Update Name
0	CreateKey
1	SetSecurity
4097	DeleteKey
4098	SetValue
4099	DeleteValue
4100	Join

The update functions for the GUM update type GumUpdateMembership are shown in Table 20.10.

Table 20.10 Update Functions for GumUpdateMembership (GUM update type 2)

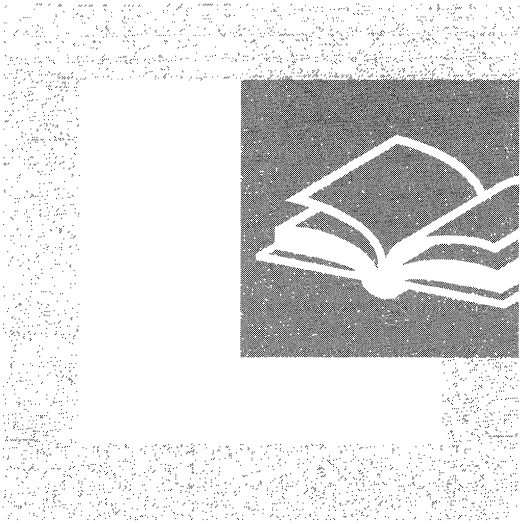
Context Value	Update Name
0	CreateNode
1	PauseNode
2	ResumeNode
3	EvictNode
4	CreateNetwork
5	SetNetworkName
6	SetNetworkPriorityOrder
7	SetNetworkCommonProperties
8	CreateInterface
9	SetInterfaceInfo
10	SetInterfaceCommonProperties
11	DeleteInterface
12	JoinBegin
13	JoinAbort
14	JoinBegin2
15	SetNetworkAndInterfaceStates
16	PerformFixups
17	PerformFixups2
18	AddNode
19	ExtendedNodeState
65537	JoinComplete

Additional Resources

- For more information about the Cluster service, see the Microsoft Platform SDK link and the Microsoft TechNet link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

PART 4

Desktop Configuration Management

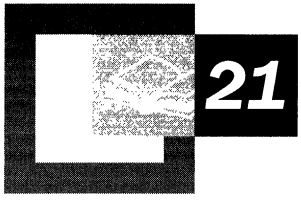


The ability to manage and control desktop configurations is essential for administrators of large networks. Part 4 provides detailed technical information that is useful to desktop administrators when they implement or troubleshoot change and configuration management features in Microsoft® Windows® 2000.

In This Part

CHAPTER 21

Introduction to Desktop Management



To maximize the productivity of users of computer systems while minimizing the total cost of ownership (TCO) associated with these systems, organizations must manage their computer systems efficiently. Microsoft® Windows® 2000 is designed to increase the availability of computer resources and reduce the overall cost of supporting users of Windows 2000 through *Group Policy*–based *change and configuration management*. The primary goal of desktop change and configuration management is to ensure that the computing resources that are necessary for users to do their jobs are available when the users need them. The challenges to desktop management include centralizing control of many personal computers, dealing with multiple computer hardware and software configurations, dealing with user accounts, and updating systems to address changing business needs. You can use Windows 2000 to accomplish all these tasks.

In This Chapter

Related Information in the Resource Kit

- For more information about Remote OS Installation, see "Remote OS Installation" in this book.
- For more information about Group Policy, see "Group Policy" in this book.
- For more information about how to manage software, see "Software Installation and Maintenance" in this book.
- For information about troubleshooting change and configuration management, see "Troubleshooting Change and Configuration Management" in this book.
- For more information about the client-side options that are available in Microsoft® Windows® 2000 Professional, see the *Microsoft® Windows® 2000 Professional Resource Kit* and the *Microsoft® Windows® 2000 Server Resource Kit Deployment Planning Guide*.

Change and Configuration Management

Change and configuration management is a set of Windows 2000 features that simplify basic tasks such as the following:

- Installing an initial operating system on a new computer.
- Managing how software is deployed and installed on personal computers to ensure that users have the software that they require to perform their jobs.
- Managing the configuration of each user's desktop.
- Replacing computers.

Change and configuration management involves managing the ongoing issues that arise as you try to ensure that users are productive when they use their computers to complete their day-to-day work.

Change and Configuration Management			
IntelliMirror			Remote OS Installation
User Data Management	**Software Installation and Maintenance**	**User Settings Management**	
Benefits	Benefits	Benefits	Benefits
My data and documents follow me. Users can have access to the data they need to do their jobs, whether online or offline, when they move from one computer to another on the network. Administrators manage this feature centrally by policy to minimize support costs.	**My software follows me.** Users have the software they need to perform their jobs. Software and optional features install "just-in-time." After it is installed, software is self-repairing. Administrators manage application and operating system upgrades as well as application deployment centrally by policy. This minimizes support costs.	**My preferences follow me.** Users see their preferred desktop arrangements from any computer. A user's personal preferences and settings for desktops or software are available wherever the user logs on. Administrators manage this feature centrally by policy to minimize support costs.	Administrators can enable remote installation of Windows 2000–based operating systems and desktop images on new or replacement computers without preinstallation or on-site technical support.
Technologies	Technologies	Technologies	Technologies
Active Directory Group Policy Offline files Synchronization Manager Enhancements to the Windows desktop Disk quotas	Active Directory Group Policy Windows Installer Add/Remove Programs in Control Panel Enhancements to the Windows desktop	Active Directory Group Policy Offline files Roaming User Profiles Enhancements to the Windows desktop	Active Directory Group Policy Dynamic Host Configuration Protocol (DHCP) Remote Installation Services

Figure 21.1 Change and Configuration Management Features

As Figure 21.1 shows, change and configuration management includes User Data Management, Software Installation and Maintenance, and User Settings Management, which are collectively known as the *IntelliMirror*™ management technologies. IntelliMirror and Remote OS Installation are the Windows 2000 change and configuration management technology set.

IntelliMirror features increase the availability of the user's computer and computing environment by intelligently storing information, settings, and applications. Remote OS Installation installs Windows 2000 Professional operating system and desktop images on new or replacement computers without on-site technical support. When you combine IntelliMirror and Remote OS Installation throughout your organization, you create a system that makes computer replacement easier.

You can use change and configuration management features to perform the following functions:

- Define computing environment settings centrally for both groups of users and computers. Then you can rely on Windows 2000 to enforce those settings.
- Make it possible for users to roam to any computer on their network and have the same computing environment, including access to their data, applications, and preference settings.
- Enable users to find all their data files and network files quickly, even when they are working offline. Offline files are cached locally and are synchronized with the server.
- Manage software installation, updates, and removal from a central location. Applications can also repair themselves if a user inadvertently removes key files.
- Replace a computer quickly and then regenerate its settings, restoring data, applications, preferences, and administrative policies.
- Allow Pre-Boot Execution (PXE)–based remote boot technology–enabled workstations to install an operating system on the local hard disk drive automatically.

Microsoft® Systems Management Server also provides change and configuration management services. You can use Systems Management Server in combination with IntelliMirror and Remote OS Installation. For information about the benefits of combining these technologies, see "Configuring and Maintaining the Network Environment" later in this chapter.

IntelliMirror

The technology behind the IntelliMirror technologies assists with managing user and computer information, settings, and applications. IntelliMirror uses *Active Directory*™ and Group Policy to provide policy-based management of users' desktops through a centrally defined policy that is based on users' business roles, group memberships, and locations. You can configure Windows 2000–based desktops to meet a new user's requirements each time that user logs on to the network.

Depending on the requirements of your organization, you can use the three IntelliMirror features either separately or together. Windows 2000 change and configuration management provides the IntelliMirror capability by using the following specific features:

- User Data Management
- Software Installation and Maintenance
- User Setting Management

User Data Management

User data can follow the user whether the user is online and connected to the network or the user is offline in a stand-alone state. The user's data follows the user because Windows 2000 can store the data in specified network locations while making the data appear local to the user. You can configure which files and folders are available manually, set them up on a per-user basis, or configure them through Group Policy.

User Data Management ensures that the items that users create, such as personal files and documents, are easily accessible and readily available to the user. If users take their work home or on the road, these users still have access to their files. The network files that a user works with when online are automatically cached on that user's computer, and available when he or she is offline. The master version of the file is stored on a server. When users reconnect to the network, any files that they have worked on are synchronized with the network version. Synchronization ensures that users have the same experience whether they are working online or offline.

Because user files are redirected to a server, information technology (IT) staff can protect the centrally stored version of the data. If user data is lost on a local computer because a hard disk drive fails, you can restore that data from the network.

The best method to make users' data follow them is to redirect specific user data folders, such as the My Documents folder, to a network location through Group Policy, and then make this location available to the users for offline use. When a user saves a file to the My Documents folder, the file is actually saved on the network location, and the local computer is synchronized with the network copy. This synchronization occurs in the background and is transparent to the user.

The user uses the computer in the same way whether in stand-alone mode or connected to the network, and the user is therefore unaffected by temporary network outages. When a user works offline, either by choice or because of a network failure, all modifications and changes to user data are made to the local copy. When the computer is reconnected to the network, synchronization with the network copy occurs automatically. When the network copy and the local copy have both changed in the interim, the Synchronization Manager prompts the user to either save both copies or synchronize one copy with the other copy.

User Data Management technologies include Active Directory, Group Policy, Offline Files, Folder Redirection, and Synchronization Manager. These technologies ensure that data is protected, is available offline, and is available from any computer on the network.

Software Installation and Maintenance

You use Software Installation and Maintenance to manage the installation, configuration, repair, and removal of software, including applications, operating system service packs, or software upgrades. Software Installation and Maintenance is designed to facilitate policy-based management of software through the entire software life cycle.

Software Installation and Maintenance provides robust software installation and automatic repair of software to groups of users and computers. By using Software Installation and Maintenance, you can define Group Policy settings that specify which applications that a user can use, regardless of which computer the user logs on to, and you can set how software files update and synchronize on a per-computer or per-user basis. You can assign software to a user or computer, or you can publish applications to a user.

Assigned Applications Assigned applications appear installed on the user's computer; that is, the user sees shortcuts for the applications on that user's desktop or **Start** menu. However, the software is advertised only on the user's computer, which means that the necessary registry entries are made, and shortcuts are placed on the desktop or the **Start** menu. The software actually is installed the first time that a user selects the software. You use this method to deploy software that is resilient and available no matter what the user does, because if the user removes the software, it is re-advertised, which means that it is installed again if the user selects it again.

Published Applications Published applications do not appear installed on the user's computer; that is, there is no evidence of the application on the desktop. Published applications are installed by using **Add/Remove Programs** in Control Panel. You use this method for managing software that is not absolutely necessary for a user to perform a job.

When you deploy applications by either assigning or publishing them, you can update them from the server. Then when the user logs on to the client computer, any new applications or updates are installed. For more information about Software Installation and Maintenance features of IntelliMirror, see "Software Installation and Maintenance" in this book.

User Settings Management

User Settings Management is used to set Group Policy settings that define customizations and restrictions that must be applied to the operating system, desktop environment, and software for each user, including language settings, custom dictionaries, accessibility, desktop configurations, and other user preferences and restrictions.

By using User Settings Management, you can centrally define computing environments for organized groups of users and computers and grant or deny users the ability to do any further customization. When users have permission, they can customize the style and default settings of their computing environment to suit their needs and work habits.

By having IntelliMirror features enabled, you can restore user settings if a computer fails and ensure that a user's desktop settings follow the user if he or she roams to another computer.

User settings follow users because IntelliMirror uses Group Policy and roaming user profiles to store all important user settings.

Settings contain three types of information:

- Vital settings, set by the user and administrator
- Temporary settings
- Local computer settings

When you manage user settings by using roaming user profiles and compatible applications, you ensure that only vital settings are retained while temporary and local computer settings are dynamically regenerated as required. This ensures that users have the same settings on any Windows 2000–based computer to which they log on.

Note The information in this chapter refers to Windows 2000 technologies that support IntelliMirror from the client perspective of Microsoft® Windows® 2000 Server to Windows 2000 Professional. Client computers that are running Microsoft® Windows® 95, Microsoft® Windows® 98, and Microsoft® Windows NT® version 4.0 or earlier cannot use these IntelliMirror technologies.

Windows 2000 Technologies That IntelliMirror Uses

Based on your organization's needs, the ways in which you manage the desktop with Windows 2000 are going to vary. There are many ways that IntelliMirror technologies can work together to provide the secure and flexible network environment that meets the needs of your users. It is important that you understand the underlying technologies that support IntelliMirror.

The following technologies are used to manage the desktop.

- Active Directory
- Group Policy
- Windows Installer
- Offline Files
- Synchronization Manager
- Folder Redirection
- Disk Quotas
- Add/Remove Programs
- Windows Desktop
- Roaming User Profiles

The following is an overview of the philosophy behind the technology and how it affects the desktop.

Active Directory

Active Directory, the directory service that is included with Windows 2000, stores network object information and implements the services that make this information available and usable to users. Active Directory presents this information through a standardized, logical structure that helps you establish the organization of domains and domain resources in a useful way. It is the foundation of Windows 2000 distributed networks.

How your organization defines sites, domains, *organizational units*, and the users and computers that are located in all of these areas in Active Directory is going to affect the resources available on each user's desktop. An organizational unit (OU) is an Active Directory container object used within domains. OUs are logical containers into which you can place users, groups, computers, and other organizational units; you can control what appears on the desktop. Organizational units can contain objects only from its parent domain.

Note An organizational unit is the smallest scope to which you can apply a Group Policy or delegate authority.

Active Directory Users and Computers is a Microsoft Management Console (MMC) snap-in that enables you to find, add, modify, delete, and organize Windows 2000 user accounts, computer accounts, security and distribution groups, and published resources in your organization's directory.

Computers configured as domain controllers are the only computers that the Active Directory Users and Computers console can view. However, the Active Directory Users and Computers snap-in can run on any Windows 2000 Computer. To administer Active Directory components from a computer that is not a domain controller, you can use the optional Administration Tools package to install the Active Directory Users and Computers snap-in.

Table 21.1 lists areas for managing user, computer, and group network access and the administrative tools in Active Directory where this takes place. It also provides you a reference to the management tools in Microsoft® Windows NT® Server version 4.0 where these tasks are performed.

Table 21.1 Active Directory Administrative Tools and Tools Used in Windows NT Server 4.0 for Managing Network Access

Task	Windows NT Server 4.0 Tool	Windows 2000 Tool
Manage user accounts	User Manager	Active Directory Users and Computers
Manage groups	User Manager	Active Directory Users and Computers
Manage computer accounts	Server Manager	Active Directory Users and Computers
Add a computer to a domain	Server Manager	Active Directory Users and Computers
Create or manage trust relationships	User Manager	Active Directory Domains and Trusts
Manage account policy (such as password criteria)	User Manager	Active Directory Users and Computers
Manage user rights (access rights)	User Manager	Active Directory Users and Computers

For more information about Active Directory, see "Active Directory Logical Structure" in this book.

Group Policy

In Windows 2000, Group Policy is the foundation of IntelliMirror. Group Policy requires Active Directory and Windows 2000–based clients. In this environment, you use Group Policy to define and control the state of users and computers in your organization. Group Policy is the MMC snap-in that you use to specify the behavior of users' desktops. You use the Group Policy snap-in to define settings that are contained in a Group Policy object, which in turn is associated with selected Active Directory container sites, domains, or organizational units. Additionally, the effect of Group Policy can be filtered by using memberships in security groups. The system maintains the state of computers without further intervention after you set Group Policy. Table 21.2 lists the components of Group Policy.

Table 21.2 Group Policy Components and Their Descriptions

Component	Description
Administrative Templates	Registry-based policy, (equivalent to System Policy in Windows NT Server 4.0)
Security Settings	Security settings for domains, computers, and users
Software Installation	Assign or Publish applications
Internet Explorer Maintenance	Administer Internet Explorer after deployment
Scripts	Logging on or logging off users and starting up or shutting down computers
Folder Redirection	Redirecting folders and files to the network

Group Policy and its extensions provide a unified replacement for many of the functions of the system policy editor in Windows NT 4.0.

Table 21.3 lists some of the ways in which you can control a user's work environment by enforcing system configuration settings for all computers that are using Group Policy. Table 21.3 also lists the equivalent tools used in Windows NT Server 4.0.

Table 21.3 Group Policy Administrative Tools and Tools Used in Windows NT Server 4.0 for Managing Policies

Task	Windows NT 4.0 Tool	Windows 2000 Tool
Set policies for users and computers in a site	Not applicable	Group Policy accessed through Active Directory Sites and Services
Set policies for users and computers in a domain	System policy editor (Poledit.exe)	Group Policy accessed through Active Directory Users and Computers
Set policies for users and computers in an organizational unit	Not applicable	Group Policy accessed through Active Directory Users and Computers
Use Security Groups to filter the scope of policy	System Policy Editor (Poledit.exe)	Edit the security descriptor for Apply Group Policy on the security tab of the Group Policy object's properties page.
Manage software	Systems Management Server	Software Installation snap-in accessed through the Group Policy snap-in

For more information about the hundreds of Group Policy settings, the order in which Group Policy settings can be processed, and how to filter and block Group Policy inheritance, see "Group Policy" in this book.

Windows Installer

Windows Installer is a client-side service that is used to manage the various phases of installation, maintenance, and removal of software. Windows Installer uses its own database to track files, Component Object Model (COM) components, registry entries, and additional items that make up the state of an installed application. It can also self-repair applications and ensure that applications are installed or removed without overwriting or deleting files required by another application.

For more information about Windows Installer, see "Software Installation and Maintenance" in this book.

Offline Files

The Offline Files feature allows a user or multiple users to gain access to network folders and files wherever they are online or offline. Because these are traditional shares, you can group files in a folder to create offline folders with permissions that allow only specific users or specific groups to gain access to files. Offline Files is often coupled with Folder Redirection to enable data to follow the user and support the User Data Management feature of IntelliMirror.

For more information about Offline Files, see Windows 2000 Server Help.

Synchronization Manager

Synchronization Manager assists in maintaining data between the local computer and a copy of the data that is replicated on the server. It provides a single location where users can go to synchronize any shared files that are available offline, regardless of the type of file or whether the program used to make the file is available offline. By using Synchronization Manager, you can allow users to control when their offline files are synchronized with files on the network. Synchronizing files when users log on and log off ensures that they are working with the latest information from the network; thus the amount of work disruptions that are caused by saving files is minimized. Synchronization can be initiated manually by the user or can be set to automatically synchronize with the network.

You can also control when user data is synchronized by using Group Policy settings. The Group Policy settings that you can apply range from the ability to automatically synchronize data before a user logs off to disabling the ability of a user to synchronize folders and files.

Folder Redirection

Folder Redirection is a way to place data in a set of folders in the user profiles on the network. Folder Redirection is a Group Policy setting that allows you to configure a set of special folders, such as the My Documents folder, from the local computer on to the network. (The My Documents folder is the location on the Windows 2000 desktop where the user can save their documents and graphic files.) For example, you can redirect the My Documents folder, usually stored on the computer's local hard disk, to a network location so that the documents in the folder are available to that user from any computer on the network.

You can combine Folder Redirection with Remote OS Installation and Software Installation and Maintenance for use with computer replacement. If a computer system fails and loses its data, you can use Remote OS Installation to install Windows 2000 Professional, Software Installation and Maintenance to restore applications, and Folder Redirection to restore critical information. Because a network copy of the data exists you can easily reestablish links to critical information

For more information about Folder Redirection and specifically how to apply Folder Redirection for roaming user profiles, see "Desktop Configuration for Roaming and Mobile Users" later in this chapter.

Disk Quotas

Disk quotas track and control disk space use for user data stored on the network. By using disk quotas, you can configure Windows to do the following:

- Prevent a user from using too much disk space.
- Log an event when a user exceeds a specified disk space limit or warning level.

When you enable disk quotas, you can set two values: the disk quota limit and the disk quota warning level. The limit specifies the amount of disk space that a user is allowed to use. The warning level logs a system event when the user nears the quota limit. For example, you can set a user's disk quota limit to 50 megabytes (MB) and the disk quota warning level to 45 MB. The user cannot store more than 50 MB of files on the volume (the area on the server where a user's data is stored). And if they store more than 45 MB of files on the volume, the disk quota system logs a system event.

If a user requires additional disk space, you can specify that they can exceed their quota limit. You can also specify not to log an event when a user exceeds either their quota warning level or their quota limit. Enabling quotas without limiting disk space use is useful when you do not want to deny users access to a volume, but you want to track disk space on a per-user basis.

For additional information about disk quotas, see Microsoft Windows 2000 Server Help.

Add/Remove Programs

Using **Add/Remove Programs** in Control Panel is one way that users can install and update applications. Users can easily add and remove Assigned and Published applications. However, each Assigned application is re-advertised on the desktop, which means that each application is reinstalled the next time the user selects it.

When a user double-clicks **Add/Remove Programs** and then clicks **Add New Programs**, a list of all of the software available to the user appears. The user is able to install only the software that you have determined that the user needs to perform his or her job. If a large number of published applications are available, the user can sort the applications based on administrator-defined categories. Categories are established by domain, not by the *Group Policy object*, which is a collection of Group Policy settings. By using the Software Installation snap-in, you can create and modify categories, and associate Assigned or Published applications with specific categories. The Software Installation snap-in is focused on a Group Policy object; when you create or edit the categories; however, you make changes on a per-domain basis. When you associate a category with a Windows Installer package, you make changes on a per-Group Policy object basis.

When a user selects the applications that he or she wants to install from the list of published applications using **Add/Remove Programs**, the computer obtains the software's advertisement information from Active Directory and then uses this advertisement information to get the application's Windows Installer package. Windows Installer then installs the application based on the application state information contained in its Windows Installer package.

For more information about **Add/Remove Programs**, see Microsoft Windows 2000 Server Help. For information about how to install applications using **Add/Remove Programs**, see "Software Installation and Maintenance" in this book.

Windows Desktop

The modified Windows desktop enhances the look and functionality of the information displayed on the screen, including dialog boxes, wizards, windows, and the **Start** menu, to make Windows 2000 easier to use. In Windows 2000, the standard default options that display on the desktop have been redesigned to reduce clutter on the screen and make it easier for users to find the items that they require. For example, items that users do not require, such as online services and My Computer, are no longer displayed automatically on the desktop.

You can use several Group Policy settings to assist in managing the configuration of the desktop. For example, you can use Group Policy to remove the **Run** command from the **Start** menu. When this policy is in use, users cannot access the **Run** command. By using another Group Policy setting, you can also modify the default setting for the **Start** menu so that options for applications that are not installed on the user's computer appear dimmed. When the user selects the dimmed option, the application installs automatically.

For more information about Group Policy settings that affect the desktop, see "Group Policy" in this book. For more information about the enhancements to the Windows desktop, see the *Windows® 2000 Professional Resource Kit.*

Roaming User Profiles

Roaming user profiles allow users to wander among computers within the corporate network. When using roaming user profiles, users can log on to any computer that is running Windows 2000 within the user's domain. After they log on, all of the user settings and documents they stored on the server in the roaming user profile copy to the local computer. Users can run applications, modify documents, and work on the computer as they normally would until they log off. When a user logs off, the user profile is copied to a server. When the user logs on to another computer, all of that user's profile information is copied to the second computer.

Many of the IntelliMirror technologies work together to specify the information available to a user who is using a roaming user profile. You use roaming user profiles primarily to preserve the user's customizations because the My Documents folder is often redirected to another share.

You can combine roaming user profiles with Remote OS Installation and Software Installation and Maintenance for use with computer replacement. If a computer system fails and loses its data, you can use Remote OS Installation to install Windows 2000 Professional, Software Installation and Maintenance to restore applications, and roaming user profiles to restore critical information. Because a network copy of the data exists you can easily reestablish links to critical information

For a more detailed description on how to set up and manage roaming user profiles, see "Roaming User Profile Settings" later in this chapter.

Remote OS Installation

In Windows 2000, Remote OS Installation is part of change and configuration management. Remote OS Installation simplifies the task of installing the Windows 2000 Professional operating system on PXE-based remote boot–enabled client computers throughout the organization. It allows computers to connect to a networked server during initial start-up, and then it allows the server to perform a local installation of Windows 2000 Professional.

Remote Installation Services (RIS) is the technology that is used during initial start-up before the resident operating system loads. RIS supports clients without an initial operating system or failed computers that need to have the operating system restored. RIS allows computer hardware connected through a LAN to find a networked RIS server and request installation of a new copy of Windows 2000 Professional appropriately configured for the user and computer. Remote OS Installation cannot be used to upgrade an existing operating system.

Remote OS Installation uses these key technologies to install Windows 2000 Professional on a computer:

- Active Directory
- Group Policy
- Dynamic Host Configuration Protocol (DHCP)
- Domain Name System (DNS)
- PXE-based remote boot technology
- Remote Installation Services

Using RIS servers to deploy and upgrade operating systems throughout a company reduces the costs incurred by either preinstalling the client computer or physically visiting each client to install the operating system. Automatically installing the operating system by using Remote OS Installation and Group Policy can reduce the IT staff support overhead in adding new computers to a network and reinstalling operating systems.

You use a RIS server as a remote source, to install the network equivalent of a CD-based installation of either Windows 2000 Professional or a preconfigured Windows 2000 Professional desktop image. The following are descriptions of these two methods.

CD-Equivalent Installation This is similar to setting up a client computer that directly uses the unattended installation options available on the Windows 2000 Professional operating system CD. The source files, however, reside across the network on available Windows 2000–based servers rather than on a local CD.

Preconfigured Desktop Image Installation This allows you to reproduce a working copy of a corporate desktop configuration, including operating system configurations, desktop customizations, and locally installed software. After the reproduced image is configured, it is stored on Windows 2000 RIS servers. On request, the server downloads these images to new computers. The new computer does not need to have identical hardware to the computer on which the image was created. Windows 2000 Professional support for Plug and Play can adjust for hardware differences.

It is important that your DHCP, DNS, and Active Directory servers are configured appropriately to work with Remote OS Installation. These services can be installed either on individual servers or the same server, and these services must be active and available in order to use RIS. RIS uses these components in several ways to detect client computer requests for service.

For more information about DHCP technology and its use, see "Determining Network Connectivity Strategies" in the *Deployment Planning Guide* and "Dynamic Host Configuration Protocol" in the *Microsoft® Windows® 2000 Server Resource Kit TCP/IP Core Networking Guide*. For more information about DNS technology, see "Introduction to DNS" in the *TCP/IP Core Networking Guide*. For more information about Remote OS Installation, see "Remote OS Installation" in this book.

Configuring and Maintaining the Network Environment

IntelliMirror and Remote OS Installation provide you the key management functions necessary for configuring and maintaining users' Windows 2000 Professional desktop environments in a Windows 2000 Server–based environment. This includes operating system installation and disaster recovery. Microsoft® Systems Management Server version 2.0 complements IntelliMirror and Remote OS Installation to provide advanced options for more complex environments.

When used together, IntelliMirror and Remote OS Installation reduce the costs of setting up new Windows 2000 Professional–based computers. These technologies also provide dynamic configuration and repair to make it easy for users to log on anywhere on a network and provide better recovery from computer failures. For example, if you use Folder Redirection, Roaming User Profiles, Software Installation and Maintenance, and Remote OS Installation, you can quickly replace a computer if a user's computer fails. You can quickly reestablish the user's required data on a replacement computer. You can use Remote OS Installation to install the user's operating system and base applications, Software Installation and Maintenance to assign or publish additional software, and Folder Redirection and Roaming User Profiles to restore the user's data. Folder Redirection and Roaming User Profiles store user data on a server and decrease the likelihood that a user loses critical data when his or her computer fails.

Whether you add or replace a computer, or return a repaired computer to the network, you can use Remote OS Installation to provide the services that reinstall the operating system, and you use IntelliMirror technologies to provide the services that quickly regenerate applications and restore key user data and personal computer settings. Used together, IntelliMirror and Remote OS Installation provide a basic set of features to manage your Windows 2000 Professional user community through centrally administered policy.

The following section provides an overview of Systems Management Server and explains when to use IntelliMirror, Remote OS Installation, and Systems Management Server separately or in combination with each other to provide specific management functions.

Microsoft Systems Management Server

Systems Management Server 2.0 is a product that is produced separately from Windows 2000.

In a simple LAN-interconnected network, IntelliMirror and Remote OS Installation provide all the change and configuration management features required to maintain a well-managed network of computers that are running Windows 2000 Professional in a Windows 2000 Server–based environment. However, in a more distributed and complex enterprise, you need to supplement this base set of features with advanced planning, deployment, and diagnostic tools. Systems Management Server provides you with this advanced set of tools. Used together, IntelliMirror, Remote OS Installation, and Systems Management Server provide a comprehensive range of features to track, maintain, and manage all Windows-based desktop computers in Windows NT–based and NetWare-based environments. Systems Management Server 2.0 provides the following management functions:

Planning Tools Systems Management Server uses Windows Management Instrumentation (WMI) and inventory scanning tools to upload detailed hardware and software inventory information into a Microsoft® SQL Server™–based repository. It also provides tools to monitor and report on application use and check for product compliance such as Windows 2000 readiness and Euro currency support. This collection of planning tools helps you monitor the configuration of your environment, complete audits and compliance checks, monitor and restrict application use, and plan for operations such as new software deployments and upgrades.

Deployment Tools By using Systems Management Server, you can distribute and install software in the background to one, ten, or tens of thousands of computers, even when no users are logged on. You can schedule and synchronize the distribution of software to Windows-based computers. This distribution is integrated with software inventory information to allow sophisticated targeting while also allowing detailed status reporting on the progress and success of each scheduled deployment. Systems Management Server is WAN-aware, allowing you to define what percentage of inter-site bandwidth can be used for management tasks. You can replace the software distribution ability available with Windows 2000 Server with the software distribution functionality available in Systems Management Server.

Diagnostics Tools Systems Management Server provides a range of advanced remote diagnostic tools to help manage desktops and servers without the need for on-site visits. This includes tools such as remote control and remote reboot, a Network Monitor to analyze network conditions and performance and a server monitoring tool, which can track critical performance information about Windows NT Server and Microsoft® BackOffice® Server.

Systems Management Server provides advanced management features to a full range of pre-Windows 2000 systems, including Windows 95, Windows 98, Microsoft® Windows NT® version 3.51, Windows NT 4.0, and Microsoft® Windows for Workgroups. In Windows 2000–based environments, Systems Management Server extends the standard management features provided by IntelliMirror to deliver an enterprise-scale management solution. Systems Management Server can assist with these areas:

Active Directory Design Systems Management Server identifies which computers exist on which subnets and which users log on to which computers.

Identifying Which Computers Can Be Upgraded Systems Management Server can generate reports from its hardware and software inventory database, showing computers listed by site, user group, or other basis, including which computers are suitable for upgrade and which computers are currently deficient.

Software Configurations Systems Management Server can generate reports from its software metering information, identifying what software is deployed and where it is still in use. You can use this information, to identify versions of software that are no longer required, and systems that you need to upgrade.

Consolidating Computer Configurations You can use Systems Management Server to deploy service packs and software patches and to remove applications, and standardize network settings. This allows you to get computers to a known baseline, ready for Windows 2000 installation.

For more information about using Systems Management Server to plan your Windows 2000 deployment, see "Windows 2000 Upgrade and Installation" in the *Deployment Planning Guide*, and see the Microsoft Systems Management Server link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Combining Management Solutions

Whether you use IntelliMirror, Remote OS Installation, and Systems Management Server separately or together depends on the environment. For example, Remote OS Installation provides additional value in environments with Windows 2000 Professional. Table 21.4 provides recommendations for combining IntelliMirror, Remote OS Installation, and Systems Management Server to provide the most appropriate solution for a particular environment.

Table 21.4 Windows 2000 Management Features

Client Operating System Mix or Network Type	Single LAN and Simple Multi-LAN with LAN-Speed Interconnects	Complex Multi-LAN and Multi-site Systems
Windows 2000–based systems only	IntelliMirror, Remote OS Installation	Systems Management Server, IntelliMirror, Remote OS Installation
Mixed Windows-based environments including Windows 2000–based systems	Systems Management Server, IntelliMirror, Remote OS Installation	Systems Management Server, IntelliMirror, Remote OS Installation
Windows 3.1, Windows for Workgroups, Windows 95, Windows 98, Windows NT 3.51, and Windows NT 4.0 environments	Systems Management Server	Systems Management Server

You can use IntelliMirror technologies individually or in combination with other Windows 2000 management features. For example, to augment the Software Installation and Maintenance functions of IntelliMirror in multi-LAN environments, IntelliMirror can be used with the Windows 2000 Distributed file system (Dfs) to provide a network of distribution servers.

Managing the Desktop

By using change and configuration management features—specifically Group Policy settings—you can create and maintain the desktop for each user's work environment. When managing the desktop, it is best to determine the type of user and computer configurations that you need and the various settings that are required.

Desktop Configuration

You can centrally define computing environments for users and computers and ensure that users get the correct environment by using IntelliMirror features: User Data Management, Software Installation and Maintenance, and User Settings Management. By using these features, you can add new users and computers, define settings for organized groups of users and computers, and apply changes to groups of users. Furthermore, by using IntelliMirror technologies, you can restore user settings if a computer fails and ensure that desktop settings follow the user if they roam to another computer.

The technologies that support User Data Management and User Settings Management are the same; the main difference between the technologies is how the user perceives them. Although a user is usually very aware of his or her data, the same user might not be aware of settings. Settings are what you control. For example, users do not need to know where they are located in the Active Directory and that various Group Policy settings have been applied.

What is available at the desktop level depends on your network configuration and specifically what Group Policy settings have been applied. You have the flexibility to tightly manage some computers on the network while loosely managing other systems. You control the information and the software that a user can access. The user does not need to be involved. Settings can also include the data that applications need to preserve the user state, such as a user's custom dictionary, .ost files, and data that controls the look and behavior of applications. Users care only that they can easily access the data that they need.

The remainder of this chapter describes a few of the settings to configure when managing the desktop: user and computer configurations, user profiles, and the features that support roaming and mobile users.

User and Computer Configurations

Group Policy is your primary tool for defining and controlling how software, network resources, and the operating system function for users and computers in an organization. In an Active Directory environment, Group Policy is applied to users or computers on the basis of their user or computer accounts that exist in sites, domains, or organizational units. Users and computers are the only types of Active Directory objects that receive policy.

User Configuration User Group Policy settings are the settings located under the User configuration node in Group Policy, which affect users and are obtained when a user logs on.

Computer Configuration Computer Group Policy settings are located under the Computer Configuration node and affect computers. These settings are obtained when a computer starts.

By using Group Policy, you can control the behavior of the client computer and determine the characteristics of the client computer user environment. You can use Group Policy to manage items and software available on the desktop. Group Policy enhances a roaming user's experience by enabling organizations to do the following:

- Set registry settings securely, and without the undesirably persistent behavior exhibited by Windows NT 4.0–type Administrative Templates.
- Specify security-oriented settings by using security options.
- Install software.
- Use Folder Redirection to redirect My Documents, Desktop, and so on, to network locations.
- Use scripts to implement Windows Script Host scripts that run at log on and log off, and start up and shut down.

For more information about Group Policy and how the snap-in extensions expand the capability of the standard Group Policy settings, see "Group Policy" in this book.

Understanding User Profiles

A profile is created the first time that a user logs on to a Windows 2000–based or Windows NT–based computer. Profiles are not user policies. A user profile is a profile that defines the Windows 2000–based environment that the system loads when a user logs on. The user has a profile even if you don't use Group Policy. It includes all the user-specific configuration settings, such as program items, screen colors, network connections, printer connections, mouse settings, and window size and position.

You can use Group Policy to centrally manage everything from the wallpaper on the computer to the applications on the desktop, or you can establish an open environment that allows users to modify their desktop and install the software of their choice. You can assign logon and logoff scripts that tightly control access to the computer; for example, you can map specific network drives, or you can use Group Policy security settings that enable open access to all drives.

A user's data might be stored only on the local hard disk drive or set so that the data roams with the user wherever he or she logs on. User data can include shortcuts to executable files, personal files, and user settings, such as a custom dictionary.

Depending on your network, you or a user might define what appears on the desktop. The following user profiles are available in Windows 2000 Server.

Local User Profile Created the first time that a user logs on to a computer and is stored on a computer's local hard disk. Any changes made to the local user profile are specific to the computer where the changes are made.

Roaming User Profile You create this profile and store it on a server. This profile is available every time that a user logs on to any computer on the network. Any changes made to a roaming user profile are updated on the server.

Mandatory User Profile
A roaming profile enabled by you and the operating system that creates and manages the roaming profile. This can be used to specify particular settings for individuals or an entire group of users. Only system administrators can make changes to mandatory user profiles. Changes made by the user to desktop settings are lost when they log off. This feature is in Windows 2000 for backward compatibility with Windows NT 4.0–based domains. You need to use Group Policy in Windows 2000, instead of mandatory profiles for the same scenarios.

Creating and Editing User Profiles

You generally do not need to do any management of profiles. You need to perform administrative tasks only when you use mandatory profiles or roaming user profiles.

Table 21.5 compares user profile tasks between Windows 2000 and Windows NT 4.0.

Table 21.5 Common User Profile Tasks

Task	Windows NT 4.0 Tool	Windows 2000 Tool
Add a path to a user profile	**User Manager** command on the **Administrative Tools** menu	Group Policy accessed through **Active Directory Users and Computers**, and then under **User Configuration**
View the contents of a user profile (using default installation on a new computer)	**Profiles** in **Winnt, user**	Windows Explorer profile stored in **Documents and Settings, User**
Copy a user profile	**System** in **Control Panel**	**System properties**, and then under **User Profiles**

(continued)

Table 21.5 Common User Profile Tasks ***(continued)***

Task	Windows NT 4.0 Tool	Windows 2000 Tool
Add a home directory to a path	**User Manager** command on the **Administrative Tools** menu	Group Policy accessed through **Active Directory Users and Computers**, and then under **User Configuration**
Add a logon script to a user profile	**User Manager** command on the **Administrative Tools** menu	Group Policy accessed through **Active Directory Users and Computers and then** under **Windows settings and scripts**

In your organization, you might be presented with many types of users and user configurations. For more information about identifying the various types of users and how best to use the different parts of IntelliMirror, see "Applying Change and Configuration Management" in the *Deployment Planning Guide*.

IntelliMirror can work with various user profiles. You can present users with various user and computer configurations, depending on their individual needs, and how the corporate network is established to support those needs. A user might be working as a roaming user or a mobile user — on a computer where he or she is the only user and has free control of the local system, or on a computer in a tightly managed system where you control the desktop configuration.

It is recommended that you use Group Policy to customize your user profiles rather than mandatory user profiles. If you choose to use mandatory profiles, it is important to know the various hardware configurations for each user; if these configurations are available, create and modify the user profile settings on a computer with the same hardware configuration. For example, a user profile determines screen placement and size of windows. Because user profiles can be accessed on various types of client computers and because these computers might have different hardware configurations (for example, having different video cards and monitors), a user profile might not look correct if the type of display is not set properly. If you are creating a mandatory user profile for several users, create a single user profile for the whole group of users only if all the users use computers that have the same type of video hardware.

For more information about how to establish Windows 2000 user profiles and the settings saved in a user profile, see Windows 2000 Server Help.

The following section describes some of the base settings that need to be established for roaming and mobile users.

Desktop Configuration for Roaming and Mobile Users

Roaming and mobile users are the two main types of users for whom you need to provide specific desktop configurations. A roaming user requires the ability to move from computer to computer throughout the organization. Although a roaming user logs on to different computers, the computers are usually connected to a network through a high-speed connection or LAN connection. A mobile user logs on at the same computer but moves to geographical locations; mobile users sometimes connect through a high-speed connection and at other times through a low-speed connection or dial-up line. Each of these users requires a different desktop configuration.

The technologies that you use when you manage the desktops of roaming and mobile users are as follows:

- Software Installation and Maintenance
- Roaming User Profiles
- Folder Redirection
- Offline Files

Software Installation and Maintenance

As previously mentioned, you can use the Software Installation extension of Group Policy to assign, publish, update, or repair software. When you are working with roaming user profiles, you might want to assign applications that users are going to require to do their jobs. You can install these applications on any computer the user travels to. For mobile users, you can assign or publish software. Publishing software adds the software to the list of applications that are available by using **Add/Remove Programs** in Control Panel. You publish software that users find useful but don't necessarily need to perform their jobs. For example, not all users need to know a computer language to perform their jobs. So you can publish software such as Microsoft® Visual Basic® or Microsoft® Visual C++®; users who need these programming languages can choose to install them.

You can also define the settings that control what occurs when an application requires an update or needs to be repaired. If the user deletes a critical file, and the application is set to repair itself, the application locates the server and attempts to fix itself by using Windows Installer technology. This behavior is transparent to the user.

You can establish how the system manages the deployment of applications and day-to-day files that are required by roaming or mobile users. Because mobile users are not always connected to the network, they might need to update their software by using a CD or low-speed network connection.

Roaming User Profiles

When a user who has a roaming user profile logs on to a computer, runs applications, modifies documents, and then logs off, his or her user profile is copied to a server when he or she logs off. When the roaming user logs on to another computer, all of the profile information, including any **Start** menu customizations and the contents of the My Documents folder, is copied to the second computer. If you have a mobile user, or a branch office user who uses the same computer all the time but connects using a low speed network connection, you should always redirect his or her My Documents folder. For example, you can use roaming user profiles combined with Folder redirection of My Documents.

Note When you store user profiles on a server, the user must wait for the information that has changed each time he or she logs on.

Folder Redirection

By using Folder Redirection, you can redirect the path of a local folder to a server location. Users can work with individual or shared documents on a secure server as if the folders are on the local drive.

You not only can redirect the My Documents folder including the My Pictures subfolder, but you can also redirect to a network location: the Desktop, Application Data, and Start menu folders. All of these folders are where the user is likely to store data. These folders are located within the Documents and Settings user profile folder on the local computer. It is best to redirect the folders by using the following setting:

\\<servername>\<sharename>*%username%*

For example, for a user named Alice, whose My Documents files you want to redirect on machine1 in the userdocs network share, you need to use:

\\machine1\userdocs*%username%*

For roaming and mobile users, My Documents is the main folder that you want to redirect. Windows 2000 uses the My Documents folder as the default location to store user documents on the Windows desktop. Standard application dialog boxes also point to applications that use the **Open** dialog box in Windows 2000 to save files to My Documents by default; so there is a greater tendency for the user to save information in this location.

Note When entering this information, let the operating system create the folder. In this way, the folder is created with all of the correct permissions.

For compatibility with Windows NT 4.0, Windows 2000 allows you to use Folder Redirection to redirect the Start menu folder. You do this by having all users point to the same read-only information. As a best practice for Windows 2000–based computers, do not use Folder Redirection to redirect the **Start** menu folder, use Group Policy to control what appears on the Start menu. Using the Group Policy snap-in:

User Configuration\Administrative Templates\Start Menu & Taskbar\

You can set Group Policy settings to:

- Remove user's folders from the **Start** menu.
- Remove common program groups from the **Start** menu.
- Remove **Documents** menu from the **Start** menu.
- Remove Network and Dial-up Connections from the **Start** menu.
- Remove **Favorites** menu from the **Start** menu.
- Remove Search from the **Start** menu.
- Remove Help from the **Start** menu.
- Remove Run from the **Start** menu.
- Add Log off to the **Start** menu.
- Disable Log off on the **Start** menu.
- Disable or remove the Shut Down command.
- Disable drag-and-drop context menus on the **Start** menu.
- Disable changes to taskbar and **Start** menu settings.
- Shade (gray) unavailable **Start** menu shortcuts for Windows Installer programs.

You can also combine Folder Redirection and Roaming User Profiles to increase performance for roaming and mobile users. Besides the improved availability and backup benefits of having the data on the network, users also have performance gains with low-speed network connections and subsequent logon sessions. Because only some of their documents are copied, these users experience performance gains when their profile is copied from the server. Not all of their data in the user profile is transferred to the desktop each time they log on—only the data that they require is transferred.

When you combine the use of Folder Redirection and Roaming User Profiles, you enable the ability to provide fast computer replacement. If a user's computer goes down, the data that a user requires can quickly be reestablished on a replacement computer.

For information about a user with a roaming user profile who roams from a Windows 2000–based desktop to a Windows NT–based desktop, see the ResourceLink link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Offline Files

By using Offline Files, you can designate that users can continue to work with a copy of network files even when they are not connected to a network. If your organization has mobile users that use portable computers to work, Offline Files gives them access to their files when they are not connected to the network, and ensures that they are always working with the current version of network files. By using a cached version of the files, users can open and update files even when they are not connected to the network. Offline Files stores the data in the computer's cache to make network files available offline. The cache is a portion of disk space that a computer accesses when it is not connected to the network. The view of shared network items that you have made available offline remains as it is when connected, even if users lose a connection to the network or they remove a portable computer from the docking station. Users can continue to work with the Offline Files as they normally do. Users have the same access permissions to those files and folders as when they are connected to the network. When users dock a portable computer and the network connection is restored, any changes they made while working offline are updated to the network.

If two users on the network make changes to the same file, they can save their version of the file to the network, or keep the other user's version, or save both. Sometimes, the file date and size information is not enough to determine which version of the file must be kept. When you are in doubt about which version to keep, save both versions and then review them. After reviewing the files, delete the one that you do not want.

Note Shared files or folders on a Microsoft Windows 2000 network can be available offline. You can make files available for offline use from any computer that supports server message block–based file and printer sharing, including Windows 95, Windows 98, and Windows NT 4.0. The Offline Files feature is not available on Novell NetWare networks. You can set up shared folders so that what is made available offline to a user is either every network file that a user opens in that shared folder or only the network files that the user selects.

Offline Files is a completely stand-alone technology, which means that you don't need to pair it with Folder Redirection and set up and configure network shares, but it works well if you do pair the two technologies. For example, if a shortcut to a file is available offline, that file is made available offline, but if a shortcut to a folder is available offline, the contents of that folder are not available offline. If you pair the two technologies, Offline Files and Folder Redirection, the shortcut and the folder are available offline.

Cache Settings

Users can also work with the information stored in the computer cache, when working offline. If more than one user is using the shared information, there are three ways to specify if and how files within the shared folder are cached locally. Figure 21.2 shows caching settings.

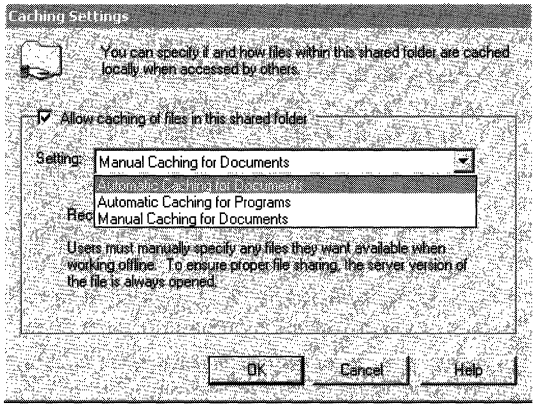


Figure 21.2 Caching Settings

By using the manual caching for documents, users manually specify any files that they want available when they are working offline. Automatic caching for documents is recommended for folders that contain user documents. Opened files are automatically downloaded and made available when users work offline. Older copies of the files are automatically deleted to make room for newer and more recently accessed files. The automatic caching of programs is used for folders with read-only data or run-from-the-network applications. To ensure proper file sharing, the server version of the file is always opened.

Roaming User Profile Settings

To create a roaming user profile there are three basic tasks:

1. Log on to a server
2. Create a network share to store the profile
3. Designate the user as a roaming user.

For more information about how to create a roaming user or mandatory profile, see Windows 2000 Server Help.

In Figure 21.3, you can see a user profile for a user called Alice. The user's data and individual computer settings are stored on the network. When roaming users log on to the network, their desktop settings, stored data, and all his or her user applications are copied to the computers that he or she is using. This provides the user with the ability to log on and have access to data from any computer within the organization's network. You can specify whether all or a portion of a user's profile is available. When a roaming profile is used, this might lengthen the time that it takes a user to log on and log off depending on the size of the user profile, because profiles download from the server when a user logs on and upload to the server when they log off.

In an environment where users are permitted to use more than one computer and where roaming user profiles are in use, it is recommended that temporary and local computer settings not roam with users. Temporary and local computer settings that roam with users can cause unnecessary overhead, including the time to download the files that need to be transferred when users log on to a system, and the differences between computers can disrupt the roaming function.

In Figure 21.3, hidden files are displayed to show all areas of the profile and what each area contains; items in bold are visible to a user who is logged on.

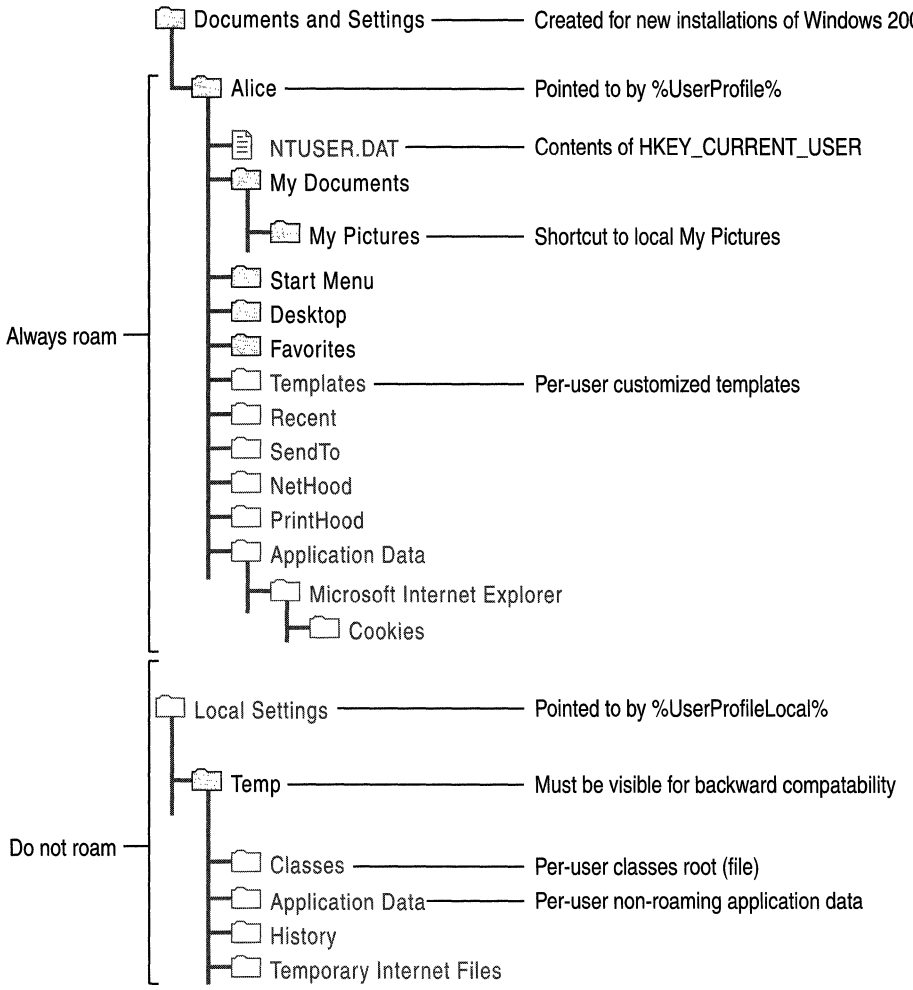


Figure 21.3 Default User Profile

When looking at a user profile, by default the only folders displayed in Windows Explorer are My Documents, My Pictures, Favorites, Start menu, and Desktop folders. The Templates, Recent, Send To, NetHood, PrintHood, and Application Data folders are hidden folders and therefore do not appear in Windows Explorer. You can show hidden files and folders in order to view the hidden folders.

The following user profile local settings folders do not roam by default: Temp, Classes, Application Data, History, and Temporary Internet Files. The Temp directory is visible to the user and is available for backwards compatibility.

If the server is not available, the local cached copy of the roaming user profile is used. If the user has not logged on to the computer before, a new local user profile is created. In either case, if the centrally stored user profile is not available when the user logs on, it is not updated when the user logs off. If the user profile is not downloaded because of server problems, it is not uploaded when the user logs off.

Enhancements to Roaming User Profiles

In Windows 2000, there are several enhancements that increase the usability and resilience of working with roaming user profiles. These enhancements are part of User Data Management and User Settings Management. The following enhancements are in Windows 2000:

- Merge algorithm
- Nonroaming folders
- Profile location
- Quotas on profile size

Merge Algorithm

The Windows 2000 merge algorithm supports the merging of user profiles at the file level and support for last writer wins. New files and updated files are not deleted or overwritten. When a document is updated, the new algorithm compares the timestamp of the destination file with the timestamp of the source file. If the destination file is newer, it is not overwritten.

When a user logs on to a computer, the current time is saved; when the user logs off, the timestamp is used to determine which files are new in the server profile and which files have been deleted in the local profile. For example, if the server profile has a document in the My Documents folder called Review.doc and this file does not exist in the local profile, either it is a new file from a different computer, or it was originally in the local profile and the user deleted it. By knowing the time when this new profile was loaded, you can compare it against Review.doc. If Review.doc was created or written to after the profile load time, the file must be preserved because it came from a different source. If the Review.doc timestamp is older than the load time, Review.doc must be deleted because it would have been copied to the local computer at load time.

In addition, some files might need to be removed from the local cache so that items that were deleted between sessions remain deleted. For example:

1. The user logs on to computer A.
2. The user creates or modifies a document on computer A.
3. The user logs on to computer B.

4. The user logs off computer B; computer B has a copy of the document.
5. The user deletes the document and logs off computer A.

To make sure that the files are deleted, the cached version of the profile is synchronized with the profile server when a user logs on. All files in the local cache that are not present in the server and that were not modified since the last logoff time are deleted. By using these changes, Windows 2000 can merge user profiles.

Windows NT 4.0 Merge Algorithm

The merge algorithms for Windows 2000 and Windows NT 4.0 work differently. Windows 2000 does not display a confusing message when you use offline files. The message that appears when you use Windows NT 4.0 prompts the user to pick a local or roaming profile. Windows 2000 also changes how the previous algorithm works.

In Windows NT 4.0, the algorithm is an Xcopy with full synchronization support. That is, it has the ability to mirror a profile from one location to another, and any extra files or directories in the destination location are deleted. The algorithm is based on the concept that there is only one master profile at any one time. When the user is logged on, the master profile is on the local computer. When the user is not logged on, the master profile is on the server. This works well for users who log on to only a single computer; but users who log on to multiple computers at the same time might lose data. A detailed description of the Windows NT 4.0 algorithm is as follows:

1. The user logs on to computer A, the primary computer.
2. The roaming profile is Xcopied from the server location to the local profile location.

For example, change the setting for a Windows NT 4.0–based computer from \\scratch\scratch\alice to C:\winnt\profiles\alice.

3. The user creates documents, changes colors, and so on. All of these changes are stored in the local profile location.
4. As the user logs off the computer, the profile is Xcopied from the local profile location back to the server location.

This is an exact mirroring process. If there are any extra files in the server location, they are deleted to make sure that the server location is a duplicate of the local profile.

Examples of Windows NT 4.0 Merge Algorithm Issues

When using Windows NT 4.0, a problem arises if the user has two or more computers. Building on the preceding example:

1. The user logs on to computer A.
2. The user logs on to computer B.
3. The user creates a document on computer A and stores it in the user profile.
4. The user logs off of computer A.
5. The user logs off of computer B.

The document that the user created in step 3 is deleted because, from the perspective of computer B, the master profile is stored locally. The extra files on the server must be deleted so that the local profile is currently the master server profile.

A similar data loss problem occurs when files are modified. For example, suppose that the user has a document called Document.doc in his or her My Documents folder in the server copy of the profile:

1. The user logs on to computer A.
2. The user logs on to computer B.
3. The user modifies the document on computer A.
4. The user logs off computer A.
5. The user logs off computer B.

The changes made to the document on computer A are lost because when the user logged off computer B, the computer overwrote the new version of the document with the old one; the computer is programmed to recognize that it had the master version of the profile.

Nonroaming Folders

In Windows 2000 and Windows NT 4.0 Service Pack 4, roaming user profiles are copied from the server to the client when the user logs on, and copied back when the user logs off. However, they introduce a per-user local settings folder into the user profile that is not copied during log on or log off sessions. In this folder, operating system components and other applications can store nonroaming per-user data.

For example, Microsoft® Internet Explorer can store a user's Favorites in the roaming portion of the user profile and store the temporary Internet files in the local, nonroaming portion of the user profile. By default, the Temp and Temporary Internet Files folders are excluded from the roaming user profile. You can configure additional folders to not roam by using the Group Policy snap-in:

User Configuration\Administrative Templates\system\Logon/logoff

and select the policy: Exclude Directories In Roaming Profile.

If you enable this policy, you can exclude additional folders. You cannot use the Exclude Directories In Roaming Profile to include the Temp and Temporary Internet Files folders in a roaming user profile.

Profile Location

When no preconfigured server-based roaming user profile exists for a user, the first time that a user logs on to a computer, a user profile folder is created for the user name. The contents of Default User are then copied to the new user profile folder. The user profile, along with the common program group settings in the All Users folder, creates the user's desktop. When the user logs off, any changes made to the default settings during the session are saved to the new user profile folder. The user profile in Default User remains unchanged.

If the user has a user account on the local computer in addition to a domain user account, or if the user has more than one domain user account, the local user profile is different for each account because different user profiles are generated for each user who logs on. When the user logs off, the settings that are changed are saved to only one user profile, that is the profile on the account with which the user logged on.

Table 21.6 shows the profile location changes based on the status of the computer and whether it has a new operating system or if the operating system was based on an upgrade. The profile moves out of the system directory to assist in computer lock-down scenarios. On upgrade computers, the profile does not move. Some applications might have hard-coded paths that break if you move the profile location.

Table 21.6 Default Location for Special Folders That Have Not Been Redirected

Operating System	Location of Special Folders
Windows 2000 new installation (no previous operating system)	*%SystemDrive%*\Documents and Settings; for example, C:\Documents and Settings
Windows 2000 upgrade of Windows NT 3.*x* or Windows NT 4.0	*%SystemRoot%*\Profiles; for example C:\WinNT\Profiles
Windows 2000 upgrade of Windows 95 or Windows 98 with user profiles disabled	*%SystemDrive%*\Documents and Settings; for example, C:\Documents and Settings
Windows 2000 upgrade of Windows 95 or Windows 98 with user profiles enabled	*%SystemRoot%*\Profiles; for example, C:\Windows\System\Profiles

For new installations of Windows 2000, there is a switch to Unatted.txt that you can use to specify the location of the Profiles folder when you install Windows 2000. To do this, set the following in the GUIUnattended section of the unattended file.

profilesdir = "%systemdrive%\profiles"

Note After Windows 2000 is installed, there is no supported way to move the Profiles folder.

Quotas on Profile Size

The Proquota.exe program is a tool that you can set to monitor the size of a user's profile. If an individual user's profile exceeds the predetermined file limit, the user cannot log off from the computer until the user reduces the size of their files.

You can manage the quota size by using the following Group Policy snap-in:

User Configuration\Administrative Templates\system\Logon/logoff

and then select the policy: Limit Profile Size.

If you are combining Folder Redirection of My Documents and roaming user profiles, it is best to not use quotas on the profile. The items that would normally be written to the user profile are done so on behalf of the user by the operating system and applications, so the user is not aware of them. Examples of these files include Custom.dic and Favorites. You can also use the policy that removes cached versions of the profile on logoff, if you are concerned with disk size on a multi-user computer—for example, a public computer where thousands of users can log on.

Specifying Security on the Desktop

By using the Security Settings extensions to the Group Policy snap-in, you can ensure that users have access to the applications and data that they require. You can also restrict access to only the applications, services, and data that the users require.

You can block policy inheritance at the site, domain, or organizational unit level, or you can enforce policy from above to verify that particular policy settings are applied. For example, you can enforce policies that are applied at the domain level so that they are not overwritten by policies that are applied last in a lower-level organizational unit. Optionally, you can enforce Group Policy on child organizational unit containers by setting No Override on the Group Policy object. You can also prevent inheritance of Group Policy from parent directory containers.

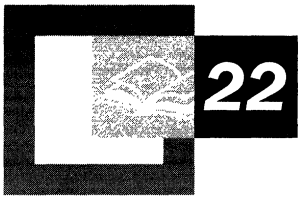
For more information about using the Security Settings extension to Group Policy, see "Group Policy" in this book. For information about setting security options, see Windows 2000 Server Help. For more information about Security Management, see the chapters under "Distributed Security" in this book.

Additional Resources

- For more information about Folder Redirection, see the Microsoft Knowledge Base link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. Search the Knowledge Base for the keywords "Folder Redirection" and "Windows 2000."

CHAPTER 22

Group Policy



Group Policy is one of the important Change and Configuration Management technologies provided in the Microsoft® Windows® 2000 operating system. Administrators use Group Policy to specify options for managed desktop configurations for groups of computers and users. Group Policy is flexible and includes options for registry-based policy settings, security settings, software installation, scripts, computer startup and shutdown, user logon and logoff, and folder redirection. Microsoft® Windows® 2000 Server includes hundreds of Group Policy settings you can configure. Group Policy allows an organization to reduce total cost of ownership by allowing administrators to enhance and control users' desktops.

In This Chapter

Related Information in the Resource Kit

- For more information about Active Directory™, see the chapters on Active Directory in this book.
- For more information about Change and Configuration Management see "Introduction to Desktop Management" in this book.
- For more information about Access Control see "Access Control" in this book.
- For more information about Group Policy backup see "Active Directory Backup and Restore" in this book.
- For information about the Software Installation snap-in, see "Software Installation and Maintenance" in this book.

Group Policy Overview

Group Policy allows you to stipulate users' environments only once, and to rely on the operating system to enforce them thereafter.

Group Policy objects are not profiles. A profile is a user environment setting that a user can change, such as: desktop settings, registry settings in NTUser.dat files, profiles directory, My Documents, or Favorites. You, as the administrator, manage and maintain Group Policy, an MMC hosted administrative tool used to set policy on groups of users and computers.

By default, Group Policy is inherited from site, to domain, and finally to the organizational unit level. The order and level in which you apply Group Policy objects (by linking them to their targets) determines the Group Policy settings that a user or computer actually receives. Furthermore, policy can be blocked at the Active Directory site, domain, or organizational unit level; or policy can be enforced on a per Group Policy object basis. This is done by linking the Group Policy object to its target and then setting the link to no override.

By default, Group Policy affects all computers and users in the site, domain, or organizational unit, and does not affect any other objects in that site, domain, or organizational unit.

Note In particular, Group Policy does not affect security groups.

Instead, you use security groups to filter Group Policy; that is, to alter its scope. This is done by adjusting the Apply Group Policy and the Read permissions on the Group Policy object for the relevant security groups, as explained later in this chapter.

Note The location of a security group in Active Directory is irrelevant to Group Policy.

Windows NT 4.0 and Windows 2000 Policy Comparison

Microsoft® Windows NT® 4.0 introduced the System Policy Editor (Poledit.exe), a tool that you use to specify user and computer configurations that it stores in the Windows NT registry. Using the System Policy Editor, you control the user work environment and enforce system configuration settings for all domain computers running Windows NT Workstation 4.0 or Windows NT Server 4.0. System Policy settings are registry settings that define the behavior of various components of the desktop environment.

In Windows 2000, you can create a specific desktop configuration for a particular group of users and computers by using the Group Policy snap-in. For Windows 2000 clients, the Group Policy snap-in almost entirely supersedes the System Policy Editor. It allows management of desktop configurations for large, possibly nested, and even overlapping, groups of computers and users. Non-local Group Policy objects exert their effect by being linked to any number of targets, which can be sites, domains, or organizational units in Active Directory.

System Policy in Windows NT 4.0, Windows 95, and Windows 98

The System Policy settings you specify with the System Policy Editor (Poledit.exe):

- Are applied to domains.
- Can be further controlled by user membership in security groups.
- Are not secure. They can be changed by a user with the registry editor (Regedit.exe).
- Persist in users' profiles, sometimes beyond their useful lives. After a registry setting is set using Windows NT 4.0 System Policy, the setting persists until the specified policy setting is reversed or the user edits the registry.
- Are limited to administratively mandated desktop behavior based on registry settings.

Note Windows NT 4.0 registry settings can be problematic when a user's security group membership changes. You might need to manually update or remove the registry settings.

The Group Policy snap-in provides built-in features for registry-based policy, security settings, software installation, scripts, and folder redirection. The Group Policy settings that you create are contained in a Group Policy object. Each Windows 2000–based computer has one local Group Policy object, and can also be subject to any number of non-local (that is, Active Directory–based) Group Policy objects.

The policy settings you specify using Group Policy represent the primary method for enabling centralized change and configuration management in Windows 2000.

Group Policy settings:

- Can be associated with sites, domains, and organizational units.
- Affect all users and computers in the site, domain, or organizational unit.
- Can be further controlled by user or computer membership in security groups.
- Are secure. Only an administrator can change the settings.

- Are removed and rewritten whenever policy changes.
- Can be used for finely tuned desktop control and to enhance the user's computing environment.

Note Windows NT 4.0 System Policy settings in the registry sometimes persisted past their useful life because these settings remained in effect until they were explicitly changed. Windows 2000 Group Policy settings do not persist past their useful life because Windows writes them to the following secure registry locations, and removes them when a Group Policy object no longer applies. The registry locations are \Software\Policies and \Software\Microsoft\Windows\CurrentVersion\Policies.

For a detailed comparison of Windows NT 4.0 System Policy as compared to Windows 2000 Group Policy, see "Applying Change and Configuration Management" in the *Microsoft® Windows® 2000 Server Resource Kit Deployment Planning Guide*.

Active Directory Structure and Group Policy

Group Policy implementation is one of the considerations in planning the Active Directory structure for your organization. The basic units of Group Policy are Group Policy objects. These are basic units in the sense that you link (or do not link, as the case might be) an entire Group Policy object at a time. It is not possible to link only a subset of a Group Policy object to a target. Using security groups to filter the scope of Group Policy also has the effect of turning the entire Group Policy object on or off; it does not function on only part of a Group Policy object. (Notwithstanding the fact that the Software Installation and Folder Redirection extensions of Group Policy exploit permissions to tailor the behavior of those particular extensions based on security group membership.)

There are two types of Group Policy objects: local Group Policy objects and non-local Group Policy objects.

Note Each Windows 2000–based computer has only one local Group Policy object.

In the rest of this section, all Group Policy objects are non-local unless otherwise specified.

Group Policy objects are stored in a Windows 2000 domain, and their effects are enabled on sites, domains, or organizational units to which they are linked.

- A Group Policy object linked to a site (using Active Directory Sites and Services) applies to all domains at the site.
- A Group Policy object applied to a domain applies directly to all users and computers in the domain and by inheritance to all users and computers in organizational units (and in generic Active Directory containers) farther down the Active Directory tree as seen in the Active Directory Users and Computers namespace.
- A Group Policy object applied to an organizational unit applies directly to all users and computers in the organizational unit and by inheritance to all users and computers in organizational units (and in generic Active Directory containers) farther down the Active Directory tree as seen in the Active Directory Users and Computers namespace.

It is not possible to link a Group Policy object to a generic Active Directory container. (A generic Active Directory container is identifiable by its plain folder icon in the Active Directory Users and Computers console. The icon for an organizational unit is similar, except that a small book is superimposed on the folder.) However, users and computers in generic Active Directory containers do receive policy by inheritance from Group Policy objects linked at a higher level of Active Directory. For example, the **Users** and **Computers** containers you see in Active Directory Users and Computers cannot have Group Policy objects linked directly to them, but they do receive domain-linked Group Policy objects by means of inheritance.

The local Group Policy object is applied first. Then site-linked Group Policy objects are applied in administratively specified order, then domain-linked ones in specified order, and lastly organizational unit-linked Group Policy objects beginning at the highest (in Active Directory hierarchy) organizational unit containing the user or computer account and ending with the lowest (closest to the user or computer) organizational unit containing the user or computer. At each organizational unit, any Group Policy objects linked to it are applied in administratively specified order.

The order of application detailed in the previous paragraph (1. Local, 2.: Site, 3. Domain, 4. Organizational Unit) is significant to the architect of Active Directory, because by default, policy applied later overwrites policy applied earlier for each setting where the later applied policy was either **Enabled** or **Disabled**. Settings that are **Not Configured** don't overwrite anything—any **Enabled** or **Disabled** setting applied earlier is allowed to persist.

This is the default behavior. Mechanisms exist that let you either force or prevent Group Policy objects from affecting groups of users or computers. The most powerful mechanisms for avoiding the default behavior are the **No Override** and **Enforce Policy Inheritance** settings. It is best to minimize the use of these.

Managing Group Policy

It is important to understand the following topics as they relate to managing Group Policy:

Group Policy Infrastructure and Mechanics. How Group Policy works, including linking Group Policy objects and filtering the scope of Group Policy using security groups.

Administrative Requirements for Using Group Policy. The rights you must have to use Group Policy in an Active Directory environment.

Microsoft Management Console Snap-in Extension Model. An explanation of what you see in the MMC console with Group Policy and its extensions in place.

Group Policy Infrastructure and Mechanics

In this section you learn about Group Policy objects, links to make them exert their effects, the snap-in you use to edit them, and security groups to refine their scope.

Group Policy Objects and the Group Policy Snap-in

You can think of Group Policy objects as the documents associated with the Group Policy snap-in. This is somewhat analogous to the association of .doc files with Microsoft® Word, or .txt files with Notepad; however, the analogy is not perfect.

Changes to a Group Policy object are not deferred until an explicit Save is executed, but take place during the actual edit.

Note You cannot open Group Policy objects in read-only mode.

Links to Sites, Domains, and Organizational Units

You can link Group Policy objects to specific sites, domains, or organizational units, thus maximizing and extending the power of Active Directory. Data within Group Policy objects is evaluated by the affected clients, which exploit the hierarchical nature of Active Directory to determine precedence of Group Policy settings in cases of conflict.

Access to the Group Policy Snap-in

You create a non-local Group Policy object by using the Group Policy snap-in, either as an extension to Active Directory snap-ins, or as a stand-alone MMC console.

The most common route to the Group Policy snap-in is from Active Directory Users and Computers. This allows you to link Group Policy objects to domains or organizational units. You can also access Group Policy through Active Directory Sites and Services. This is how you link Group Policy objects to sites. From these two Active Directory consoles, Group Policy is accessible by means of a context menu. You right-click the site, domain, or organizational unit, point to Properties, and then click the **Group Policy** tab. For specific examples on how to create a Group Policy object, see Windows 2000 Help.

Filtering by Security Group Membership

You can filter the effects of Group Policy on computers and users by using membership in security groups and setting discretionary access control list (DACL) permissions. This implementation ensures faster processing of Group Policy objects than would otherwise be possible. Furthermore, by using security groups, you can limit who in your organization can create Active Directory links to Group Policy objects, as well as who has access to create and modify Group Policy objects.

For details, see "Using Security Groups to Filter and Delegate Group Policy" later in this chapter.

Administrative Requirements for Using Group Policy

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), and you must have Modify Rights to the selected directory site, domain, or organizational unit. The system volume folder is created when you install a Windows 2000 domain controller or promote a Windows 2000 server to domain controller.

By default, Group Policy affects all computers and users in an Active Directory site, domain, or organizational unit to which the Group Policy object is linked. However, you can filter the effects of Group Policy, based on users' or computers' membership in Windows 2000 security groups. To filter Group Policy, you use the **Security** tab on a Group Policy object's **Properties** page to set permissions. You also use permissions to delegate the use of the Group Policy snap-in.

Microsoft Management Console Snap-in Extension Model

The main nodes of the Group Policy snap-in are MMC snap-in extensions. These extensions include Administrative Templates, Scripts, Security Settings, Software Installation, Remote Installation Services, Internet Explorer Maintenance, and Folder Redirection.

By default, all the available Group Policy snap-in extensions load when you start the Group Policy snap-in. You can modify this default behavior by using the MMC method of creating custom consoles and by using policy settings to control the behavior of MMC.

See the settings under User Configuration\Administrative Templates\Windows Components\Microsoft Management Console\ and its subfolders for Group Policy settings concerning MMC.

Developers can create an MMC extension to the Group Policy snap-in to expand its capability to provide additional settings. These snap-in extensions can, in turn, be extended. An example of such an extensible snap-in is the Security Settings snap-in, which includes several snap-in extensions.

For information about creating MMC consoles for delegating Group Policy and related tasks, see "Delegating Control of Group Policy" later in this chapter.

For information about the Group Policy snap-in extensions, see "Extensions to the Group Policy Snap-in" later in this chapter.

For more information about Microsoft Management Console and Group Policy, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Configuring Group Policy

To use Group Policy, you need to know the features of the Group Policy user interface and their roles in configuring Group Policy.

Group Policy Snap-in Namespace

The root node of the Group Policy snap-in displays 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

For example:

Default Domain Policy [MSMSRV01.Reskit.com] Policy

The next level of the namespace has two nodes: Computer Configuration and User Configuration. These are the parent folders that you use to configure specific desktop environments and to enforce Group Policy on groups of computers and users, respectively, on the network.

Computer Configuration

Computer configuration includes all computer-related policy settings that specify operating system behavior, desktop behavior, security settings, computer startup and shutdown scripts, computer-assigned application options, and application settings. Computer-related Group Policy is applied when the operating system initializes and during the periodic refresh cycle, explained later in this document. In general, computer policy takes precedence over conflicting user policy. Figure 22.1 shows the Group Policy Snap-in Console Computer Configuration.

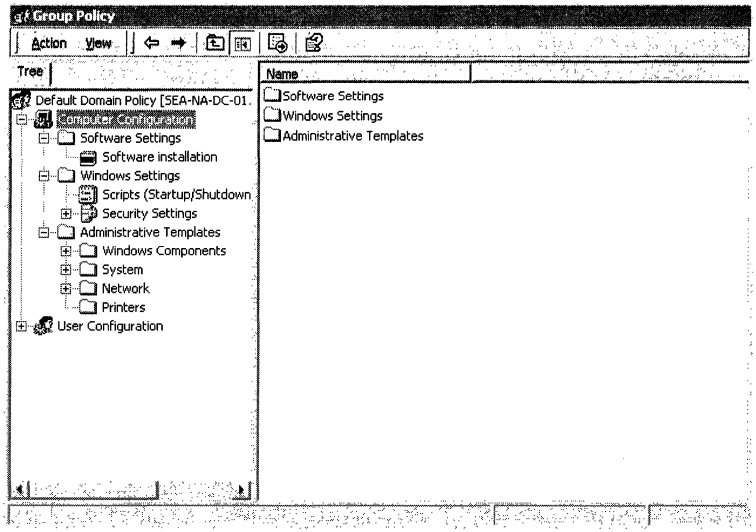


Figure 22.1 The Group Policy Snap-in Console Computer Configuration

User Configuration

User configuration includes all user-related policy settings that specify operating system behavior, desktop settings, security settings, assigned and published application options, application settings, folder redirection options, and user logon and logoff scripts. User-related Group Policy is applied when users log on to the computer and during the periodic refresh cycle. Figure 22.2 shows the Group Policy Snap-in Console User Configuration.

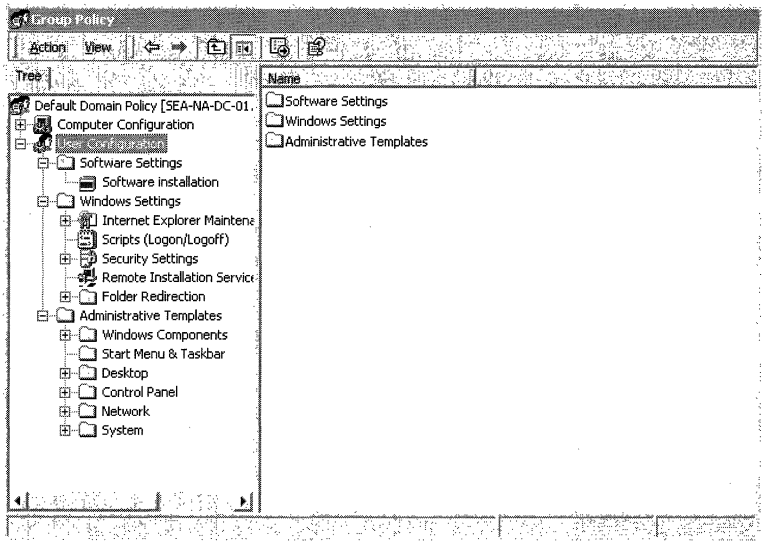


Figure 22.2 The Group Policy Snap-in Console User Configuration

In certain strictly managed computing environments, it is useful to mandate a specific desktop configuration regardless of which user logs on to the computer. Schools, libraries, public kiosks, some laboratories, and reception areas are candidates for policy of this sort. You implement this by appending (or, more severely, replacing) the User Configuration settings for the user account with the User Configuration settings for the computer account. This process is called loopback and it is explained in "Group Policy Loopback Support" later in this chapter.

There are several child nodes under the Computer Configuration and User Configuration parent nodes. These include:

- Software Settings, which is a location for independent software vendors (ISVs) to add further extensions. If no nodes have been added by ISVs, then Software Settings contains just the Software Installation extension included with Windows 2000.
- Windows Settings, which holds extensions provided by Microsoft.
- Administrative Templates, which shows namespace for registry-based policy settings. The Administrative Templates namespace is created by adding .adm files. You do this by right-clicking either of the Administrative Templates nodes, and then clicking "Add/Remove Templates."

Extensions to the Group Policy Snap-in

A Group Policy snap-in extension can extend the Group Policy namespace under the User Configuration or Computer Configuration nodes, or both. Most of the snap-in extensions extend both of these nodes, but frequently with different options. The Group Policy snap-in extensions included with Windows 2000 are listed below.

Administrative Templates These include registry-based Group Policy, which you use to mandate registry settings that govern the behavior and appearance of the desktop, including the operating system components and applications. There are over 450 of these settings available for you to configure, and you can add more using .adm files. To avoid undesirably persistent registry settings, any additional registry settings should be placed in \Software\Policies or \Software\Microsoft\Windows\CurrentVersion\Policies. See Group Policy Overview in this chapter about undesirably persistent registry settings.

Security Settings You use the Security Settings extension to set security options for computers and users within the scope of a Group Policy object. You can define local computer, domain, and network security settings.

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. You use Software Installation to install applications. The target computer needs to have the Windows 2000 operating system in place, as well as the client-side extension for Software Installation, Appmgmts.dll. To install Windows 2000 on a remote computer, use Remote Installation Services.

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 you like. Your options include Microsoft® Visual Basic® Scripting Edition (VBScript), JavaScript, Perl, and MS-DOS®-style batch files (.bat and .cmd).

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. Group Policy requires a genuine Windows 2000 client, not merely a client of Active Directory running on a previous version of Windows.

Internet Explorer Maintenance You use Internet Explorer Maintenance to administer and customize Microsoft® Internet Explorer on Windows 2000–based computers.

Folder Redirection You use Folder Redirection to redirect Windows 2000 special folders from their default user profile location to an alternate location on the network, where you can centrally manage them. Windows 2000 special folders include My Documents, Application Data, Desktop, and **Start** Menu.

Administrative Templates

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 Policy Group Policy object. Figure 22.3 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.

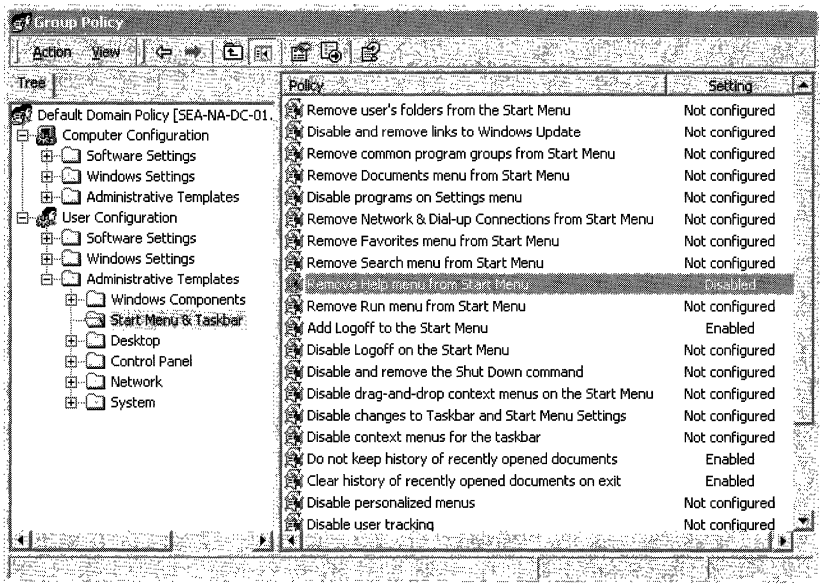


Figure 22.3 Administrative Template Group Policy Settings

Note The Windows NT 4.0 System Policy Editor uses files called administrative templates (.adm files) to determine which registry settings you can modify by presenting a namespace for those settings in the System Policy Editor. Windows 2000 .adm files have new features, such as Explain text. The Windows 2000 Resource Kit CD-ROM includes a searchable reference file, GP.chm, with details about the administrative templates settings included with Windows 2000 Server.

The Administrative Templates nodes of the Group Policy snap-in present registry-based Group Policy settings to the administrator. Administrative Templates govern a variety of behaviors for the Windows 2000 operating system and its components and applications. These settings are written to the HKEY_CURRENT_USER or HKEY_LOCAL_MACHINE portion of the registry database, as appropriate.

The .adm file is a Unicode text file that specifies a hierarchy of categories and subcategories that together define how the options are displayed through the Group Policy snap-in user interface. Unicode support for .adm files is new in Windows 2000. It also indicates the registry locations where you need to make changes if a particular selection is made, specifies any options or restrictions in values that are associated with the selection, and in some cases, specifies a default value to use if a selection is activated. By default, three .adm files, System.adm, Inetres.adm, and Conf.adm, which together contain more than 450 settings appropriate to Windows 2000 operating system clients, are installed in the Group Policy console. Inetres.adm contains settings for Internet Explorer, and System.adm has a wide variety of settings. There is also a Conf.adm, containing Microsoft® NetMeeting® settings, which is not loaded by default.

Note See the **Explain** tab of each Group Policy setting's **Properties** page for more details on the policy settings within the .adm file.

The Administrative Templates nodes of the Group Policy snap-in can be extended by using custom .adm files. For more information about creating .adm files, see Windows 2000 Help.

Undesirably Persistent Registry Settings

Windows NT 4.0 registry settings remain in effect until they are explicitly reversed. Windows 2000 registry settings, by contrast, are removed and rewritten each time policy changes. Be aware of this possibly undesirable behavior if you consider using Windows NT 4.0–type registry settings on Windows 2000–based computers.

For more information, see "Using Windows NT 4.0 Administrative Templates in the Windows 2000 Group Policy Console" later in this chapter.

Other Group Policy Extensions That Use the Registry

Both Remote Installation Services (RIS) and Disk Quotas use the registry. RIS has a node in the Group Policy console, but no client-side extension; that is, no .dll on the client computer. This is not surprising, because the client typically won't have an operating system. Disk Quotas is an example of a component with a client-side extension (Dskquota.dll), but no node in the Group Policy console.

Remote Installation Services

Remote Installation Services (RIS) is an optional component included in the Windows 2000 Server operating system. You can use the RIS extension of Group Policy to control which screen options (such as Automatic Setup, Custom Setup, and Restart Setup) are available to users during the client installation wizard.

When a client computer enabled with Preboot Execution Environment (PXE) remote-boot technology accesses the remote installation server to install the operating system, the Remote Installation Services server checks for Group Policy pertaining to remote installation options defined for the user. The Boot Information Negotiation Layer (BINL) service running on the RIS server performs this work. It impersonates the user who logs on to the RIS client-side pre-boot user interface, and evaluates the Group Policy objects to calculate the resulting policy. Based on the resulting policy, it determines which screens are sent to the pre-boot RIS client code for display to the user.

RIS policies are stored in the Sysvol folder at the following location: Policies\{*<GUID of GPO>*}\User\Microsoft\RemoteInstall\oscfilter.ini. For detailed information about Remote Installation Services, see "Remote OS Installation" in this book.

Security Settings

You can define a security configuration within a Group Policy object. A security configuration consists of settings applied to one or more security areas supported on Windows 2000 Professional or Windows 2000 Server. The specified security configuration is then applied to computers as part of Group Policy enforcement.

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, file, folder, and so on), and **Local Users and Groups** in **Computer Management**. You can continue to use existing tools to change specific settings whenever necessary.

The security areas that can be configured for computers include:

Account Policies These are computer security settings for password policy, lockout policy, and Kerberos policy in Windows 2000 domains.

Note These settings are only set at the domain level. If they are set at the organizational unit level, they are ignored.

Local Policies These include security settings for audit policy (Audit successful or failed logon attempts), user rights (who has network access to the computer) assignment, and security options (the ability to connect to a computer anonymously).

Event Log This controls settings such as size and retention method for the Application, Security, and System event logs. You can access these logs using Event Viewer.

Restricted Groups Allows you to control who needs to and who does not need to belong to security sensitive groups, as well as which other groups a security sensitive group needs to belong to. This allows administrators to enforce a membership policy regarding sensitive groups, such as Enterprise Administrators or Payroll. For example, it might be decided that only two users should be members of the Enterprise Administrators group. You can define the Enterprise Administrators group as a restricted group that contains only those two members. If a third user is added to the group (for example, to accomplish some task in an emergency situation), that user is automatically removed from the Enterprise Administrators group the next time policy is enforced. This mechanism can also be used to enforce group memberships on workstations in the domain (that is, enforcing that certain administrators from the domain are in the local Administrators groups on workstations).

System Services These control startup mode and access permissions for system services, such as who is allowed to stop and start the fax service.

Registry This is used to configure security settings for registry keys, including access control, audit, and ownership.

File System This is used to configure security settings for file-system objects, including access control, audit, and ownership.

Incremental Security Templates

Windows 2000 includes several incremental security templates. By default, these templates are stored in %systemroot%\Security\Templates. These predefined templates can be customized using the Security Templates MMC snap-in and can be imported into the Security Settings extension of the Group Policy snap-in.

These security templates are to be applied to Windows 2000–based computers that are configured with the Windows 2000 default security settings. They modify the default security settings incrementally, not cumulatively.

Note You should not apply these incremental templates to Windows 2000 systems that have been upgraded from Windows NT 4.0.

You should only apply these incremental templates onto Windows 2000 systems that have been clean-installed onto NTFS partitions. For NTFS computers that have been upgraded from Windows NT 4.0 or earlier, a Basic security template can be applied to configure the upgraded computer with the Windows 2000 default security settings. This is described in the following section. You cannot secure Windows 2000 systems that are installed on FAT file systems.

Security Configurations

Security configurations provide preconfigured sets of security settings that you can apply. You can configure them to Compatible, Secure, or High Secure Settings.

Compatible

The compatible template is provided in case you do not want to risk making end users into Power Users so that they can run older applications. This works on workstations and servers, and the template is called Compatws.inf.

Using this template, normal users—that is non-administrator and non-Power Users—cannot run most older applications on Windows 2000. This is because the default permissions granted to normal users are secure and most applications need to be rewritten to function properly in this environment. Applications that are Certified For Windows 2000 can be run successfully by a normal user. The Compatible configuration liberalizes the default permissions for the Users group so that older applications are more likely to run. Microsoft® Office 97 should run successfully when users are logged on as a User to a Windows 2000–based computer that has had the Compatible security template applied over the default settings. This is not considered a secure environment.

Secure

The Secure configuration provides increased security for areas of the operating system that are not secured by the default access control permissions. This works on workstations, servers, and domain controllers, and the templates are called Securews.inf and Securedc.inf

This configuration includes increased security settings for Account Policy, Auditing, and some well-known security-relevant registry keys. Access control lists are not modified by the secure configuration because this configuration assumes that default Windows 2000 security settings are in effect, and that users are members of the Users group. The Secure configuration removes all members of the Power Users group to enforce this assumption.

High Secure

The High Secure configuration provides increased security over the secure configuration. This works on workstations, servers, and domain controllers, and the templates are called Hisecws.inf and Hisecdc.inf.

The High Secure configuration requires that all network communications be digitally signed and encrypted. Because of these requirements, Windows 2000–based computers configured with the High Secure template might not be able to communicate with previous version clients such as Microsoft® Windows 98®–based (or earlier) computers that do not support the same network communication protections. The High Secure configuration also grants Power Users the same access to file system and registry keys as normal users. This allows users running Certified For Windows 2000 applications to offer inherent Power User capabilities such as the ability to create shares and change the system time without giving those same Power Users the lax permissions necessary to run noncertified applications. The High Secure template, when applied to servers, removes the Terminal Server user from all file system and registry ACLs, thus ensuring that users logging on to Terminal Server environments are also subject to the same secure access control policy as normal users.

Windows 2000 Default Security Templates

As noted earlier, the security templates described above (Compatible, Secure, and High Secure) incrementally modify the default Windows 2000 security settings that exist when Windows 2000 is clean-installed onto an NTFS partition. If you would like to apply these defaults to upgraded computers, or to clean-installed computers that have been subsequently modified, the following templates can be used:

- Basicwk.inf—applies default settings for Windows 2000 Professional–based computers for all areas except User Rights and Group Membership.
- Basicsv.inf—applies default settings for Windows 2000 Server–based computers for all areas except User Rights and Group Membership.
- Basicdc.inf—applies default settings for Windows 2000 Domain Controllers for all areas except User Rights and Group Membership. User Rights and Group Memberships are not modified by the basic templates because these templates are most often used for undoing file system or registry ACL changes, or to apply the default Windows 2000 ACLs to computers which have been upgraded from Windows NT 4.0. In these cases, customers usually want to maintain the existing User Rights and Group Memberships.

Software Installation

You can use the Software Installation snap-in to centrally manage software distribution in your organization. You can assign and publish software for groups of users, and you can assign software for groups of computers.

You assign applications to groups of users so that all users who require the applications have them on their desktops, without you or other technical personnel having to install the application on each desktop. When you assign an application to a group of users, you can be sure it is always available to them. When users log on to Windows 2000, the application shortcut appears on the **Start** menu, and the registry is updated with information about the application, including the location of the application package and the location of the source files for the installation. With this "advertisement" information about users' computers, the application is installed the first time users activate the application. When users select the application from the **Start** menu the first time, it installs and then opens. Users can remove assigned applications using Add/Remove Programs in Control Panel, but only for the duration of a logon session. The next time they start their computer, the application icon reappears.

You can also publish applications to groups of users, making the application available for users to install if they choose to do so. When you publish an application, no shortcuts to the application appear on users' desktops, and no local registry entries are made. That is, the application has no presence on the user's desktop. Information needed to published applications is stored in the Group Policy object.

To install a published application, users can use the Add/Remove Programs in Control Panel, which includes a list of all published applications that are available for them to use. Or, users can open a document file associated with a published application (for example, an .xls file to install Microsoft® Excel).

Scripts

Previously, the only native scripting language supported by the Windows operating system was the MS-DOS command language. In Windows 2000 this is expanded to include the Microsoft® ActiveX® scripting architecture. 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. You can use any WSH scripting tool including VBScript programming system and Microsoft® JScript® programming system to create scripts. Independent software vendors provide WSH support for other popular scripting languages. You can use Windows Script Host to run .vbs and .js scripts directly on the Windows desktop or command console, without the need to embed the scripts in an HTML document. MS-DOS-type batch files (with .bat and .cmd extensions) also function using Windows 2000.

The Scripts node located under Computer Configuration allows you to specify startup and shutdown scripts, and to specify logon and logoff scripts.

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

Table 22.1 shows some Group Policy settings that control how scripts are run.

Table 22.1 Group Policy Settings That Control Scripts

Location	Group Policy settings
Computer Configuration/Administrative Templates/System/Logon/	Run Logon Scripts Synchronously
	Run Startup Scripts Asynchronously
	Run Shutdown Scripts Visible
	Run Startup Scripts Visible
User Configuration/Administrative Templates/System/Logon/Logoff	Run Logon Scripts Synchronously
	Run Legacy Logon Scripts Hidden
	Run Logon Scripts Visible
	Run Logoff Scripts Visible

Scripts can cause the system to appear hung if an errant script (one that prompts the user for input) runs hidden. This occurs because the default wait time is 600 seconds. You can change this default using Group Policy. The setting is in the following location: Computer Configuration\Administrative Templates\System\Logon\Maximum wait time for Group Policy scripts.

If you run scripts in a minimized window, you can stop the scripts processing by normalizing the window.

You can also use the **Disable the Command Prompt** setting found under User Configuration\Administrative Templates\System. This prevents batch files from running (files with .cmd and .bat extensions). This optional setting should not be used for Terminal Services clients, because it prevents the Application Compatibility scripts from running. See Windows Explain text for details.

Legacy logon scripts are those scripts specified in the User object. You need to carefully consider these scripts if you have a mixed environment that includes Windows NT 4.0, Microsoft® Windows® 95, Windows 98, and Windows 2000 clients. The Windows 2000 and the Windows 98 clients properly run .vbs and .js scripts. However, to run .vbs and .js scripts on Windows NT 4.0 and Windows 95 clients, you need to embed the scripts in batch (.bat) files. The scripts continue to run in a normal window. Alternatively, you can install Windows Script Host to run unembedded scripts on Windows NT 4.0 and Windows 95 clients. The names of scripts and their command lines are stored in a .pol file in the form of registry keys and values.

For more information about Windows Script Host, see the Windows Script Technologies link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Folder Redirection

You use the Folder Redirection extension to redirect any of the following special folders in a user profile to an alternate location (such as a network share):

- Application Data
- Desktop
- My Documents
  - My Pictures
- Start Menu

You can redirect a user's My Documents folder to \\<*server name*>\<*share name*>\%username% and provide them with the following advantages:

- You can ensure that a user's documents are available when they roam from one computer to another, with or without roaming user profiles.
- When using roaming user profiles, you can reduce the time it takes to log on to and log off from the network. In Windows NT 4.0, the My Documents folder is part of the Roaming User Profile. This means that the My Documents folder and its contents are copied back and forth between the client computer and the server when the user logs on and logs off. By moving the My Documents folder out of the user profile, you can expedite user logon and logoff.
- You can store user data on the network rather than on the local computer, which allows you to manage and protect it.
- You can make a user's network-based folders available to them when they are disconnected from the corporate network by using Offline Folder technologies.

Analogous advantages apply to any redirected folder, not to only the My Documents folder. Redirecting to a DFS share provides a degree of safety in case of server failure. For more information about using offline folders, see "Remote OS Installation" in this book.

Extending the Group Policy Snap-in

Third-party application developers can extend the Group Policy snap-in to provide Group Policy specific to their applications. For this purpose, they can:

- Create an administrative template (.adm file). For more information, see Windows 2000 Help.
- Create a Group Policy MMC snap-in extension and provide the user interface for setting Group Policy specific to their application. For storing and distributing the policy, the following mechanisms are recommended:
  - The easiest is to use the API specific to the Group Policy MMC snap-in to write registry-based Group Policy to the Group Policy template. For more information about writing registry-based Group Policy to the Group Policy template, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

- Use the **GetFileSysPath** function to store non-registry-based (file-based) policy information in a Group Policy template subfolder. You should use the *<company name>\<application name>\<version>* naming convention for this folder. Then place the required files in that Group Policy template subfolder. On the client side, Winlogon calls the client-side extension for the tool. This in turn processes the information stored in the directory in the Group Policy template. The application developer must use this mechanism appropriately. By storing the data in a Group Policy template subfolder, the application capitalizes on the built-in mechanisms of Group Policy (the Group Policy template and Winlogon) for applying special non-registry-based policy. For more information about the GetFile SysPath function, see the SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.
- You can store data in the Group Policy container. It is strongly recommended to use either the Group Policy container or the Group Policy template, not both.

Client-side Extensions to Group Policy

Most of the Group Policy snap-in extensions also include client-side extensions. These extensions are dynamic-link libraries (DLLs) that are responsible for implementing Group Policy at the client computers.

There are some client-side extensions, such as Disk Quotas, that don't correspond to a snap-in. For more information about the client-side extensions, see "Computer Policy for Client-Side Extensions" later in this chapter.

For information about Microsoft Management Console, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Group Policy Storage

There are two kinds of Group Policy objects: local and non-local. Only one local Group Policy object is stored on each Windows 2000–based computer. Non-local Group Policy objects are Active Directory–based.

Non-Local, 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), which is used to keep them synchronized.

Figure 22.4 shows data storage locations which are typically contained in a Group Policy object.

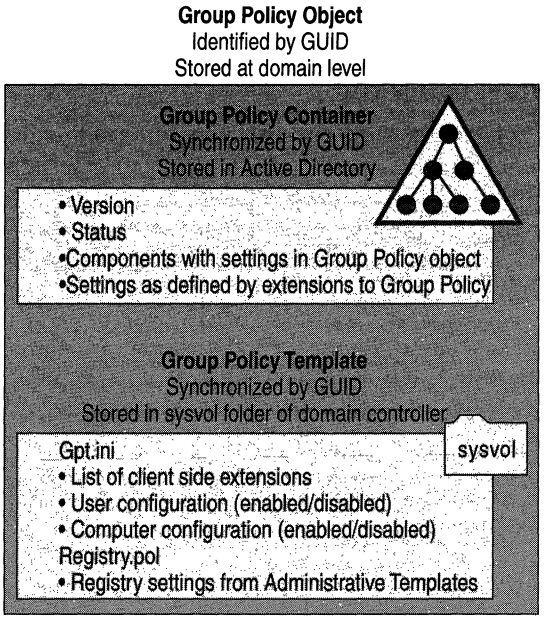


Figure 22.4 Group Policy Storage Model

Note This information about storage implementation is purely informational. As an administrator you don't interact with the Group Policy template and the Group Policy container directly, but through Active Directory tools such as the Group Policy console.

Group Policy Container

The Group Policy container is an Active Directory storage area for Group Policy object properties; it includes both computer and user Group Policy information. The Group Policy container has the following properties:

- Version information. This makes sure that the information is synchronized with the Group Policy template information.
- Status information. This indicates whether the Group Policy object 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:

For example, the Group Policy container stores information used by the Software Installation snap-in to describe the status of the software available for installation. This data repository contains data for all applications, interfaces, and APIs that provide for application publishing and assigning.

Group Policy Template

Group Policy objects also store Group Policy information in a folder structure called the Group Policy template that is located in the System Volume folder of domain controllers (Sysvol) in the \Policies subfolder. The Group Policy template is the container where Administrative Template–based policy settings, Security Settings, applications available for Software Installation, and script files are stored.

When you modify a Group Policy object, the directory name given to the Group Policy template is the GUID of the Group Policy object that you modify. For example, a Group Policy template folder might be named as shown in the following example:

%systemroot%\sysvol\SYSVOL\www.Reskit.com\Policies\{47636445-af79-11d0-91fe-080036644603}

A Group Policy snap-in can store data outside the Group Policy object; however, this requires that at least a link to the Group Policy object be stored either in a Group Policy container (Active Directory data store) or in a Group Policy template (file-type data stored on the Sysvol folder).

Note Group Policy is not backed up separately from the rest of Active Directory. To back up Active Directory, you need to be a member of Backup Operators group. The required privilege is **Backup Files and Directories**. For instructions on backing up Active Directory, see "Active Directory Backup and Restore" in this book.

Gpt.ini File

At the root of each Group Policy template folder is a file called Gpt.ini. 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.
- Version number of the Group Policy snap-in extension that created the Group Policy object.

Local Group Policy Objects

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

If Read permission is withdrawn from the Local Administrator group, Group Policy does not apply. This is a convenient way to exempt Local Administrators from a group Policy object even though they have the Apply Group Policy permissions set to **Allow**.

The local Group Policy object Gpt.ini file contains the following information:

GPCUserExtensionNames This Includes a list of GUIDs that tells the client side engine which client-side extensions have User data in the Group Policy object. The format is: [{*<GUID of client-side extension>*}{*<GUID of MMC extension>*}{*<GUID of second MMC extension if appropriate>*}][repeat first section as appropriate].

GPCMachineExtensionNames This includes a list of GUIDs that tells the client-side engine which client-side extensions have Computer data in the Group Policy object.

Options This refers to Group Policy object options such as User portion disabled or Computer portion disabled.

GPCFunctionalityVersion This is the version number of the Group Policy extension tool that created the Group Policy object.

Group Policy Template Subfolders

The Group Policy template folder contains a tree of subfolders. The number of subfolders that are present in the tree depends on the Group Policy object; however, at least two subfolders are always present. They are Machine and User. The following is a description of each folder:

Machine Includes a Registry.pol file that contains the registry settings that are applied to computers. When a computer initializes, this Registry.pol file is downloaded and applied to the **HKEY_LOCAL_MACHINE** portion of the registry.

User Includes a Registry.pol file that contains the registry settings that are applied to users. When a user logs on to a computer, this Registry.pol file is downloaded and applied to the **HKEY_CURRENT_USER** portion of the registry.

The Group Policy template folder also includes a Gpt.ini file which contains version information. For Active Directory–based Group Policy objects, this file contains the version number of the Group Policy object in a line of this form:

```
Version=<version number>
```

The version number is the decimal representation of an eight-digit hexadecimal number (a DWORD). The four least significant digits represent the Computer Settings version number, and the four most significant digits represent the User Settings version number. For example, if you see

```
Version=65539
```

then the Computer Settings version is 3, and the User Settings version is 1, because 65539 converted to hexadecimal is 0X00010003.

The Group Policy template folder can also include the following subfolders:

Adm Contains all of the .adm files for this Group Policy object.

Machine\Scripts\Shutdown Contains scripts that run when the computer shuts down.

Machine\Scripts\Startup Contains scripts that run when the computer starts.

Machine\Applications The contents depends on what applications are computer-assigned with a given Group Policy object.

Machine\Microsoft\Windows NT\Secedit Contains GptTmpl.inf, the default security configuration settings for a Windows 2000 domain controller.

User\Applications Contains the advertisement files (.aas files) used by the Windows installer.

User\Documents & Settings Contains Fdeploy.ini, which holds information about the Folder Redirection status of the current user's special folders.

User\Microsoft\RemoteInstall Contains OSCfilter.ini, which holds user choices for operating system installation through Remote Installation Services.

User\Microsoft\IEAK Contains settings for the Internet Explorer Maintenance Snap-in.

User\Scripts\Logoff Contains scripts that are run when the user logs off the computer.

User\Scripts\Logon Scripts to be run when the user logs on to the computer.

Note The User and Machine folders are created during installation, and other folders are created as needed when policy is set.

Registry.pol Files

The Administrative Templates extension of Group Policy saves information in the Group Policy template in text files with the name Registry.pol. These files contain the customized registry settings that are applied to the Machine or User portion of the registry which you specify using the Group Policy snap-in. The Windows 2000 Registry.pol file is analogous to the Windows 95 or Windows 98 Config.pol file and the Windows NT 4.0 NT Config.pol file.

Two Registry.pol files are created and stored in the Group Policy template, one for Computer Configuration, which is stored in the \Machine subdirectory, and one for User Configuration, which is stored in the \User subdirectory.

Note The format of the .pol files in the Group Policy template differs from that of the .pol files in previous versions of Windows.

The .pol files created by Windows NT 4.0 and Windows 95 can be applied only to the operating system on which they were created. The .pol file produced by the Windows NT 4.0 System Policy Editor was a binary file, whereas the Registry.pol file produced by Administrative Templates node of the Group Policy snap-in is a text file with embedded binary strings.

To view .pol files without applying them to the registry, use the Regview.exe tool located on the *Microsoft® Windows® 2000 Server Resource Kit* companion CD.

For additional information about Registry.pol files, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Group Policy Object Links

Group Policy objects are actually applied to a site, domain, or organizational unit by using a link. A non-local Group Policy object that is not linked to a site, domain, or organizational unit has no effect on any user or computer anywhere, even in the storage domain.

No Override as Compared to Block Policy Inheritance

You can set **No Override** on a specific Group Policy object link so that Group Policy objects linked at a lower-level of Active Directory—closer to the recipient user or computer account—cannot override that policy. If you do this, Group Policy objects 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, then you need to prioritize them. Links higher in the list have priority on all Configured (that is, **Enabled** or **Disabled**) settings.

If you have linked a specific Group Policy object to a domain, and set the Group Policy object link to **No Override**, then the configured Group Policy settings that the Group Policy object contains apply to all organizational units under that domain. Group Policy objects linked to organizational units cannot override that domain-linked Group Policy object.You can also block inheritance of Group Policy from above in Active Directory. This is done by checking **Block Policy inheritance** on the **Group Policy** tab of the **Properties** sheet of the domain or organizational unit. This option does not exist for a site.

Some important facts about **No Override** and Block Policy are listed below:

- **No Override** is set on a link, not on a site, domain, organizational unit, or Group Policy object.
- **Block Policy Inheritance** is set on a domain or organizational unit, and therefore applies to all Group Policy objects linked at that level or higher in Active Directory which can be overridden.
- **No Override** takes precedence over **Block Policy Inheritance** if the two are in conflict.

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 menu.

Multiple Group Policy Objects

Each non-local Group Policy object is stored in a specific domain, which we can call the storage domain. The storage domain should not be confused with a domain to which the Group Policy object is linked.

You can link multiple Group Policy objects to a single site, domain, or organizational unit. However, many sites, domains, and organizational units, can all obtain policy from a single Group Policy object by way of links to it, regardless of which domain the Group Policy object is stored in.

A Group Policy object can be linked to its storage domain, and if network connections to other domains are slow this is desirable. However, it doesn't have to be linked to its storage domain.

It is also possible to link a single Group Policy object more than once to a single site, domain, or organizational unit, though it is seldom useful to do so.

For performance reasons, it is best to avoid linking to a Group Policy object in a different domain.

Cross-Domain Editing of a Group Policy Object

Before you can edit a Group Policy object, the following two conditions must be met:

- You have Read/write permissions (or Full Control) of the Group Policy object.
- You are either logged on to the storage domain of the Group Policy object, or logged on to a domain that is trusted by the storage domain.

Using Security Groups to Filter and Delegate Group Policy

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

Filtering the Scope of a Group Policy Object

You can refine which groups of computers and users a particular Group Policy object influences by using Windows 2000 security groups. To do this, use the **Security** tab on the **Properties** page of the Group Policy object.

Filtering affects the Group Policy object as a whole. That is, you cannot use security groups to apply (or prevent from applying) only some of the settings in a Group Policy object. However, this is not true in the cases of Folder Redirection and Software Installation, which have further ACLs set at the Group Policy object level to further refine behavior based on security group membership.

Setting Security Permissions for Receiving Group Policy

A discretionary access control list (DACL) is a list of permissions (such as Read, Apply Group Policy, and Full Control) on a Group Policy object or other object. You use the DACL on a Group Policy object to allow or deny access to the Group Policy object by users and computers according to their membership in security groups.

To use the **Security** tab on the **Properties** page for a Group Policy object, right-click the root node of the Group Policy snap-in, click **Properties**, and then click **Security**.

An alternative is to open the **Properties** page of a given site, domain, or organizational unit, then select the **Group Policy** tab, right-click a **Group Policy object** in the Group Policy object list, select **Properties**, and then click the **Security** tab. Group Policy objects that you can access this way are linked to the site, domain, or organizational unit.

You can specify which groups of users and computers have Apply Group Policy access control entries (ACEs) set to enable access to the Group Policy object. ACEs are permission entries within a discretionary access control list (DACL). Groups that have Apply Group Policy and Read access to the Group Policy object receive the configured Group Policy settings contained in it if they are subject to the Group Policy object through Active Directory. By default, authenticated users have both Apply Group Policy and Read permissions, but not Write or Full Control. This means that by default, users cannot modify the information in the Group Policy object. By default, domain administrators, enterprise administrators, and the local system have Full Control, without Apply Group Policy. By default, administrators are also authenticated users, which means that they also have the Apply Group Policy attribute set. For more information, see "Editing Group Policy Objects" later in this chapter.

Note It is recommended that you remove Read permission from groups whose members don't need to receive policy and contain users who are not administrators because this data can be viewed by any users with Read permission. Group Policy processes faster if both the Read and Apply Group Policy settings are disabled when the Apply Group Policy setting is not needed. In addition, Group Policy fails if a user has Read access to more than 1,000 Group Policy objects stored in one domain. For more information about Group Policy failing when more than 1,000 Group Policy objects are present, see "Troubleshooting Change and Configuration Management" in this book.

Network administrators (members of the Enterprise Administrators or Domain Administrators group) can also use the **Security** tab on the Group Policy object **Properties** page to determine which administrator groups can modify policy settings in Group Policy objects. To do this, the network administrator can define groups of administrators (for example, marketing administrators), and then provide them with Read/write access to selected Group Policy objects. This allows the network administrator to delegate control of Group Policy objects.

Having full control of a Group Policy object does not enable you to link it to a site, domain, or organizational unit. However, you can grant that ability using the Delegation of Control Wizard.

Delegating Control of Group Policy

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.

Non-local Group Policy, like all Active Directory–based administrative tools, requires a Windows 2000 domain controller. Group Policy, like most other Windows 2000 administrative tools, is hosted in MMC consoles. The rights to create, configure, and use MMC consoles, therefore, have policy implications. You can control these rights through Group Policy under

<Group Policy object name>/User Configuration/Administrative Templates/Windows Components/Microsoft Management Console/

and its subfolders.

Table 22.2 lists the security permission settings for a Group Policy object.

Table 22.2 Security Permission Settings For A Group Policy Object.

Groups or Users	Security permission
Authenticated User	Read with Apply Group Policy ACE
Domain Administrators Enterprise Administrators Creator Owner Local System	Full Control without Apply Group Policy ACE

Note By default, if you are an administrator, you are also an authenticated user, which means that you have the **Apply Group Policy** attribute set. For more information, see "Editing Group Policy Objects" later in this chapter.

To administer Group Policy, you need to log on to a local or remote domain controller, which requires special permission. If you are a domain administrator or you are the built- in administrator on a domain controller, you have this permission.

Non-administrators can log on to a domain controller only if they have Log On Locally permission. This is part of the Default Domain Controllers Group Policy object, linked to the Domain Controllers organizational unit in Active Directory Users and Computers. The setting is found under Computer Configuration/Windows Settings/Security Settings/Local Policies/User Rights Assignment/Log on locally.

It is recommended that you create a security group containing those users who should be able to log on locally to the domain controller, and add them to the list of groups shown on the **Log On Locally** form. Remember that computer policy for the domain controller must refresh before the new permissions take effect.

Managing Group Policy Links for a Site, Domain, or Organizational Unit

The **Group Policy** tab in the site, domain, or organizational unit's **Properties** page allows you to specify which Group Policy objects are linked to this site, domain, or organizational unit. This property page stores the user's choices in two Active Directory properties called **gPLink** and **gPOptions**. The **gPLink** property contains the prioritized list of Group Policy object links and the **gPOptions** property contains the **Block Policy Inheritance** policy setting for domains or organizational units. The **Block Policy Inheritance** policy setting is not available for sites.

Figure 22.5 illustrates the creation of a new Group Policy object from within Active Directory Users and Computers. It has the default name **New Group Policy Object**, which you can change to something more descriptive. It will be stored in the noam.reskit.com domain, and it will automatically be linked to the Manufacturing organizational unit.

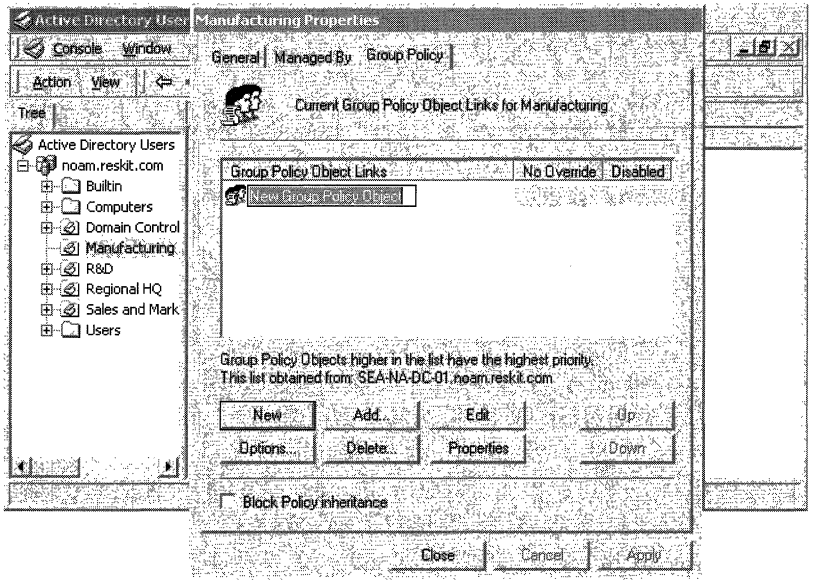


Figure 22.5 Manufacturing Properties of New Group Policy Object

Active Directory supports security settings on individual properties. Thus, a non-administrator can be given Read and Write access to specific properties. If non-administrators have Read and Write access to the **gPLink** and **gPOptions** properties, they can manage the list of Group Policy objects linked to that site, domain, or organizational unit. To give a user Read and Write access to these properties, use the Delegation of Control Wizard and select the **Manage Group Policy links** predefined task.

Creating Group Policy Objects

By default, only Domain Administrators, Enterprise Administrators, Group Policy Creator Owners, and the operating system can create new Group Policy objects. If the domain administrator wants a non-administrator or group to be able to create Group Policy objects, that user or group can be added to the Group Policy Creator Owners security group. When a non-administrator who is a member of the Group Policy Creator Owners group creates a Group Policy object, that user becomes the Creator Owner of the Group Policy object. Then the user can edit the Group Policy object. Being a member of the Group Policy Creator Owners group gives the non-administrator full control of only those Group Policy objects that the user creates or those explicitly delegated to that user. It does not give the user full control of any other Group Policy objects, and does not allow the user to link Group Policy objects to sites, domains, or organizational units.

Editing Group Policy Objects

By default, Group Policy objects give Domain Administrators, Enterprise Administrators, the operating system, and the Group Policy object Creator Owner full control without the **Apply Group Policy** attribute. They can edit the Group Policy object. But even if members of those groups have accounts in Active Directory sites, domains, or organizational units linked to the Group Policy object, the policy settings contained in that Group Policy object do not apply to them unless both of the following two conditions pertain:

- They have **Apply Group Policy** set to **Allow** as members of another security group.
- They don't have **Apply Group Policy** set to **Deny** as members of any security group.

By default, Authenticated Users have Read access to the Group Policy object with the **Apply Group Policy** attribute set.

Note By default, if you are an administrator, you are also an authenticated user, which means that you have the **Apply Group Policy** attribute set. If this is not what you intend, you have two choices:

- Remove authenticated users from the list, and add a security group with the **Apply Group Policy** attribute set to **Allow**. This new group should contain all the users who this Group Policy is intended to affect.
- Set the **Apply Group Policy** attribute to **Deny** for the Domain and Enterprise Administrators, and possibly the Creator Owner groups. This will prevent the Group Policy object from being applied to members of those groups. Remember that an ACE set to **Deny** always takes precedence over **Allow**. If a given user is a member of another group that is set to explicitly **Allow** the **Apply Group Policy** attribute for this Group Policy object, it will still be denied.

When a non-administrator creates a Group Policy object, he or she becomes the Creator Owner of the Group Policy object. When an administrator creates a Group Policy object, the Domain Administrators group becomes the Creator Owner of the Group Policy object.

To edit a Group Policy object, the user must have both Read and Write access to the Group Policy object. A Group Policy object cannot be opened in read-only mode. In other words, if you can open the Group Policy snap-in, you can edit the Group Policy object that appears in the namespace. Moreover, the changes occur during the edit. There is no "Save" or "Activate" step. As a precaution, you might want to unlink a Group Policy object from any site, domain, or organizational unit during the edit. Or you can leave it linked, but disable both the User and Computer nodes.

To edit a Group Policy object, the user must be one of the following:

- An administrator.
- A Creator Owner.
- A user with delegated access to the Group Policy object. That is, an administrator or the Creator Owner must have delegated access to this user by using the **Security** tab in the Group Policy object **Properties** page, and adding them to the Group Policy Creator Owners list.

Examples of Group Policy Delegation

Below are three examples of Group Policy delegation.

Example 1

In this example, control of an organizational unit is delegated to a non-administrative user so that a user can link existing Group Policy objects to the organizational unit but not create new Group Policy objects.

Throughout this example, a security group can take the place of the individual user.

1. In **Active Directory Users and Computers** snap-in, right-click the **Organizational Unit** that you want to delegate, and select **Delegate Control**.
2. In the **Delegation of Control Wizard**, click **Next** to go past the introduction page. You will be prompted for the names of the users and groups to which you want to delegate control.
4. Select a previously defined user, and click **Next**.
5. In the list of **Predefined Tasks**, select **Manage Group Policy links**, and then click **Next**.
6. Click **Finish** to complete the changes.

The user who you selected can add and delete Group Policy object links for the organizational unit whose control you delegated, to any Group Policy objects to which they have Read access.

If the Group Policy object is stored in another domain, the user's domain must be trusted by the storage domain.

Example 2

In this example, a user is given permission to create new Group Policy objects.

This permission is often useful in combination with the right to create links, as described in the previous example. To allow for creation of new Group Policy objects, you need to add the user to the **Group Policy Creator Owners** administrators group.

1. In **Active Directory Users and Computers**, navigate to the **Users** container in the root of the domain.
2. Double-click **Group Policy Creator Owners**.
3. In the **Properties** page, select the **Members** tab.
4. Click **Add**, and then add the user selected above to the security group.

The user can create new Group Policy objects, and the specific user who created each object becomes the Creator Owner of that Group Policy object.

You create Group Policy objects from within Active Directory Users and Computers by right-clicking the domain or organizational unit, clicking **Properties**, then the **Group Policy** tab, and then **New.** These objects are, by default, linked to the domain or organizational unit that has focus when they are created. Thus, a user with delegated rights, or you as an administrator, or any user who carries out the task of creating a Group Policy object in this way must have permission not only to create the Group Policy object, but also permission to link it to the domain or organizational unit. Otherwise the **New** button on the **Properties** sheet for the domain or organizational unit is shaded.

Example 3

In this example, control of a Group Policy object is delegated to a non-administrator user or group of users.

1. Open a **Group Policy object** in the **Group Policy** snap-in.
2. Right-click the root node, select **Properties**, and click **Security**.
3. Click **Add** to add the group of users or user, and then click **Full control**.
4. Clear the **Apply Group Policy** option or leave it checked depending on your purpose.

   This example shows how to delegate control of the Group Policy object. For this, you do not need the Apply Group Policy permission. If you clear the **Apply Group Policy** check box, the **Full Control** check box is also cleared, but the user still has Read/Write access to the Group Policy object.
5. Click **OK** to save the changes.

The user or group of users can edit the Group Policy object.

Creating MMC Consoles to Delegate Group Policy

You can delegate Group Policy administrative rights by creating and saving Group Policy MMC consoles, and then specifying which users and groups have access permissions to the Group Policy object, site, domain, or organizational unit. You define permissions for a Group Policy object by using the **Security** tab on the **Properties** page of the Group Policy object. These permissions grant or deny access to a Group Policy object to specified groups.

This type of delegation is augmented by the Group Policy settings available for MMC. Several settings are available in the Administrative Templates node, under Windows Components, Microsoft Management Console. These settings enable you to establish which MMC snap-ins the affected user can or cannot run. You can specify this as inclusive, which only allows a set of snap-ins to run, or as exclusive, which does not allow a set of snap-ins to run. See the **Explain** tab text of the individual policy setting for more information.

You can create custom consoles (.msc files) for Group Policy as for any other snap-in.

You can create and save custom Group Policy consoles that include only a subset of the Group Policy snap-in extensions. For example, you can create a custom Group Policy console that includes only the Security Settings extension. This allows you to define Group Policy settings in a modular fashion.

You can also create custom consoles that contain instances of Group Policy focused on different Group Policy objects or that contain snap-ins unrelated to Group Policy.

The computer on which the console runs must hold any DLLs used by the snap-ins. If the computer is a domain controller, the DLLs are probably present already. If not, their presence on the Windows 2000 member server or Windows 2000 Professional–based computer can be ensured by assigning or publishing the Windows 2000 Administration Tools. The package is called Adminpak.msi, and you can find it on the Windows 2000 Server companion CD.

There are several ways to start the Group Policy snap-in depending on your purpose. See Windows 2000 Help for more information. It is recommended that you delegate Group Policy using Custom Consoles. To do this, you should start Group Policy as a stand-alone snap-in:

▶ **To start Group Policy as a stand-alone snap-in**

1. Click **Start**, click **Run**, type **MMC**, and then click **Enter**.
2. In the MMC window, on the **Console** menu, click **Add/Remove Snap-in**.
3. On the **Standalone** tab, click **Add**.
4. In the **Add Snap-in** dialog box, click **Group Policy**, and then click **Add.**
5. In the **Select Group Policy object** dialog box, click **Browse** to find the Group Policy object you want to manage.
6. Click **Extensions**, and then select the extension snap-ins you want to use.
7. Click **Finish**.
8. Click **OK**. The Group Policy snap-in opens with focus on the Group Policy object you specified.
9. After you specify the policy settings you want to use, click **Save As** on the **Console** menu to save your settings in an .msc file.

To set access permissions, use the **Security** tab on the **Properties** page of the selected Group Policy object. These permissions allow or deny access to the Group Policy object by specified groups.

There are dozens of Group Policy settings that allow or deny access to various snap-ins and snap-in extensions. Check the following folders for settings that might be relevant for your organization:

- User Configuration/Administrative Templates/Microsoft Management Console
- User Configuration/Administrative Templates/Microsoft Management Console/Restricted/Permitted snap-ins
- User Configuration/Administrative Templates/Microsoft Management Console/Restricted/Permitted snap-ins/Extension Snap-ins
- User Configuration/Administrative Templates/Microsoft Management Console/Restricted/Permitted snap-ins/Group Policy

Group Policy Processing

Group Policy is processed in the following order:

The local Group Policy object. This is the only source of Group Policy for stand-alone computers or computers in workgroups. The local Group Policy object is always processed.

Active Directory linked Group Policy objects. Site first, domain next, and organizational unit last, including any nested organizational units, from parent to child. At each site, domain, or organizational unit, one, many, or no Group Policy objects can be linked. If more than one link is present, those links can be prioritized.

Note The **Block policy inheritance** or **No Override** options can affect the presence or absence of Group Policy objects in the list of Group Policy objects to be processed, but cannot change their order. The blockade occurs at the domain or organizational unit level, thus removing all non-local Group Policy objects that would otherwise be processed earlier, except those set to **No Override**. The local Group Policy object cannot be blocked. The **No Override** setting for Group Policy is an attribute of a link, and therefore applies to a *particular* Group Policy object, and only at the *particular* site, domain, or organizational unit to which it is linked.

Computer policy is processed at startup and then user policy is processed when the user logs on. Although computer policy is applied before user policy, if user and computer policy settings specify different behavior, the computer policy will generally prevail. This is not enforced by the Group Policy infrastructure, but is rather a convention that is followed by the operating system and by applications that exploit Group Policy unless there are specific reasons that the convention is not appropriate for a given policy setting.

Note There are policy processing issues that arise if you use Windows NT 4.0 and migrate to a Windows 2000 environment. For more information, see "Migration Issues Pertaining to Group Policy" later in this chapter.

Most Group Policy settings are implemented at the client computer by DLLs on the client. These DLLs are called client-side extensions. Remote Installation Services is an exception. RIS has no client-side extension because it is used to install an operating system remotely, and a DLL is useless without an operating system.

For each client-side extension, the Group Policy object processing order is obtained from a list of Group Policy objects, which is obtained from the **GetGPOList** Microsoft® Win32® function. Each client-side extension processes the resulting list of Group Policy objects.

In most cases policy settings specified in the Computer Configuration node have precedence over the same setting if one exists in the User Configuration node. There are a few exceptions and their behavior is set forth in the Explain text for those settings. An example is Administrative Templates\Windows Components\Windows Installer\Always install with elevated privileges, which requires the setting in both Computer and User Configuration to be enabled or it is not activated. See the Explain text for that policy setting for details.

Group Policy affects only users and computers contained in sites, domains, and organizational units. Specifically, Group Policy objects are not applied to Security Groups.

Synchronous and Asynchronous Processing

Asynchronous refers to processes that do not depend on each other's outcome, and can therefore occur on different threads simultaneously. The opposite is synchronous. Synchronous processes wait for one to complete before the next begins. For those Group Policy settings for which both types of processes are available as options, you choose between the faster asynchronous or the safer, more predictable synchronous processing.

By default, the processing of Group Policy is synchronous. Computer policy is completed before the CTRL+ALT+DEL dialog box is presented, and user policy is completed before the shell is active and available for the user to interact with it.

Note You can change this default behavior by using a policy setting for each so that processing is asynchronous. This is not recommended unless there are compelling performance reasons. To provide the most reliable operation, leave the processing as synchronous.

Group Policy for computers is applied at computer startup. For users, Group Policy is applied when they log on.

Periodic Refresh Processing

You can specify that Group Policy be processed periodically. By default, this is done every 90 minutes with a randomized offset of up to 30 minutes. You can change these default values by using a Group Policy setting in Administrative Templates. Setting the value to zero minutes causes the refresh rate to be set to seven seconds.

If you want to change this setting, edit the Default Domain Controllers Group Policy object. It is linked to the Domain Controllers organizational unit. The setting is located under Computer Configuration/Administrative Templates/System/Group Policy/Group Policy Refresh Interval for Computers.

Caution Short refresh intervals are intolerable in a production environment. Every policy refresh causes the Windows shell to be refreshed, which in turn causes all open context menus to close, a brief flicker of the screen, and so on. Such intervals are intended to be used only in test or demonstration scenarios.

The default period is every five minutes for domain controllers. There is a setting for this under Computer Configuration/Administrative Templates/System/Group Policy/Group Policy Refresh Interval for Domain Controllers.

Software Installation and Folder Redirection processing occurs only during computer startup or when the user logs on, not during the periodic refresh. For these extensions, periodic processing is inappropriate. For example, in the case of Software Installation, if an application is no longer assigned, it can be removed automatically, if you have used Group Policy to set it up that way. If a user is using the application while Group Policy tries to uninstall it, or if an assigned application upgrade takes place while a user is using it, the user will encounter an error.

Optional Processing of Group Policy Even If It Has Not Changed

To achieve the highest level of policy settings security, activate the **Process Even If The Group Policy Objects Have Not Changed** policy for each of the Group Policy client-side extensions that require it. These policy settings are located in the Computer Configuration node, under Administrative Templates, System, Group Policy. Each client-side extension has a policy setting for controlling the policy processing. By default, each Group Policy client-side extension updates its policy settings only when they have changed. Choosing this option ensures that the selected settings are applied at every logon session to Active Directory, but forgoes the performance optimization achieved by skipping the application of policy settings when they have not changed. For information about Windows 2000 security, see the chapters under "Distributed Security" in this book.

Group Policy and Network Bandwidth

When Group Policy detects a slow link, it sets a flag to indicate this to client-side extensions.

The default settings for whether policy is applied over a slow link are shown below. ON indicates that processing occurs even if the link is judged to be slow.

- Security Settings—ON (and cannot be turned off)
- Administrative Templates—ON (and cannot be turned off)
- Software Installation—OFF
- Scripts—OFF
- Folder Redirection—OFF
- Internet Explorer Maintenance—OFF

For all but the Administrative Templates and Security snap-ins, a policy is provided for toggling the settings.

Setting Policy for Slow-Link Definition

You can use Group Policy to set the definition of a slow link for computers and users, and for user profiles.

Note Windows 2000 adds an IP algorithm to ping the server, whereas Windows NT 4.0 just measures the file system performance.

The following algorithm is used to determine whether the link should be considered slow:

Ping the server with 0 bytes of data and time the number of milliseconds. Call this value time#1. If it is less than 10 milliseconds, then assume it is a fast link, and exit.

Ping the server with an uncompressible 2 kilobytes (KB) of data, and time the number of milliseconds. Call this value time#2. The algorithm uses an already compressed .jpg file. If it used a compressible file, the modem would compress it and make the network appear faster than it is.

DELTA = time#2 - time#1. This removes the overhead of session setup. DELTA is the time in milliseconds to move 2 KB.

DELTA is measured three times, and the average of the three values of DELTA obtained is called AVG.

Then the connection speed Z, measured in kilobits per second (Kbps), is:

Z = 32000/AVG.

The correctness of this formula is more apparent with the units in place:

(Z kilobits / second) = 2 * (2 KB) * (8 bits/ byte) * (1000 milliseconds / second) / (AVG milliseconds).

2 KB of data have moved through each modem, Ethernet card or other device in the loop once in each direction, so this equation for calculating the one-way bandwidth has a leading factor of 2 on the right side.

Note The speed Z used here is the average of the upload and download speeds. In most cases, this average is the same as the download speed itself. However, in some cases the upload and download speeds are different enough that you should take this into account. An example of this is Asymmetric Digital Subscriber Line (ADSL). Using ADSL you might have upload speeds of 128 Kbps and download speeds of 768 Kbps.

Z is compared with 500 Kbps (or an alternative threshold of your choice if you change from the default Group Policy setting of 500 Kbps).

If Z is less than 500 Kbps the connection is considered slow, otherwise it is considered fast.

You can set the default value of 500 Kbps in the Group Policy console under *<Group Policy object name>*/Computer Configuration/Administrative Templates/System/Group Policy/Group Policy slow link detection.

To specify policy settings for Group Policy slow link detection for computers, you use the Computer Configuration\Administrative Templates\System\Group Policy node. To set this policy for users, you use the User Configuration\Administrative Templates\System\Group Policy node.

For User Profiles, the Slow network connection time-out for user profiles policy is located in the Computer Configuration\Administrative Templates\System\Logon node. The user profile code first tries to contact (or ping) the server. If the server does not have IP support, it falls back to measuring the file system's performance. You specify a threshold connection speed in kilobits per seconds, and a threshold transit time in milliseconds, when configuring this policy setting.

Registry Reads

Group Policy snap-in extensions can temporarily claim (or lock) a mutex (mutual exclusive) for policy, and then release that mutex. APIs exist to allow a client-side extension to claim the mutex, read the required values, then release the critical section. If it is not released in 10 minutes, the client-side extension is forced to release it. This ensures that the background refresh of Group Policy does not occur during the read process.

For more information about the Windows 2000 Group Policy APIs, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Specifying a Domain Controller for Setting Group Policy

Two methods are available to set domain controller options for Group Policy. One method is to use the Group Policy snap-in user interface, where the user can set domain controller options by using the DC Options dialog box, as described next. The other method allows the primary domain administrators to set domain controller options by using a policy in the Administrative Templates node, as described in Specifying Policy for Domain Controller Options listed later.

The Group Policy snap-in **View** menu contains an entry called DC Options, which opens the Options for the domain controller selection dialog box, where you will be able to specify a preference for a domain controller to use for editing Group Policy. Figure 22.6 shows the **Options for domain controller selection** dialog box.

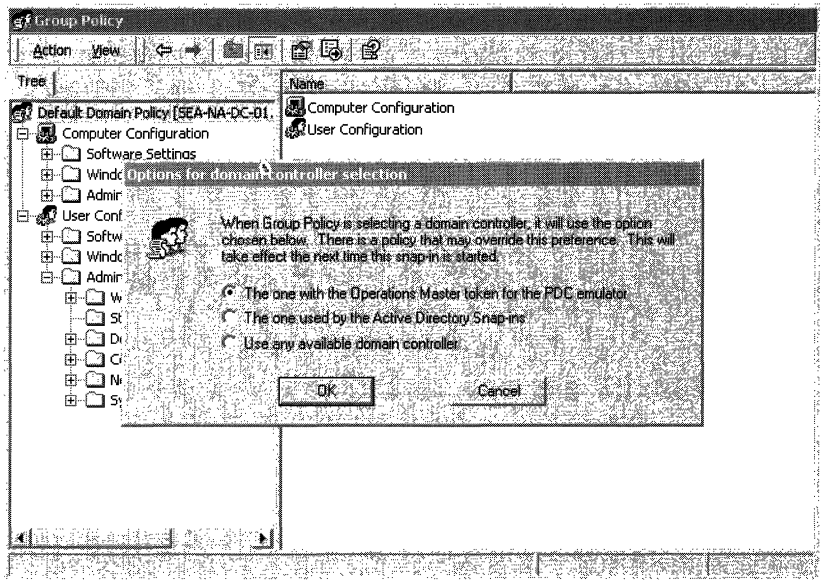


Figure 22.6 Options for Domain Controller Selection Dialog Box

In the **Options for domain controller selection** dialog box shown in Figure 22.6, you can choose the following options:

The one with the Operations Master token for the PDC emulator. This is the default and preferred option. Using this option helps ensure that no data loss occurs. This forces the Group Policy snap-in to use the same domain controller. Data loss can occur if two administrators are working on changes to the same Group Policy object on different domain controllers within the replication cycle. Group Policy writes data to the Group Policy object for each change. If two administrators are editing a Group Policy object on different domain controllers, it increases the possibility of changes being overwritten by replication. It is strongly recommended that you limit the number of administrators permitted to administer Group Policy, and that you make sure that Group Policy uses the primary domain controller emulator Operations Master. It is also recommended that administrators be aware of other administrators who might be editing the same Group Policy object.

The one used by Active Directory Snap-ins. Uses the domain controller that Active Directory management snap-ins are using. Each of these snap-ins includes an option for changing which domain controller is the focus of its current operations. When this option is selected, the Group Policy snap-in uses the same domain controller.

Use any available domain controller. The third, and least desirable option in most cases, allows the Group Policy snap-in to choose any available domain controller. When this option is used it is likely that a domain controller in the local site will be selected.

You can override all of these options using a policy setting, as described in the following section.

Specifying Policy for Domain Controller Options

The Group Policy snap-in uses the primary domain controller emulator operations master token when editing a Group Policy object. This token makes sure that the Group Policy snap-in is always focused on the same domain controller. User preference options and policy settings are available to modify this behavior so that Group Policy can use a different domain controller.

If you are the primary domain administrator, you can use a policy to specify how Group Policy chooses a domain controller—that is, you can specify which domain controller option should be used. If the selected option is not available, the user receives an error message. When this occurs, the DC Options menu item is shaded (unavailable) because a policy is in place that overrides any setting that the user picks. This policy allows domain administrators to indicate that all administrators must use the primary domain controller, for example. The domain controller options settings are available in the **User Configuration, Administrative Templates, System, Group Policy** node of the Group Policy snap-in. The available domain controller options are the same as the preference settings listed above in the **Options for domain controller selection** dialog box description.

For example, if you are an administrator on one continent and the primary domain controller is on another, you can make your policy edits locally, so that the performance is acceptable. Remember, though, that if someone else edits the same Group Policy object simultaneously, the winner depends on the unpredictable actions of the network.

If the Group Policy snap-in cannot reach the intended domain controller, by default you receive the following error message: "Error Handling on Failure to Reach a Domain Controller." Then you are given the option to cancel the operation or make a selection to retry accessing a domain controller using the following choices:

- The one with the Operations Master token for the primary domain controller emulator.
- The one used by Active Directory Snap-ins.
- Use any available domain controller.

If instead of the error message just described, you get the message "Failed to find a domain controller. There may be a policy that prevents you from selecting another domain controller," then check to see whether the following Group Policy setting is in effect:

<Group Policy object name>/User Configuration/Administrative Templates/System/Group Policy/Group Policy domain controller selection

Domain Controller Selection Results

Table 22.3 shows the results of various combinations of domain controller conditions. The following terms are used in Table 22.3:

- **Primary Domain Controller**: is the domain controller with the Operations Master token for the primary domain controller emulator.
- **Inherit**: is the domain controller used by Active Directory snap-ins.
- **1) and 2)** : means that 1) is tried first then 2).

Table 22.3 Domain Controller Selection Results

User preference	Policy	Inherit domain controller	Results
Undefined	Undefined	N/A	1) Primary domain controller 2) Prompt
Primary domain controller	Undefined	N/A	1) Primary domain controller 2) Prompt
Inherit	Undefined	Yes	Inherit
Inherit	Undefined	No	Any domain controller
Any domain controller	Undefined	N/A	Any domain controller
N/A	Primary domain controller	N/A	Primary domain controller only
N/A	Inherit	Yes	Inherit
N/A	Inherit	No	Any domain controller
N/A	Any	N/A	Any domain controller

Client-side Processing of Group Policy

Some of the Group Policy components include client-side extensions (.dlls) that implement Group Policy at client computers. The client-side extensions are loaded as needed when a client computer is processing policy. The client computer first gets a list of Group Policy objects. Next, it loops through all the client-side extensions and determines whether each client-side extension has any data in any of the Group Policy objects. If a client-side extension has data in a Group Policy object, the client-side extension is called with the list of Group Policy objects that it should process. If the client-side extension does not have any settings in any of the Group Policy objects, it is not called.

Table 22.4 lists the client-side extensions.

Table 22.4 Client-side Extensions

Client-side extension	DLL file name
Registry (in Administrative Templates)	Userenv.dll
Disk Quota (in Administrative Templates)	Dskquota.dll
Folder Redirection	Fdeploy.dll
Scripts	Gptext.dll
Software Installation	Appmgmts.dll
Security	Scecli.dll
IP Security	Gptext.dll
EFS (Encrypting File System) Recovery	Scecli.dll
Internet Explorer Maintenance	iedkcs32.dll
Remote Installation Services	none

Client-side Extension Preferences

The values that the client-side extension puts in the registry are preferences as opposed to Group Policy settings. However, if you decide that the client-side extension should run across a slow link, regardless of the amount of data, you can enable these policies.

All MMC snap-ins register themselves in the following location in the registry:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MMC\SnapIns\{*Snap-in-GUID*}

Possible REG_SZ value names are as follows:

- About
- NameString
- NodeType
- Provider
- Version

In addition to the location mentioned earlier, Group Policy Client Side Extensions also register themselves in the following location in the registry:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\GPExtensions\{CSE-GUID}

Possible REG_DWORD value names are as follows:

Important The REG_DWORD value names listed here are informational only. It is recommended that you do not change these.

- EnableAsynchronousProcessing
- LastPolicyTime
- MaxNoGPOListChangesInterval
- NoBackgroundPolicy
- NoGPOListChanges
- NoMachinePolicy
- NoSlowLink
- NoUserPolicy
- NotifyLinkTransition
- PerUserLocalSettings
- PrevSlowLink
- RequiresSucessfulRegistry
- Status
- This possible REG_SZ valuename
- ProcessGroupPolicy
- This REG_EXPAND_SZ valuename
- DllName

Note LastPolicyTime and Status are set automatically by the Group Policy engine, not by the client-side extensions.

Computer Policy for Client-side Extensions

A computer policy exists for each of the Group Policy client-side extensions. Each policy includes a maximum of three options. Some of the client-side extensions include only two computer policy options; in those cases, this is because the third option is not appropriate for that extension. The computer policy options are as follows:

Allow processing across a slow network connection. When a client-side extension registers itself with the operating system, it sets values in the registry, specifying whether it should be called when policy is applied across a slow link. Some extensions move large amounts of data, so processing across a slow link can hurt performance. Installing a large application across a 28.8 Kbps modem line is impractical.

Do not apply during periodic background processing. Computer policy is applied at startup, as well as periodically in the background, approximately every 90 minutes. User policy is applied at user logon, then every 90 minutes. Some extensions process policy only initially, not periodically, because processing that took place in the midst of a user's session would be disruptive. For example, with Software Installation, applications are installed or upgraded during the initial run and not in the background. If it were done in the background, a hapless user might be running an application even while having it uninstalled. Or the application might have a shared component that is in use by another application, preventing the installation from completing successfully. The **Do Not Apply During Periodic Background Processing** option gives you the ability to override this logic and force the extension to either run or not run in the background.

Process Even If The Group Policy Objects Have Not Changed. By default, if the Group Policy objects on the server have not changed, it is not necessary to continually reapply them to the client, because the client should already have all the settings. However, users might be able to change some settings if they are administrators of their computers. In this case, it might make sense to reapply these settings when the user logs on or during the periodic refresh cycle to get the computer back to the desired status.

For example, if you have used Group Policy to define a specific set of security options for a file, and the user with administrative privileges logs on and changes it, then, you might want to set the policy to process Group Policy even if the Group Policy objects have not changed. This makes sure that security is reapplied periodically and at every startup. This also applies to applications. Group Policy installs an application, but the end user can remove the application or delete the icon. The **Process Even If The Group Policy Objects Have Not Changed** option gives you the ability to restore the application at the next user logon session.

Table 22.5 lists the client-side extensions that include only two computer policy options, as well as the reason for this.

Table 22.5 Client-side Extension and Policy Options

Client-side extension	Missing policy checkbox	Reason
Administrative Templates	Slow link (Allow processing across a slow network connection)	Registry policy is always applied because it controls the other client-side extensions.
Security Settings	Slow link (Allow processing across a slow network connection)	To ensure that security settings are in effect, they must always be applied, even across a slow link.
Folder Redirection	Background processing (Do not apply during periodic background processing)	User might be using the folders or their contents.
Software Installation	Background processing (Do not apply during periodic background processing)	It would be disconcerting to the user to have an application uninstall while it is open.

The processing of policy is also affected by issues that are not governed by specific policy settings and not apparent in the user interface. The include the following:

Messages and Events When Group Policy is applied, a WM_SETTINGCHANGE message is sent, and an event is signaled. Applications that can receive window messages can use it to respond to a Group Policy change. Those applications that do not have a window to receive the message (as with most services) can wait for the event.

On-Demand Processing Group Policy can also be applied on demand. To do this, applications can call the RefreshPolicy function, which allows applications to request a policy refresh. The administrator can refresh policy from the command line as follows:

1. Click **Start,** and then click **Run**.
2. To refresh policies under the **Computer Configuration** node, type the following, and then click **OK**:

   secedit /refreshpolicy MACHINE_POLICY [/enforce]
3. To refresh policies under the **User Configuration** node, type the following, and then click **OK**:

   secedit /refreshpolicy USER_POLICY [/enforce]

The optional "/enforce" switch causes policy for the Security and EFS extensions to refresh whether or not there is a policy change. For other extensions it has no effect.

Time Limit for Processing of Group Policy There is a time limit of 60 minutes for all the client-side extensions to finish processing policy. An errant client-side extension that is not finished after 60 minutes will be stopped and the associated policy settings will not be processed. There is no Group Policy setting to change the default time limit.

Using Group Policy on Stand-alone Computers

You can set local Group Policy for computers that are not members of a domain. To set local Group Policy, you use the Group Policy snap-in focused on the local computer. You can gain access to the Group Policy snap-in by typing "MMC" at the command prompt, adding the Group Policy snap-in to the MMC console, and focusing the Group Policy snap-in on the local computer.

Local Group Policy Object

Local Group Policy objects exist on stand-alone computers, however it consists of only the Group Policy template portion of a Group Policy object. The location of the local Group Policy object is %SystemRoot%\System32\GroupPolicy. Each Group Policy extension snap-in queries Group Policy to receive the Group Policy object type (local or Active Directory–based), and then determines if it should be displayed in the console.

Table 22.6 indicates whether or not each Group Policy snap-in extension opens when the Group Policy snap-in is focused on a local Group Policy object.

Table 22.6 When Group Policy Snap-in Loads

Group Policy snap-in extension	Loads when Group Policy snap-in focused on local Group Policy object
Security Settings	Yes
Administrative Templates	Yes
Software Installation	No
Scripts	Yes
Internet Explorer Maintenance	Yes
Remote Installation Services	No
Folder Redirection	No

Starting Group Policy on Windows 2000 Professional

Windows 2000 Professional does not provide a preconfigured MMC console for accessing non-local Group Policy directly, except for Security Settings, which can be accessed from Control Panel. However, you can create your own custom Group Policy console by taking the following steps:

▶ **To start the Group Policy snap-in on Windows 2000 Professional**

1. Click **Start**, click **Run**, type **MMC**, and then click **OK**.
2. In the MMC window, on the **Console** menu, click **Add/Remove Snap-in**.
3. On the **Standalone** tab, click **Add**.
4. In the **Add Snap-in** dialog box, click **Group Policy**, and then click **Add**. The **Select Group Policy object** dialog box appears.
5. Click **Local Computer** to edit the local Group Policy object, or **Browse** to find the Group Policy object that you want to use.
6. Click **Finish**.
7. Click **OK**. The Group Policy snap-in opens focused on the specified Group Policy object.

Note To use the Group Policy snap-in focused on a remote computer, you must have administrative rights on the target computer in addition to appropriate permission to use the snap-in.

Using the Group Policy Snap-in Focused on a Remote Computer

The Group Policy object seen at the root node of the Group Policy console is said to have "focus." The console can be focused on any computer's local Group Policy object, or any Active Directory–based Group Policy object.

Note Focusing the Group Policy snap-in, whether on a remote computer or the local computer, or on an Active Directory–based Group Policy object, must be done when the extension is added to an MMC console file, or as a command line option. The focus cannot be changed while the Group Policy console is in use.

▶ **To add Group Policy to an MMC console focused on a specific remote computer**

1. Click **Start**, click **Run**, and type **MMC**. Or you can open an existing saved console such as Console1.mmc.
2. In the **MMC** window, on the **Console** menu, click **Add/Remove** Snap-in.
3. On the **Standalone** tab, click **Add**.
4. In the **Add Snap-in** dialog box, click **Group Policy**, and then click **Add**. By default this is set to open on the local computer.
5. Click **Browse**.
6. You can now select a Group Policy object from Active Directory or, as in this case, select the **Computer** tab.
7. Select **Another Computer**.
8. Either type in the computer name or click **Browse** to locate it.
9. Select the domains to which you have access in the **Look in** drop-down list.

The supported computer name formats are:

- NetBIOS names; for example:

  ```
  ThisComputer
  ```

- DNS-style; for example:

  ```
  ThisComputer.Reskit.com
  ```

You can start the Group Policy snap-in with the following two command line switches:

- Specific computer

  ```
  /gpcomputer:<machinename>
  ```

  Where <machinename> can be either a NetBIOS or a DNS-style name.

  For example:

  ```
  gpedit.msc /gpcomputer:"ThisComputer"
  ```

  or

  ```
  gpedit.msc /gpcomputer:"ThisComputer.Reskit.com"
  ```

  Note that there is no space following:

  ```
  /gpcomputer:
  ```

  Also, the quotes are necessary, not optional.

- Specific ADSI path

  ```
  /gpobject:"<ADSI path>"
  ```

  For example:

  ```
  /gpobject:"LDAP://CN={31B2F340-016D-11D2-945F-
  00C04FB984F9},CN=Policies,CN=System,DC=Reskit,DC=com"
  ```

  in which the GUID for the Group Policy object is a made-up example.

For these command line options to function with a saved console file, you must select the check box titled "Allow the focus of the Group Policy snap-ins to be changed when launching from the command line." This only applies if you save the console. The Gpedit.msc file supplied with Windows 2000 has this option enabled.

Note: The Security Settings extension does not support remote management for local policy in Windows 2000.

Local Group Policy Object Processing

The local Group Policy object is processed even when the **Block Policy Inheritance** option has been specified on a domain or organizational unit.

Local Group Policy objects are always processed first, and then non-local (that is, Active Directory–based) policy is processed. If a computer is participating in a domain, and a conflict occurs between non-local and local computer policy, then by default, non-local policy prevails by overwriting local policy. If a computer withdraws from a domain, local Group Policy object policy settings are still applied and assume greater importance because they can no longer be overwritten.

If the Computer Account object and User Account object are both managed by Windows NT 4.0 domain controllers and are therefore not in Active Directory, then no local Group Policy object will be processed. For details about other interoperability situations that can arise during migration, see "Migration Issues Pertaining to Group Policy" later in this chapter.

Group Policy Loopback Support

Group Policy applies to the user or computer in a manner that depends on where both the user and the computer objects are located in Active Directory. However, in some cases, users might need policy applied to them based on the location of the computer object alone. The Group Policy loopback feature gives you the ability to apply Group Policy objects that depend only on which computer the user logs on to.

Note Loopback is supported only in a purely Windows 2000–based environment. Both the computer account and the user account must be in Active Directory. If either account is managed by a Windows NT 4.0–based domain controller, loopback does not function. The client computer must be a Windows 2000–based computer.

The following scenario describes the loopback feature. In this scenario, you have full control over the computers and users in this domain because you have been granted domain administrator rights. Figure 22.7 shows the Reskit domain.

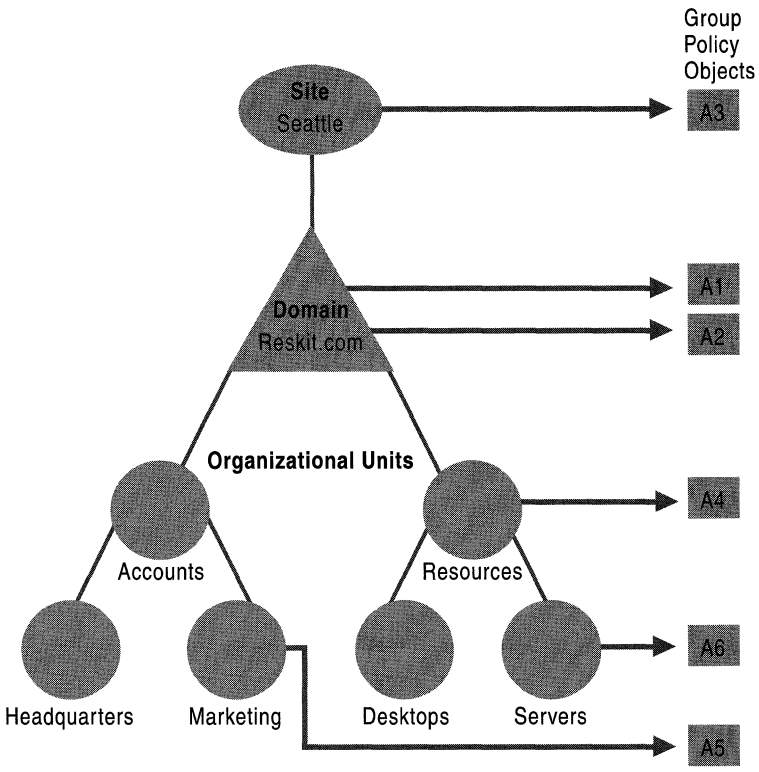


Figure 22.7 The Reskit Domain

When users work in their own workstations, they should have Group Policy applied to them according to the policy settings defined, based on the location of the user object. However, when users log on to a computer whose computer object is in the server's organizational unit, they should receive user policy settings based on the computer object location, rather than the user object location.

In Figure 22.8, normal user Group Policy processing specifies that computers located in the server's organizational unit have the Group Policy objects A3, A1, A2, A4, and A6 applied in that order during computer startup. Users of the Marketing organizational unit have Group Policy objects A3, A1, A2, and A5 applied in that order, regardless of which computer they log on to.

In some cases this processing order might not be appropriate; for example, when you do not want applications that have been assigned or published to the users of the Marketing organizational unit to be installed while they are logged on to the computers in the Servers organizational unit. With the Group Policy loopback support feature, you can specify two other ways to retrieve the list of Group Policy objects for any user of the computers in the Servers organizational unit. You can use either the Merge Mode or the Replacement Mode.

Merge Mode. In this mode, when the user logs on, the user's list of Group Policy objects is gathered normally by using the **GetGPOList** function, and then **GetGPOList** is called again using the computer's location in Active Directory. Next, the list of Group Policy objects for the computer is added to the end of the Group Policy objects for the user. This causes the computer's Group Policy objects to have higher precedence than the user's Group Policy objects. In this example, the list of Group Policy objects for the computer is A3, A1, A2, A4, and A6, which is added to the user's list of A3, A1, A2, and A5, and thus results in A3, A1, A2, A5, A3, A1, A2, A4, and A6 (listed in lowest to highest priority).

Replace Mode. In this mode, the user's list of Group Policy objects is not gathered. Only the list of Group Policy objects based upon the computer object is used. In Figure 22.7, the list is A3, A1, A2, A4, and A6.

The loopback feature was implemented in the Group Policy engine, not in the **GetGPOList** function. When the Group Policy engine is about to apply user policy, it searches in the registry for a computer policy, which specifies which mode user policy should be applied in. Then, based upon this policy, it calls **GetGPOList**, as appropriate.

Supporting Windows NT 4.0, Windows 95, and Windows 98 Clients

The Windows 2000 Group Policy does not provide client support for Windows NT 4.0–based, Windows 95–based, or Windows 98–based computers, notwithstanding the availability of Active Directory client support for those earlier versions of Windows.

Policy support for Windows NT 4.0–based clients is provided by using Windows NT 4.0–style administrative templates (.adm files) and Windows NT 4.0 System Policy Editor (Poledit.exe) files.

Windows 95 and Windows 98 clients need to be managed with the System Policy Editor.

Client computers that are running Windows NT 4.0, Windows 95, or Windows 98 need to have the .pol file (Config.pol for Windows 98 or Windows 95, or Ntconfig.pol for Windows NT 4.0) created on the client computer's operating system copied to the domain's Netlogon share. This is: %systemroot%\SYSVOL\sysvol\<*domain name*>\SCRIPTS under Windows 2000, or %systemroot%\winnt\system32\Repl\Import\Scripts under Windows NT 4.0.

For information about installing the System Policy Editor, see Windows 2000 Server Help. The System Policy Editor is included with Windows 2000 Server but is not included with Windows 2000 Professional. The Windows 2000 Optional Administrative Tools package (Adminpak.msi), which includes the System Policy Editor, comes on the Windows 2000 Server companion CD for installation onto computers running Windows 2000 Professional.

Using Windows NT 4.0 Administrative Templates in the Windows 2000 Group Policy Console

On the menu bar of the Group Policy console, the **Show Policies Only** setting is under the **View** button. This is active (checked) by default. It prevents Windows NT 4.0 Administrative Templates, which are used to supply namespace for the System Policy Editor, from supplying that same namespace in the Group Policy console. This is safest, because Windows NT 4.0 registry-based policy is undesirably persistent from the Windows 2000 administrative perspective, and it is best to not use System Policy on Windows 2000 clients. However, it is possible to 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 are hidden again.

It is recommended that you use the **Enforce Show Policies Only** setting, in User Configuration/Administrative Templates/System/Group Policy if you delegate Group Policy administrative tasks using custom consoles. This prevents users of the console from unchecking **Show Policies Only**.

Migration Issues Pertaining to Group Policy

In organizations containing many networked Windows NT 4.0–based computers, including primary domain controllers, backup domain controllers, client workstations, and client stand-alone servers, it might not be practical to upgrade all computers simultaneously to Windows 2000. Therefore, when a network of Windows NT 4.0–based computers is upgraded to Windows 2000, you need to know what to expect during and after the migration process.

Before the migration is complete you might need to manage a domain in which all of the following types of computers participate:

- Windows 2000 Server domain controllers
- Windows NT 4.0 Server domain controllers
- Windows 2000 Professional clients
- Windows 2000 Server clients
- Windows NT 4.0 Workstation clients
- Windows NT 4.0 Server clients
- Computers running earlier versions of Windows, such as Windows 98

There are many interactions to consider. Fortunately, the conditions on the client side are not complicated.

The Client Side

The only computers that are subject to and are able to use Group Policy are Windows 2000–based computers. Client computers running Windows NT 4.0 receive System Policy as administered through the System Policy Editor (Poledit.exe). Client computers running Windows 98 or Windows 95 are also managed using System Policy compatible with those operating systems. You can only use the .pol files created using Poledit.exe on computers running the operating system on which the .pol file was created. Windows NT–based computers cannot use Config.pol (Windows 98 or Windows 95 System Policy), nor can Windows 98–based or Windows 95–based computers use NTConfig.pol (Windows NT System Policy).

The Domain Controller Side

The interaction of Windows NT 4.0 System Policy and Windows 2000 Group Policy during migration is described in this section. You can assume that the client computers, meaning all computers other than the domain controllers, run either Windows 2000 Professional or Windows 2000 Server unless the contrary is specifically stated.

For a user to log on to a domain successfully, both the user and the computer must be known to the domain. You need to know what behavior to expect when computer or user accounts, or both, have not yet been upgraded from Windows NT 4.0 to Windows 2000.

Computer and User Accounts Both on Windows NT 4.0 Domain Controllers.

These accounts might be on the same or different domain controllers. There might be Windows 2000 domain controllers on the intranet as well; however, they don't handle these particular accounts. The user and computer are not in Active Directory.

System Startup

Local Group Policy for the computer is applied when the computer starts up.

User Logon

Windows NT 4.0 System Policy for the computer is applied. Then, Windows NT 4.0 System Policy for the user is applied. Then, if local Group Policy has changed since it was last applied, the following policy settings are applied: Local Group Policy for the user, followed by Windows NT 4.0 System Policy for the user.

Computer and User Accounts Both on Windows 2000 Domain Controllers

The user account and computer account are both in Active Directory. There might be Windows NT 4.0 domain controllers on the intranet as well, but they are not involved in the startup/logon negotiation because Windows 2000 clients prefer Windows 2000 domain controllers.

System Startup

Windows 2000–based computer Group Policy is applied at boot time.

User Logon

Windows 2000 user Group Policy is applied when the user logs on.

Computer is Managed in a Windows NT 4.0 Account and User is Managed in a Windows 2000 Account

The user account is in Active Directory, and the computer account is not. The computer account is managed by a Windows NT 4.0 domain controller. This is a common scenario.

System Startup

Local Group Policy for the computer is applied when the computer starts up.

User Logon

When the user logs on, the computer receives System Policy, and then the user receives all Group Policy to which the user is entitled. The user does not receive System Policy.

Upgrading the Computer Accounts

Persistent registry settings can be an issue when upgrading the computer accounts from Windows NT 4.0 to Windows 2000. While the client computer was subject to System Policy, its registry received settings outside the approved Group Policy trees, and these are not removed on the client when the domain controller is upgraded. You should look for unwanted residual effects of System Policy and take corrective steps, such as using Regini.exe, found in %systemroot%/System32/, to remove the old settings.

For example, Windows NT 4.0 has a Logon Banner policy. In Windows 2000, Logon Banner policy is handled differently, in Security Settings rather than using an Administrative Template. If you observe after upgrading the computer account that the Windows NT 4.0 Logon Banner policy is still in force, then reverse that setting on a one-time basis.

It is recommended that you avoid issues such as these by giving the client computer a freshly installed Windows 2000 operating system, rather than an upgrade. If you do this, there are no holdover Windows NT 4.0 registry settings.

User is Managed in a Windows NT 4.0 Account and Computer is Managed in a Windows 2000 Account

The computer account is in Active Directory, and the user account is not. The user account is managed by a Windows NT 4.0 domain controller.

Windows NT 4.0 resource domains (often containing computer accounts, printers, shared folders, and so on.) are often made into Windows 2000 organizational units in Active Directory. In this way, what were several Windows NT 4.0 resource domains can be handled in just one Windows 2000 domain. Because fewer computers are typically needed when upgrading resource domains in this way than when upgrading all the user accounts, this migration status is less common than the previous one.

System Startup

All Group Policy to which the computer is subject is applied to the computer when it boots.

User Logon

System Policy is applied to the user when the user logs on. If the local Group Policy object has changed since it was last processed, the following policy settings are applied: Local Group Policy for the user, followed by Windows NT 4.0 System Policy for the user. Computer System Policy is not applied.

Upgrading the User Accounts

During the time that the user accounts were managed by a Windows NT 4.0 domain controller, the client computers might have had their registries altered outside the approved Group Policy trees. When the domain controller holding the user accounts is upgraded to Windows 2000, these settings remain on the client computers unless the administrator undoes them by means of System Policy or—easier for the administrator—the client computers get fresh installations of Windows 2000.

Trust Relationships with Previous Versions of Windows

There is a subtle migration issue you need to avoid 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 (call it A) with a previous version trust relationship to a Windows NT 4.0 domain controller (call it B). You upgrade B to Windows 2000 and then link an organizational unit managed by A to a Group Policy object stored in B's domain. A user in the organizational unit logs on to A expecting to receive policy from the Group Policy object stored in B's domain—but it doesn't work. The reason is that the upgrade of the domain controller does not automatically upgrade the trust relationship, and the user won't have access to the Sysvol share on B.

To solve this problem, you need to break the trust after upgrading B to Windows 2000. Then create a new Windows 2000–style trust and the user receives Group Policy as expected.

Best Practices

Use Group Policy in Preference to Windows NT 4.0 System Policy

System Policy is undesirably persistent from a Windows 2000 perspective. Group Policy is cleaned up and refreshed whenever policy changes.

Disable Unused Parts of a Group Policy Object

If you notice that under the User Configuration or Computer Configuration node of the console, a Group Policy object only has settings that are **Not Configured**, then you can avoid processing those settings by disabling the node. This expedites startup and the logon session for those users and computers subject to the Group Policy object.

Disabling both parts of a Group Policy object makes it behave as if it is not linked to any site, domain, or organizational unit, even though the links still exist.

Use the Block Policy Inheritance and No Override Features Sparingly

Routine use of these feature makes it difficult to troubleshoot policy.

Minimize the Number of Group Policy Objects Associated with Users in Domains or Organizational Units

The more Group Policy objects are applied to a user, the longer it takes to log on.

Filter Policy Based on Security Group Membership

Keep in mind that a Group Policy object will not apply to a user if the Read or Apply Group Policy access control entries (ACEs) are not set to **Allow** on security groups of which the user is a member. This is the mechanism by which policy can be prevented from applying to users (or computers) who would otherwise be subject to it either by links or by inheritance. It is a good, efficient mechanism, and the administrator can greatly expedite the logon and startup experiences of the users in his or her organization by exploiting it fully.

Override User-Based Group Policy with Computer-Based Group Policy Only When Necessary

Do this only if you need the desktop configuration to be the same regardless of which user logs on.

Avoid Cross-Domain Group Policy Object Assignments

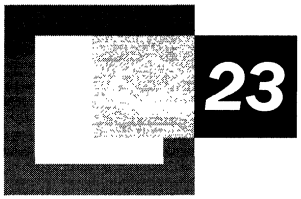
The processing of Group Policy objects slows the logon session and startup if Group Policy is obtained from another domain.

Additional Resources

- For the most recent information about Group Policy in Windows 2000 Server, see the Microsoft Windows 2000 Server link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.
- For information about the administrative templates settings that are included with Windows 2000 Server, see the searchable reference file GP.chm on the Windows 2000 Server Resource Kit CD-ROM.
- For more information about Group Policy in Windows 2000 Server, see the ResourceLink link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.
- For more information about the Group Policy API, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.
- For information about the Group Policy API see the *Microsoft® Windows® 2000 Platform Software Development Kit.*
- For information about Microsoft Management Consoles (MMC), see the *Microsoft® Windows® 2000 Platform Software Development Kit.*
- For Group Policy walk-throughs, see the Windows 2000 Management Services link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

CHAPTER 23

Software Installation and Maintenance



With Microsoft® Windows® 2000, you can manage software throughout your organization by using a Group Policy–based software management system. This centralized Group Policy–based environment minimizes support costs and allows you to efficiently install and update software without visiting users' desktops.

In This Chapter

Related Information in the Resource Kit

- For more information about Active Directory™, see "Active Directory Logical Structure" in this book.
- For more information about Group Policy, see "Group Policy" in this book.

Introduction

You can use the software installation and maintenance feature of IntelliMirror™ to centrally manage software installation, repairs, updates, and removal throughout your organization. You can use the software installation and maintenance feature to create a managed software environment in which the following conditions exist:

- Users have access to the applications that they need to do their jobs, no matter which computer they are using.
- Computers have the applications (such as virus protection software) that they need without a visit from a technical support person.
- Applications are updated, maintained, and removed according to rules that you define.

To take advantage of the software installation and maintenance feature, you need to be running Microsoft® Windows® 2000 Server, with Active Directory and Group Policy on a server, and Microsoft® Windows® 2000 Professional on your client computers.

Note Use Remote OS Installation for the initial operating system installation; use software installation for operating system upgrades. For information about Remote OS Installation, see "Remote OS Installation" in this book.

Table 23.1 lists the three software installation and maintenance features provided with Windows 2000 Server.

Table 23.1 Software Installation and Maintenance Features

Component	Role
Software Installation snap-in	Used to manage software in a Group Policy–based software management system.
Windows Installer	Installs software that is packaged in Windows Installer .msi files.
Add/Remove Programs in Control Panel	Used by users to manage software on their own computers.

Software Installation

The Software Installation snap-in is a Microsoft Management Console (MMC) snap-in that extends Group Policy and is your primary tool for managing software. By using the Software Installation snap-in, you can establish a Group Policy–based software management system so that you can centrally manage the following:

- Initial deployment of software, including shrink-wrapped productivity applications and in-house or line-of-business applications.
- Upgrades, patches, and quick fixes for software. You can update the software or replace it. You can even upgrade the operating system by using service packs.
- Removal of software. You can remove software when users no longer require it or when you no longer support it.

By using the Software Installation snap-in, you can centrally manage the installation of software on a client computer either by *assigning* applications to users or computers or by *publishing* applications for users. Assigned software is available on 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.

Assign required or mandatory software to users or to computers. Assign software when you do not want users to be able to choose not to install software or to be able to remove it. For example, if all of your users need e-mail to do their jobs, assign Microsoft® Outlook® or another e-mail program to all users. If all of your computers require virus protection, assign virus protection software to all computers.

When you assign software to a user, the next time that user logs on to a computer the software is advertised on the desktop, and local registry settings, including file name extensions, are updated. The software advertisement follows the user regardless of which computer he or she uses. The software is set to fully install on first use, which means the software is installed the first time that the user chooses an advertised shortcut on the desktop or **Start** menu or opens a document with the associated file name extension.

When you assign software to a computer, the application is advertised and the installation is performed when it is safe to do so. Typically, installation occurs when the computer starts up, so that there are no competing processes on the computer. Computer-assigned software is fully installed the next time that the computer is restarted.

Publish software that users might find useful to perform their jobs. When you publish an application, it does not appear installed on users' computers but is instead available for users to install by using **Add/Remove Programs** in Control Panel. For example, not everyone needs to make business presentations as a daily part of their jobs, but many users do read presentations that others prepare. You can choose to assign Microsoft® PowerPoint® to users who need it and publish the application to users whose needs you cannot anticipate.

For more information about assigning and publishing software, see "Manage Your Software" later in this chapter. For information about how to assign or publish software, see Windows 2000 Server Help.

Windows Installer

Windows 2000 uses Windows Installer technology, which uses the following components to install, manage, and maintain software:

- An operating system service that performs the installation, modification, and removal of the software in accordance with the information in a Windows Installer package.
- The *Windows Installer package* (an .msi file), which is a database containing information that describes the installed state of the application.
- An application programming interface (API) that allows applications to interact with Windows Installer to install or remove additional features of the application after the initial installation is complete.

Note Windows Installer is a base service of the Windows operating system; therefore, it is available with Windows 2000, Microsoft® Windows NT® version 4.0, Microsoft® Windows® 98, and Microsoft® Windows® 95.

Windows Installer works by managing the state of the installation. Because Windows Installer knows the state of the software, it can ensure that the software is installed correctly. If there is a problem during installation, Windows Installer can roll back the installation, ensuring the computer is left in a working (known good) state. The management of state also allows you to modify the installation, adding or removing features after installation. Because Software Installation uses Windows Installer, users can take advantage of self-repairing applications. Windows Installer notes when a program file is missing and immediately reinstalls the damaged or missing files thereby fixing the application. And finally, using the state of installation to manage the installation also allows for complete removal of the software when it is no longer needed.

For more information about Windows Installer, see "Windows Installer Technology" later in this chapter.

Add/Remove Programs in Control Panel

You can control which software is available to users within **Add/Remove Programs** in Control Panel by using Group Policy settings. Users can then use **Add/Remove Programs** to install, modify, or remove an existing application or repair a damaged application.

In order to maintain software in a Group Policy–based centrally managed state, you might choose not to allow the installation of software from local media, such as a compact disc (CD) or floppy disk. By using Group Policy, however, you can give users the ability to install software from local media, from a defined Active Directory location (assigned and published software on a corporate network), or from the Internet.

Because software installation can provide your users immediate access to the software that they need to perform their jobs on the desktop, users are going to have an easy and consistent experience when working with software throughout its life cycle. Users no longer need to look for a network share, use a CD, or struggle to install, fix, and upgrade software themselves.

Phases of Software Management

Using a systematic method for deploying and upgrading software shortens the time that it takes to get software to the users who need it. You can use software installation and maintenance to manage software throughout the phases that are described in Table 23.2.

Table 23.2 Phases of Software Management

Phase	Process
Preparation	Analyze the software needs of your organization, and then gather or create Windows Installer packages for the software you want to install.
Distribution	Create a software distribution point on the network, and ensure that the software you want to manage is available.
Targeting (scope of management)	Determine who must get the software—who needs the software to perform his or her job?
Installation	Manage the state of the software, whether it is installed, modified, repaired, or removed.

Preparation Phase

During the preparation phase, you determine how to centrally manage the software that is going to be available to users in your organization. To prepare for the deployment of new software in your organization, you must do the following:

- Analyze your organization's software requirements by using your overall organizational structure within Active Directory and your available Group Policy objects.
- Gather or create Windows Installer packages, or if necessary, determine the existing setup programs for the software that you want to install.

Important It is your responsibility to acquire the appropriate number of licenses for the software that you deploy.

Analyze Software Requirements

Use the Windows 2000 Server components listed in Table 23.3 to assist you in documenting your overall organizational structure and the groups of users to whom you are going to deploy software. These components include Active Directory, Group Policy, and the Software Installation snap-in.

Table 23.3 Windows 2000 Software Installation Server Components

Component	Overall Function	Software Installation and Maintenance Function
Active Directory	The directory service included with Windows 2000 Server. It stores information about objects on a network and makes this information available to users and network administrators. Active Directory gives you an intuitive hierarchical view of the network and a single point of administration for all network objects.	With Group Policy, provides the scope of management mechanism to locate users and computers and stores software installation and maintenance information.

(continued)

Table 23.3 Windows 2000 Software Installation Server Components *(continued)*

Component	Overall Function	Software Installation and Maintenance Function
Group Policy	A tool that allows you to centrally manage user and computer settings, software policies, scripts, application deployment, and security settings within Active Directory.	Allows you to deploy applications within a Group Policy object that are associated with an Active Directory container such as a site, domain, or organizational unit.
Software Installation snap-in	An MMC snap-in that you use to assign or publish applications to users or computers. For more information about Software Installation, see "Software Installation" earlier in this chapter.	Used to centrally manage the installation and maintenance and upgrade of applications.

Note You can use Microsoft® Systems Management Server (SMS) as an additional tool to assist with analyzing your existing organizational structure, including collecting details on software that is currently installed on existing systems. For information about how to use SMS to analyze your network infrastructure, see "Using Systems Management Server to Analyze Your Network Infrastructure" in the *Microsoft® Windows® 2000 Server Resource Kit Deployment Planning Guide.* The results of this analysis can help you to determine what network infrastructure changes that you might need to make prior to managing software with Windows 2000.

Active Directory

Plan your organization's Active Directory with software installation and maintenance and Group Policy in mind. In Active Directory, you create domains as security boundaries and typically base them on functional divisions. Another common reason to create a separate domain is because of bandwidth restrictions over a wide area network (WAN). If many of your users connect over slow links, you need to define an appropriate deployment strategy for those users. Domain design can influence or be influenced by the bandwidth necessary for your software management strategy.

Important Because you manage software by using Group Policy, make sure to review the structure of Active Directory and determine which Group Policy objects apply to each Active Directory container, site, domain, and organizational unit.

For more information about Active Directory, see Windows 2000 Server Help. For information about expanded guidelines for domain planning, see "Active Directory Logical Structure" in this book.

Group Policy

Companies benefit from Group Policy–based centralized administration because both users and computers have the applications that users need to do their jobs, no matter what computer they are using.

By using Group Policy, you specify requirements for your users' environments, and then you can rely on Windows 2000 to continually manage these requirements. You can use the software installation and maintenance technology built into Group Policy to centrally manage software, whether it is assigned to users and computers or published to users. For example, by using Group Policy, you can tightly manage task-oriented desktops and assign specific software, thus creating a resilient and tamper-proof environment. If you need a more adaptable environment for roaming users, you can set Group Policy to provide the flexibility to install applications from the server or another location such as a CD or other form of removable media.

Group Policy security settings filter the users or computers that receive specific software. As a best practice, separate user and computer restrictions by organization-wide concerns and department or organizational unit concerns. This helps determine which Group Policy settings you apply at the site level and which you apply at the domain and organizational unit level.

Table 23.4 describes strategies and considerations for applying software deployment policies by using Group Policy and Active Directory. Use the strategies that meet your business goals.

Table 23.4 Strategies and Considerations for Software Deployment

Strategy	Consideration
Create organizational units based on software management needs.	Allows you to target applications to the appropriate set of users. Group Policy security settings are not required to target the appropriate set of users.
Deploy software high in the Active Directory tree.	Makes it easy to provide all users in an organization with access to an application. Reduces administration because you can deploy a single Group Policy object rather than having to re-create that object in multiple containers that are lower in the Active Directory tree.

(continued)

Table 23.4 Strategies and Considerations for Software Deployment *(continued)*

Strategy	Consideration
Deploy multiple applications with a single Group Policy object.	Reduces administration overhead by allowing you to create and manage a single Group Policy object rather than multiple Group Policy objects. The logon process is faster because a single Group Policy object deploying 10 applications processes faster than 10 Group Policy objects deploying one application each. Appropriate in organizations where users share the same core set of applications.
Avoid publishing or assigning one application multiple times in the same Group Policy objects or in a series of Group Policy objects that might apply to a single user or computer.	Makes it difficult for you to determine which instance of the application applies to the user or computer.

Keep in mind that when you manage software installation and maintenance, Group Policy can be applied sequentially at the following levels within Active Directory:

- Site
- Domain
- Organizational Unit (OU)

Note There is no support for Group Policy that is installed on client computers that are running Windows NT 4.0 or earlier, Windows 98, or Windows 95; therefore, software installation and maintenance supports the management of software on Windows 2000 Professional.

Gather or Create Windows Installer Packages

After you have analyzed your organization's software requirements, you can gather or create your Windows Installer packages. You can only assign or publish software by using Software Installation if the file type fits one of the following categories:

- Native Windows Installer package (.msi) files that are developed as a part of the application and take full advantage of Windows Installer.

- Repackaged application (.msi) files that you use to repackage applications that do not have a native Windows Installer package in much the same way that you repackage software today to customize installations.
- An existing setup program—an application (.zap) file—that installs an application by using its original Setup.exe program. (These files can only be published.)

Native Windows Installer Packages

Native Windows Installer packages provide the best overall software installation and maintenance experience. Native Windows Installer packages are developed as a part of the application and take full advantage of Windows Installer. The author or publisher of the software can supply a natively authored Windows Installer package. For example, Microsoft® Office 2000 provides a Windows Installer package. Native Windows Installer packages support on-demand installation, meaning that you can set individual application features so that they do not install until the user actually uses them. For example, Help files are not installed until the user attempts to gain access to Help within the application. These packages also support self-repairing applications if an application is corrupted. For example, if a user accidentally deletes Winword.exe, Microsoft® Word recognizes that a critical file is missing and reinstalls it. Tools are available from a variety of third-party tool vendors that developers can use to create Windows Installer packages. For more information about creating Windows Installer packages, see "Windows Installer" earlier in this chapter.

Repackaged Windows Installer Packages

Repackaged Windows Installer packages provide the next best software installation and maintenance experience. You can repackage applications that do not have a native Windows Installer package in much the same way that you repackage software today to customize the installation. The only difference in repackaging software with a Windows Installer repackager is that the output of the repackaging operation is a Windows Installer package rather than a package in a proprietary format. Repackaged applications work in the same way that native Windows Installer packages do. However, a repackaged Windows Installer package contains a single product, and all components and applications associated with that product, that is installed as a single feature. Native Windows Installer packages contain a single product that is made up of many features that can be individually installed.

Using Existing Setup Programs

In some situations, you might not be able to justify developing a native Windows Installer package or repackaging the application to create a Windows Installer package. To publish these existing software applications, you need to define the Setup.exe or Install.exe files into a .zap file to deploy them.

A .zap file is a text file that contains instructions about how to publish an application. You can use .zap files for applications to be phased out within a few months. A .zap file does not support the features of Windows Installer. For example, if an application requires a user to have administrator privileges to install it, preparing the application as a .zap file and publishing it still requires the user to have administrator privileges, whereas you can control these privileges in a Windows Installer package. When you deploy an application by using a .zap file, the application is installed by using its original Setup.exe program. The software is published so that users can select it only by using **Add/Remove Programs** in Control Panel.

Customizing Software

After you have your software in the necessary package format, you might decide that you want to customize the software for your organization. For example, you might want to remove unnecessary features from an application, such as clip art, and add customized templates. In the past, if you wanted to customize the installation of software you had to either adjust the setup program instructions, which was both difficult and problematic, or repackage the software after it was installed. The Windows Installer package format addresses this customization issue by allowing you to transform the original package. With Windows Installer, you can create a *transform*. A transform is a specialized Windows Installer package that when associated with a Windows Installer package at deployment time modifies the original Windows Installer package.

You can use the following file types to modify an existing Windows Installer package:

- Transforms (.mst files), also called modifications, provide a way for you to customize the installation of an application by using software installation and maintenance. You can use transform files to deploy the specific features of the software to match the needs of the users. For example, you can include or exclude specific features of an application to provide a set of features that users require to do their jobs. For more information about transforms, see "Customizing Windows Installer Packages" later in this chapter.
- Software patches, service packs, and some software update (.msp) files, including bug fixes, can be distributed by using the .msp file type and can be used to update an existing .msi file. An .msp file is a package modification that provides instructions about applying the updated files and registry keys in the software patch, service pack, or software update.

You cannot deploy an .mst or .msp file alone by using software installation and maintenance; the file must modify an existing Windows Installer package.

Distribution Phase

After you have established your plan for software management and acquired the software and licenses for the software that you want to deploy, prepared package formats for the software, and performed any customization on the packages, the next step is to copy the software and your Windows Installer packages to software distribution points.

Software Distribution Points

Software distribution points are network locations from which users are able to get the software that they need.

The main task of the distribution phase is to get the software to the software distribution points. Figure 23.1 illustrates this process.

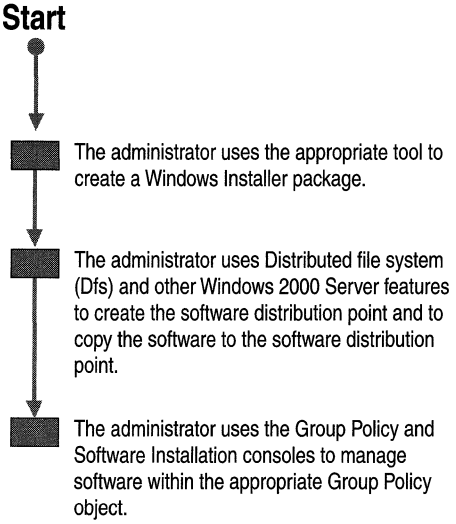


Figure 23.1 Distribution Phase of Installation Process

▶ **To copy software to a software distribution point**

1. Create the folders for the software on the file server that is going to be the software distribution point, and make the folders network shares. For example:

   \\<*servername*>\<*sharename*>

2. Replicate the software to the software distribution points by placing or copying all necessary files and components to a distribution share or shares. Place each application (the package and all related installation files) in a separate folder on the software distribution point.

   Note Some software supports an administrative setup by using special commands to facilitate the creation of a software distribution point, which you can use to enter the software key once for all users and the server and share name for the software distribution point to which the files are copied.

3. Set permissions on the folders so that only administrators have read and write permissions, and users can only read the files from the software distribution point folders and shares.

Note Windows 2000 software installation and maintenance does not address the distribution phase directly; however, you can choose to use other Windows 2000 services, such as the Distributed file system (Dfs), to manage the software distribution points during the distribution phase. For more information about managing software distributions points by using Dfs, see "Distributed File System" in this book.

Important Software licenses are required for software that is written by independent software vendors and distributed by using software distribution points. It is your responsibility to limit the number of users who can gain access to software through software distribution points to the number of licenses that you have on hand. It is also your responsibility to verify that you are working within the guidelines that are provided by each independent software vendor.

Distributing Windows Installer Packages

Because Windows Installer is an operating system–based service, it does not matter how you actually get Windows Installer packages to the client computer. For example, if you want to distribute your application on a CD, you can burn a CD that contains the Windows Installer package, the necessary files for the software, and an Autoplay.inf file to call Windows Installer to determine whether the software is already installed. If the software is not found, it is installed from the CD.

If you are deploying software to many users in a large organization that is using Windows 2000 Server and Active Directory, and all of the workstations are using Windows 2000 Professional, you can use the software installation and maintenance feature of Windows 2000 to deploy the Windows Installer–based software. You can also use a combination of software installation and Systems Management Server to enable large-scale deployments. Table 23.5 illustrates what each method can do.

Table 23.5 Microsoft Systems Management Server and Microsoft Windows 2000 Software Distribution

Method	Software Installation	Systems Management Server
Distribution	A distribution mechanism is not built into Software Installation. It relies on other services, such as Dfs to provide load balancing. Not recommended for use across slow network links.	Full distribution support including compression, bandwidth throttling, priorities, checkpoint restart, scheduling, load balancing, and replication.

Deploying Software by Using Software Installation

Windows 2000 software installation uses a *pull model* for software deployment, which makes software available to users as it is needed. Applications are fully installed, when a user selects a file by choosing the file name extension of an application or chooses to use a user-assigned application for the first time. However, for a satisfactory end-user experience software installation and maintenance usually requires a high-speed local area network (LAN) connection between the client computer and the distribution server where the installable software is held.

Deploying Software by Using Microsoft Systems Management Server

Microsoft® Systems Management Server version 2.0 supports a robust distribution model that you can use with Windows 2000 software installation and maintenance. You can use SMS to analyze your network infrastructure for the distribution phase and then use Windows 2000 software installation and maintenance for the targeting and installation phases of software installation.

Systems Management Server is a particularly useful tool if you are deploying software to many users in a large organization. It includes desktop management and software distribution features that significantly automate the task of upgrading software on client computers.

Systems Management Server uses a *push model* for software deployment, which you can use to coordinate and schedule software deployments, even arranging for off-hours distribution and installation, and to plan a single or multiple-phase rollout of software. It provides you with the ability to control and synchronize software deployments over multiple sites, helping to reduce compatibility issues that might otherwise occur.

The following are some of the areas where you might want to supplement software installation and maintenance by using Systems Management Server.

Use Systems Management Server for Non-Windows 2000–based Clients

Systems Management Server can distribute Windows Installer–based software to Windows 2000 Professional, Windows NT version 4.0, Microsoft® Windows NT® version 3.51, Windows 98, and Windows 95. You cannot centrally manage the non-Windows 2000–based computers with Group Policy settings; but with Systems Management Server, these computers benefit from the capabilities that are built into Windows Installer, such as self-repairing applications.

Use Systems Management Server to Deploy Software over Slow Links

By default, software installation and maintenance does not operate over slow network or dial-up connections. If you need to deploy software to users who can connect only over slow network links, such as mobile users, Systems Management Server provides options for doing this.

Use Systems Management Server for Software Licensing and Metering

Software installation and maintenance technology does not have the ability to license or meter software.

Use Systems Management Server to Identify Computer Configurations

Before you distribute a managed application, you can use Systems Management Server to determine current computer configurations to make sure that the appropriate computers have the necessary system requirements to run the application.

Note Systems Management Server and Software Installation are not intended to be used in isolation. Organizations with large change and configuration management issues might want to consider a combination of both products to address their needs.

Using Remote OS Installation

The Remote OS Installation feature of Windows 2000 is designed specifically to deploy Windows 2000 Professional. It can also be used to deploy a Windows 2000 Professional operating system with preinstalled software, such as a company's standard Windows 2000 Professional–based desktop image and its base applications. Remote OS Installation can be used to install or reinstall the operating system or desktop image on computers for which the configuration is unknown or even nonoperational.

Remote Installation Services (RIS) is the software service that allows a user to install Windows 2000 Professional from a RIS server with minimal interaction. RIS servers can provide unattended system setup, as well as fast recovery and configuration of remote-boot Preboot Execution Environment (PXE)–enabled network-compatible client computers.

By using IntelliMirror and Remote OS Installation together, you can reduce the costs incurred in setting up new computers that are running Windows 2000 Professional or you can provide disaster recovery from computer failures. You can provide dynamic configuration and repair, making it easy for users to log on anywhere in the network and have access to their applications and data.

For more information about setting up a Remote Installation Services Server, see "Remote OS Installation" in this book. For more information about IntelliMirror, see "Introduction to Desktop Management" in this book.

Remote OS Installation, software installation and maintenance, and Systems Management Server each perform a complementary range of change and configuration management tasks. Table 23.6 shows how each can be used separately or in combination in order to provide specific management functions:

Table 23.6 Management Function and the Deployment Service to Use

Management Function	Remote OS Installation	Software Installation	SMS
Install operating system and standard desktop images	X		
Manage persistent software settings		X	
Basic disaster recovery	X	X	
Manage non-Windows 2000–based environments			X
Inventory, advanced deployment, and troubleshooting			X
Change and configuration management	X	X	X

Targeting Phase

The main task of the targeting or scope-of-management phase is to determine the software needs of your users. During the preparation phase, you analyzed the groups to whom you need to deploy software. In the Distribution Phase, you established the software distribution points where the software is going to be located. (You need to know where the software files are located in order to complete the targeting phase.) In this phase, you do the following:

- Create and modify Group Policy objects in order to efficiently manage software for each individual and group.
- Create a pilot to test how you want to assign or publish software to users or computers.

Manage Your Software

By using the Software Installation snap-in, you can assign software to users or computers and publish software to users managed by a Group Policy object that is associated with an Active Directory container. Because one Group Policy object can manage many users or computers, you can use Group Policy and security descriptors to refine software management to a smaller subset of users or computers. Software management includes the following:

- Using the Group Policy snap-in to add or edit Group Policy objects.
- Assigning or publishing software for users and computers.
- Configuring software for management by associating transforms or customizations and creating upgrade relationships.

Add or Edit Group Policy Objects

As Figure 23.2 illustrates, in a Group Policy object, you can manage software for computers by selecting the Computer Configuration node or you can manage software for users by selecting the User Configuration node.

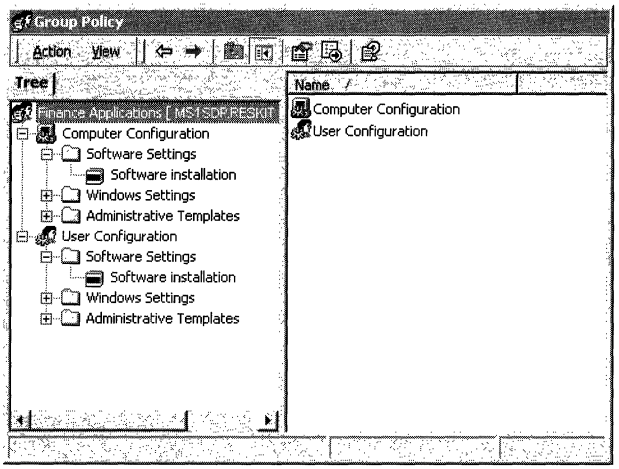


Figure 23.2 Software Installation User and Computer Configuration

Computer Configuration Contains software settings that apply to the computer and therefore all users who log on to the computer.

User Configuration Contains software settings that apply to users regardless of which computer they log on to.

The Software Installation snap-in that you use to assign and publish software is located in the computer and user configuration nodes, as illustrated in Figure 23.2.

Assign or Publish Software

Given that software can be either assigned or published and targeted to users or computers, you can establish a workable combination to meet your software management goals. Table 23.7 details the different approaches.

Table 23.7 Assigning or Publishing Software

Scenario	Publish	Assign (User)	Assign (Computer)
When the administrator deploys the software, it is available for installation after:	The next user logs on.	The next user, to whom this application's Group Policy applies, logs on.	The next time that the computer is started.
The software is fully installed:	By the user from **Add/Remove Programs** in Control Panel.	By the user from the **Start** menu or a desktop shortcut.	Automatically when the computer is started.
If the software is not installed and the user opens a file associated with the software, is the software installed?	Yes, if Auto-Install is selected.	Yes.	Does not apply. The software is already installed.
Can the user remove the software by using **Add/Remove Programs**?	Yes, and they can choose to install it again by using **Add/Remove Programs**.	Yes, and the software is available for installation again from the typical installation points.	No. Only the local administrator can remove the software. However, the user can repair the software.
Supported installation files:	Windows Installer packages and .zap files.	Windows Installer packages.	Windows Installer packages.

Published Software

When you publish software for a user, there is no evidence of the software on the user's desktop. That is, published software does not appear to be installed on the user's computer. There is no Windows Installer advertisement information about the software on the computer either in the registry or as shortcuts on the desktop or the **Start** menu. The user can install the published software by using **Add/Remove Programs** in Control Panel.

When a users clicks **Add/Remove Programs** and selects **Add New Programs**, a list appears of all of the categories for the available software and, within their specific categories, the software that is published for them. Users can install only software that you have determined they need to perform their jobs.

If there is a large amount of software available, users can quickly locate the software that they need based on the categories that you have predefined. For example, a user might see software sorted by Administration, Everyone, Finance, Sales, and Shipping categories. If the user is a member of the sales team, he or she installs applications from the Everyone and Sales categories. The user would not be able to see or to install applications from the Administration, Finance, or Shipping categories because he or she is not a member of these organizational units. For information about creating categories, see "Categorizing Applications" later in this chapter.

Figure 23.3 shows the installation process for published applications.

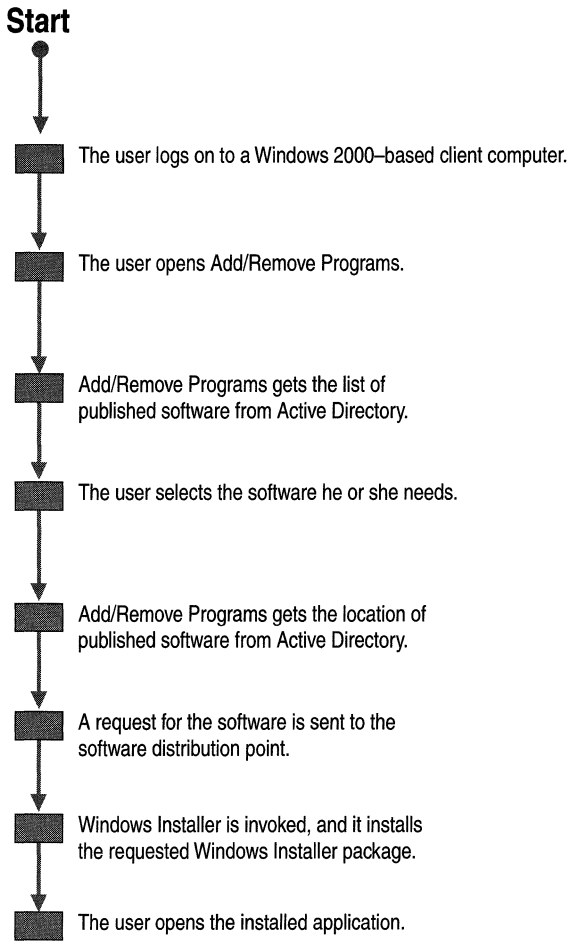


Figure 23.3 Typical Installation of Published Applications

After the user installs a published application, it behaves like an assigned application until the user removes the application by using **Add/Remove Programs** or until you remove the application.

Assigned Software

You can assign software to either users or computers.

User assignment is most powerful for software that users require to perform their jobs. Software that is assigned to the user appears to be installed on any computer to which the user logs on. When called by WinLogon, the Application Management extension to Group Policy advertises the application in the local computer's registry and as an item on the **Start** menu or a shortcut on the desktop. The user installs the assigned application by selecting the application from the **Start** menu.

Figure 23.4 shows the installation process for user-assigned applications.

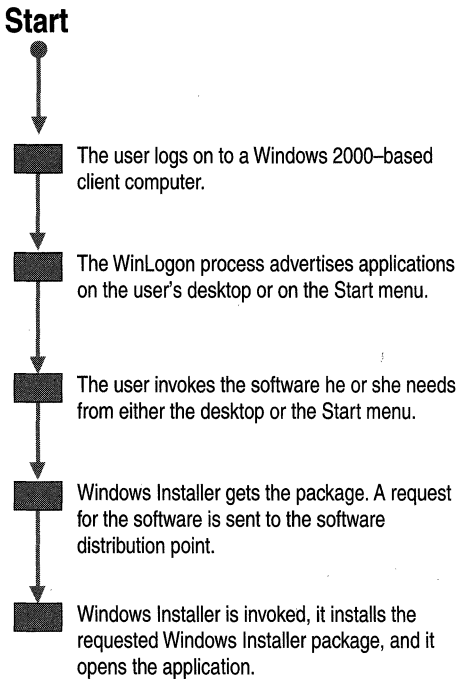


Figure 23.4 Typical Installation of User Assigned Software

You can also assign an application to a computer. In this case, the application is installed for all of the users who use the computer the next time that the computer restarts, which is the next time that the Group Policy for the computer is applied.

Note If several users use one computer, and everyone who uses the computer uses a particular application, that application might be a candidate for assignment to the computer.

Assigned software is resilient. For example, if a user removes the application by using **Add/Remove Programs**, the advertisement information is reapplied to the user or to the computer, and therefore, it is reinstalled the next time it is selected or the computer is restarted.

Configure Software for Management

Windows 2000 software installation and maintenance's scope of management is defined by Group Policy, which uses Active Directory. A Group Policy object can contain several settings that affect how an application is installed, managed, and removed. By using the Software Installation snap-in, you can configure the deployment options for an individual package or for a group of software applications.

Individual Package Deployment Options

For each application, there are several settings that can be managed in any application's properties dialog box on the **Deployment** tab. Figure 23.5 shows the **Deployment** tab for Microsoft® Outlook® 2000.

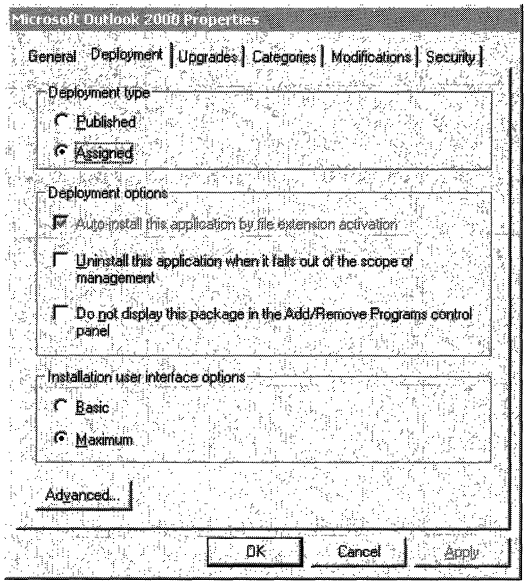


Figure 23.5 Application Deployment Properties

Table 23.8 describes the options that are available on the **Deployment** tab of an application's properties dialog box.

Table 23.8 Application Deployment Properties

Area	Option	Function
Deployment type	Assigned or Published	Select to assign or publish the application. This has no immediate effect on users who have already installed the application, but it might affect them in the future. For example, if an application is changed from assigned to published, that application is no longer re-advertised if a user removes it.
Deployment options	Auto-install this application by file name extension activation	Select this option so that a published application is installed automatically when a user double-clicks an associated file name extension and software associated with that extension is not already installed on the computer.
Deployment options	Uninstall this application when it falls out of the scope of management	Select this option so that when the Group Policy object that deployed an application no longer applies to a particular user or group of users, the application is removed.
Deployment options	Do not display this package in **Add/Remove Programs** in Control Panel	Select this option to prevent the application from appearing in **Add/Remove Programs**. If a user double-clicks an associated file name extension or gains access to an advertised shortcut, the application is installed. This option assists you in running a pilot with new software.

(continued)

Table 23.8 Application Deployment Properties *(continued)*

Area	Option	Function
Installation user interface options	Basic or Maximum	Select one of these options to determine how much information the user sees during installation of the application. **Basic** provides only error messages and progress bars. **Maximum** shows the user any information about the installation that the creator of the package provides.
Advanced button	Ignore Language Remove previous installations of this product Advanced diagnostic information	Select to ignore language when assigning or publishing a package. Select to remove previous installations of this product for users if software installation and maintenance finds the software installed by any means other than Group Policy–based Software Installation. This button also displays diagnostic information including product code, deployment count, and script name.

Auto-Installation

On the **Deployment** tab, as displayed in Figure 23.5, you can choose to Auto-install an application by file name extension activation. Without regard to whether an application is assigned or published, users can force an application to be installed by invoking a document or file that is associated with the application.

It is common for users to receive documents as attachments in e-mail messages without having the software associated with the attachment installed on their computer. When given the options, in most cases, users do not know which file type to choose in order to successfully open the attachment. With Windows 2000 software installation and maintenance, you can configure the appropriate software to install automatically.

Figure 23.6 shows the process of software being installed by document invocation.

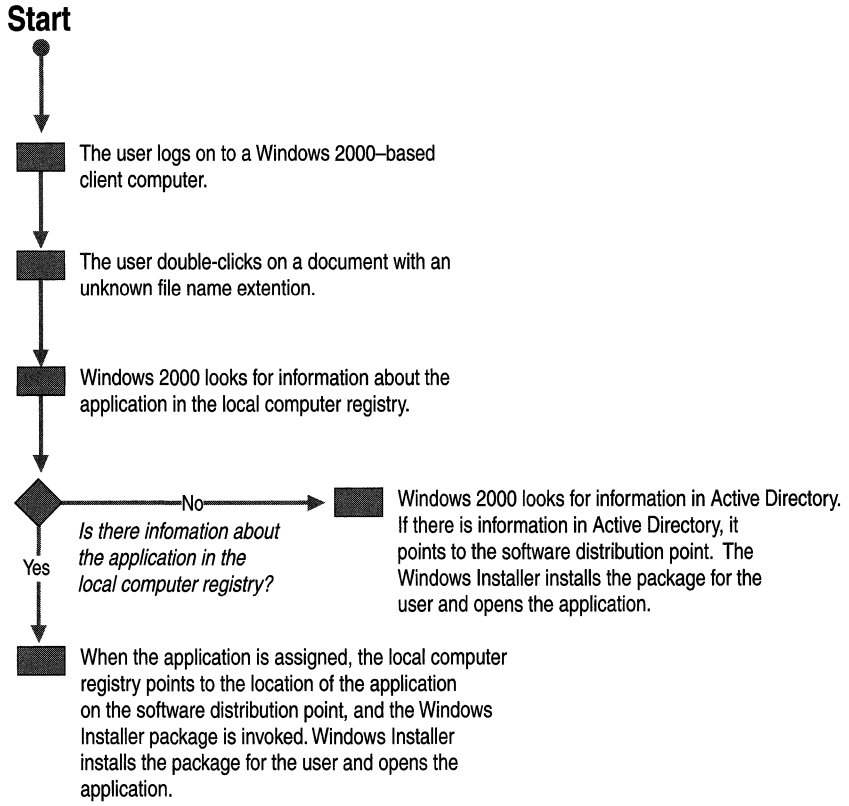


Figure 23.6 Installing Software by Document Invocation

A user double-clicks an unknown document or attachment to open it. If there is an assigned application advertised on the user's computer, but the application has not yet been installed, Windows Installer installs the application and opens the document.

If there is no application for that document type either installed or advertised on the computer, Windows 2000 Professional looks in Active Directory for the published application that is associated with the attachment file name extension.

If the application is published, Windows Installer uses the advertisement script for the application in Active Directory to install the application and then opens the document for the user.

Important If you have more then one application assigned for the same file name extension the last one installed on the local computer is the application that opens the file.

Prioritizing File Name Extensions

In the Group Policy object, you can select a file name extension and configure a priority for installing on-demand applications. An application is installed on demand when a user selects a file with a file name extension for a published application that is not already installed on the local computer. Configure the file name extension priority so that users who are managed by this Group Policy object always install the correct published application when the file name extension is selected. The first application that is listed in the Software Installation snap-in on the **File Extension** tab is the application installed in association with the extension.

▶ **To configure the file name extension priority**

1. Open the Group Policy object in which you want this setting to apply.
2. Click the applicable **User Configuration** or **Computer Configuration node.** Click **Software Settings**, and then click **Software Installation.**
3. Right-click and select **Properties**
4. On the **File Extension** tab, in the **select file extension** drop-down list, select the file name extension.
5. In the **Application precedence** list, move the chosen application to the top of the list by using the **up** or **down** buttons, and then click **OK**.

Note File name extension associations are managed on a per-Group Policy object basis. Changing the priority order in a Group Policy object affects only those users who have that Group Policy object applied to them.

Defining Settings for a Group of Applications

You can also use the Software Installation snap-in to determine the settings for a group of applications. To manage several applications at one time, use the **General** tab in the **Software Installation Properties** dialog box, shown in Figure 23.7. You can use this tab to globally define the default settings for the new packages within the Group Policy object.

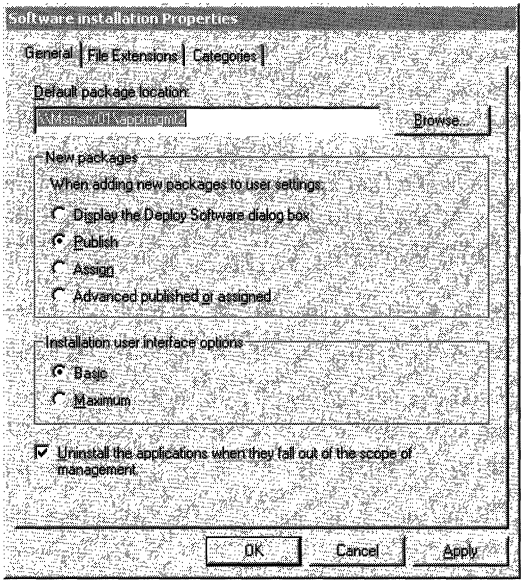


Figure 23.7 Software Installation Properties Dialog Box

Table 23.9 describes the options available on the **General** tab of the **Software Installation Properties** dialog box. You can use the **General** tab to define the default software distribution point where packages are stored and to determine how you want to deploy the new packages.

Table 23.9 Software Installation Properties

Area	Option	Function
Default package location		Defines the default software distribution point.
New packages	Display the Deploy Software dialog box	Displays the Deploy Software dialog box for each new package so that you can choose to either assign, publish, or set advanced assign and publish properties for each package.

(continued)

Table 23.9 Software Installation Properties *(continued)*

Area	Option	Function
	Publish	Select this option to add several packages when all of the packages are to be published.
	Assign	Select this option to add several packages when all of the packages are to be assigned.
	Advanced published or assigned	Select this option to add transforms and modifications to the package.
Installation user interface options	Basic or Maximum	Select one of these options to determine how much information the user sees during installation of the application. **Basic** provides only error messages and progress bars. **Maximum** shows the user any information about the installation that the creator of the package provides.
Uninstall the applications when they fall out of the scope of management		Removes the applications when the user is removed from the scope of management by this Group Policy Object.

The **Advanced published or assigned** option might be the most common selection because it allows you the most control over the package properties.

Note It is recommended that you begin by deploying and testing one application at a time in a pilot environment.

Important Use the **Uninstall applications when they fall out of scope of management** option carefully. If you assign an application to computers in a Group Policy object associated with an Active Directory site and select this option, a mobile user whose computer is managed by the Group Policy object might lose applications when the user moves to a new site. For example, if a user with Microsoft® Office assigned to his or her computer travels to another site for business and Office is not assigned to computers in the new site, the application is removed when the user starts his or her computer in the new site.

Categorizing Applications

You can organize assigned and published applications into logical categories to make it easier for users to locate the appropriate application in **Add/Remove Programs**.

Windows 2000 does not have predefined categories. In the Software Installation snap-in, you can create and modify categories. The categories that you establish are per domain, not per Group Policy object. You need to define them only once for the whole domain. To avoid conflicts, it is recommended that only one administrator create categories for the organization.

Note When you are assigning or publishing software, and you associate an existing category with the Windows Installer package that is being either assigned or published, you are working on a per-Group Policy object basis.

If you have a large amount of software to manage, define categories so that it is easy for your users to recognize their applications. Create categories that reflect your organizational structure, as when users in a department use a common set of applications. For example, a category named Finance might have Microsoft® Excel and other accounting applications. You can also have categories that reflect job functions. For example, a category named Project Managers might include Microsoft® Project, Excel, and other project management applications. Or you might decide to classify all applications by the type of software. For example, a category named Presentation Tools might include Microsoft® Publisher, PowerPoint, and other graphics programs that supported by your organization. You can also define broad categories like line-of-business tools, site-licensed applications, and beta applications.

Modifications

You can determine some of the application properties, such as modifications, only when the software is first deployed, and you can change some of the properties, such as whether or not the software is visible in **Add/Remove Programs**, later by modifying the package properties.

For more information about updating software by using patches and upgrades, see "Updating Software by Using Patches and Upgrades" later in this chapter.

Targeting Software for Multilingual Users

Multinational corporations share information and collaborate on a worldwide basis. Up to this point, the operating system support and the files that you generated by using an application in one language were not necessarily compatible with a file that was created by using the same application in a different language.

Windows 2000 Server makes the process of deploying and supporting Windows 2000 across national and language boundaries easier and more flexible. The multilanguage version of Windows 2000 was designed with a single worldwide executable file that supports most European and East Asian languages, as well as languages such as Arabic and Hebrew, which are read right-to-left. Instead of deploying a different localized version of Windows for each language that your organization needs, you can deploy Windows 2000 with the MultiLanguage Pack to all international users. The worldwide executable file makes multilanguage setups much easier because there are few differences in setup routines, registry settings, and component configurations. The single worldwide executable file also streamlines the development and deployment of customized solutions.

Note Windows 2000 supports many languages. Before managing software, verify that your applications run on an operating system that is running in a native language mode.

Software Installation and Maintenance and Multilanguage Issues

Although a Windows 2000 Professional–based computer has many different locale and language settings, for the purposes of software installation and maintenance, only the system locale matters. The system locale affects the code page that is installed, and code page availability, more than any other factor, affects how a language-specific or multilanguage application behaves. For additional information about locale and language settings, see the Locales and Language link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/.

Applications typically support one language. This supported language, or the *product language*, is expressed in the Windows Installer package for the application.

To determine whether or not to install an application on a Windows 2000–based computer, software installation and maintenance checks to see the following:

1. If the **ignore language** parameter is set in the Software Installation snap-in for the managed package. If **ignore language** is set, the package is either advertised or installed without regard to whether or not the system locale and product language match.

2. If the system locale and the product language match. If they do, the package is either installed or advertised.
3. If the system locale is any value, and the product language is either neutral or English. If so, the package is either installed or advertised.

Important Users roaming between computers with different locales might produce unexpected results.

Use caution when deploying two versions of an application with different product languages (for example, an English and a German version of the same application) in the same Group Policy object if the two applications have the same product code. If the two applications have the same product code, only one of them is installed for users or computers. If you need to make the same application available to users in multiple languages, the different language versions of the application have to have different product codes.

Pilot Program

Design a pilot program to test, troubleshoot, and prepare software for deployment. The pilot also establishes an ongoing system for testing and maintaining applications, including how to work with patches, software upgrades, and software removal, and the creation of a software lab where pilots and testing can be performed on an ongoing basis.

Software deployment affects both client and server performance. Within the pilot, consider strategies for optimizing software deployment on your network and on client computers. It is important that you clearly understand the current arrangement of servers and client computers throughout your network before you attempt to manage operating system updates and applications. For example, you need to consider the impact on the network performance of allowing software to be downloaded over slow network connections. You also need to carefully define who can install and update applications. These and other considerations greatly affect the number of support calls that you receive.

You can manage software evaluation or a pilot in the following ways:

- Evaluate the software outside the corporate managed environment—for example, in a laboratory or test network environment.
- Create a Group Policy object to manage the evaluation, and assign or publish the application to users or computers that are managed within that Group Policy object.

- Modify the security descriptors on an existing Group Policy object or on the assigned or published package to control who can install the package for evaluation.
- By assigning or publishing software to your testers, they can test all methods to verify that you have the results that you need for the group of users or computers to receive the application.

Note In order to easily manage who has the application and not to affect actual users if a package is improperly deployed, assign or publish software to users grouped within a test Group Policy object, rather than try to manage the pilot by controlling security descriptors.

To ensure that potential application problems are resolved before you deploy new software or upgrades, it is best that the test manager begin early to develop a plan for testing and maintaining software. For more information about setting up a pilot and testing existing applications, see "Conducting Your Windows 2000 Pilot" and "Testing Applications for Compatibility with Windows 2000" in the *Deployment Planning Guide*.

Software Installation User Scenarios

It is recommended that each application be evaluated from the perspective of the users who require the application. The following are some questions to consider:

- Does everyone in the company require access to the software? Or does only a select group of users need to use it?
- How often does the software require updates?
- Do you need to customize the software for some divisions in the company?
- Is the software an in-house application or software that is provided by an independent software vendor?
- How often do you need to update user templates or custom files that you want to manage by using the software?

The answers to these questions and other information about your users and your organization can help determine the best way to conduct your pilot and manage software. For example, if you have an in-house application that is updated monthly and that is required only by the finance division, and all of the users in this division are in the finance organizational unit, you might want to assign the application to the finance organizational unit and manage updates by packaging them as a separate Windows Installer package that you also manage by assigning within the same organizational unit.

Within the pilot, test all conceived manners of deploying software. If there are additional templates or files that you want to distribute with an application to only the Help desk personnel, place the templates and files in a separate Windows Installer package. Then assign or publish both the application and the templates package to the Help desk organizational unit. When there are new templates or files, or changes to these files, you can upgrade the existing templates package. You do not have to reinstall the entire application. During the initial deployment, if you add the templates to the application package, the entire application would need to be upgraded when a new template or file is required.

The following scenarios explore some of the additional issues that can arise when deploying software. They can help you when you are working on the various pilot tests that you need to establish, based on the users with whom you are working.

These scenarios highlight some best practice tips that Windows 2000 software installation and maintenance supports. Although it is hard to provide a single answer that is going to work for all organizations and all situations, these scenarios illustrate some of the more commonly encountered situations. They include recommendations on how best to use the software installation and maintenance feature for specific types of users.

Roaming User or Floating Worker

In many organizations, some users move or "float" from one location to another in the organization to perform their jobs. Some receptionists are roaming users. A key point about the roaming or floating worker is that although they log on to different computers to perform their jobs, these computers are usually connected by a high-speed connection or a LAN connection.

Note By default, software installation and maintenance policy settings are not applied over a slow link. So roaming or floating users who connect to their organization's network by slow links might not see changes to their software. Group Policy can change this. For more information about slow links and Group Policy, see "Group Policy" in this book.

When working with roaming users, there are two ways to assign applications—to the users or to the computers. For example, if there is an application that all of the receptionists use, assigning that application to the computers so that it is already installed on the computers that the receptionists use might make the most sense. But, this might also be a scenario where you need to assign the software to these users. Then when a receptionist relieves another receptionist, he or she logs on to the computer and sees the applications assigned to him or her (because the application is advertised). The application is only installed for the user if they actually run the application.

In another example, if you have users who roam between two computers with different locales (languages), manage the software carefully or the users might not see the set of applications that they expect to see—for example, if one computer is running the U.S. English language version of Windows 2000 and one is running the German language version Windows 2000, some applications, whether they were assigned or published, might not work correctly in both languages. If you are supporting roaming users between languages, test the applications thoroughly to determine whether there are any problems.

You can use the Software Installation snap-in to manage software that is designed for single or multiple languages. You need to ensure that software is managed in the way that users expect to have it work on their computers.

Mobile Users

In many organizations, some users travel extensively to perform their jobs. For example, sales personnel often spend more time at customers' offices than their own offices. A key point about mobile workers or traveling professionals is that although these users log on to the same computer, their computers sometimes connect through a high-speed line and sometimes through a low-speed (or dial-up) line.

You might decide that it is best to publish software to these users, which ensures that any customization (transform) that applies to the software installs the software locally on the users' computers. This might be more effective than leaving the software or the feature to either install on first use or to run from a network location.

Additionally, you want to allow mobile workers to keep some software available on local media while they are traveling. For example, for mobile professionals that make a lot of presentations, bids, and quotes, it might make sense for you to allow them to have Office on a CD. Then if they need the source files to install or repair a feature, they can use the source media.

Note If users are having trouble staying connected when they download software, verify that the connection speed and Group Policy settings are set appropriately. With Windows 2000, you can determine the connection speed that is considered to be a slow link.

Shared Computers

In many organizations, users share computers. This is quite common on the factory floor or in a business such as a bank, where tellers have the same job function but might work at a different counter (and therefore computer) on different shifts. In these environments, the software is often task-based, and although users change, the software does not (however, the software might track who is logged on). You might want to group these users or computers so that you can manage them from a single Group Policy object and then use Software Installation to assign software to the computers (either by the Computer Configuration or the User Configuration of the Group Policy object namespace). Thus, the software is going to be available for every user of that computer.

Be aware that software that is assigned to the computer is installed when the computer restarts. If computers restart between shifts, the time that it takes to install the software (the first time after the software is assigned to the computer or is upgraded) might affect the total startup time of the computer. Again, this increase in startup time occurs only if new software has been assigned or the existing software is upgraded. You can use this feature to manage the software and essentially reinstall the software environment every time the computer restarts.

Another place that computers are shared is a computer lab or classroom environment where users share computers typically for a short period of time (for example, while they are taking a class). This scenario differs from the previous shared computer scenario in that each user might either use the same software as the previous user (they are in the same class) or they might use different software. It is also likely that these computers do not move. Therefore, this might be a case in which using a site to manage software makes sense; although grouping the computers into a single organizational unit would also work. Choose the method that gives you the right level of control for applying Group Policy and software installation and maintenance. Depending on the requirements, you might still decide to assign software to the computer. This can work well if the software is written to keep user information (such as configuration information and saved files) separate from software information (such as executable files). Another way to manage this environment is to assign software to the user (each student has his or her own identifier and must log on) so that each user gets the software that they need for their training.

Group Policy can also limit the changes that users can make to the desktop and other settings on these computers so that you can tightly manage each computer.

Note It is recommended that you consider using Remote OS Installation in these shared computer environments. Then, if you ever have to rebuild the lab, you can do so in a quick and efficient manner.

Staging Computers

You can use Windows 2000 Remote OS Installation to stage and standardize your computers by installing the operating system and any standard applications at the same time.

When you first acquire computers, a lot of time is spent preparing them for your users. Without regard to the source of the computers or how the original equipment manufacturer (OEM) has prepared them, most organizations format the hard disk drives of the new computers and then configure them according to their organization's standard configuration.

The best way to do this is to prepare a standard configuration on a typical computer, including the application software and then make an image on the remote operating system installation server. For example, if you want to bring in new computers and customize it to include Windows 2000 Professional, virus protection software, and Office 2000, use the following steps:

1. Setup and configure Remote OS Installation.
2. Use Group Policy to create the appropriate Group Policy objects to manage the computers in the organization.
3. Create a software distribution point for the virus protection software and Office 2000.
4. Create a transform to customize Office 2000 in the way that you want it configured.
5. Use the Software Installation snap-in to assign the virus protection software and Office 2000 to the computers in the appropriate Group Policy object.

Note If you assign the software to the user, the computer-assigned version can be removed, and the user version advertised and then installed on first usage. Although the end result is that the user has the software, this process might take additional time.

---

6. Install Windows 2000 on the source computer (the computer does not have to have exactly the same hardware, but it must use the same Hardware Abstraction Layer [HAL] as the client computers on which the image is going to be installed), and configure the operating system in the way that you want it.

7. Add the computer to the Active Directory container where it is going to reside when it is deployed. This container has the Group Policy object with the virus protection software and Office 2000 assigned to the computer that is associated with it. (Care must be taken to configure this computer with software from the Group Policy object where the computer is going to reside; otherwise, the software might be removed and reinstalled if a different Group Policy is applied to the computer when it is deployed.)

Note When you install Office 2000 as part of a Remote Installation Services (RIPREP) image, you must turn off 8.3 name creation. Change the value of the **NtfsDisable8dot3NameCreation** registry entry from 0 (default) to 1 in order to turn off 8.3 name creation. **NtfsDisable8dot3NameCreation** is in:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\FileSystem

See the procedure "To turn off 8.3 name creation."

---

8. Start the computer, and software installation and maintenance installs the virus protection software and Office 2000 (software assigned to the computer installs when the computer starts).
9. After Windows 2000 Professional, the virus protection software, and Office 2000 have been installed, use the RIPrep tool of Remote OS Installation to build a Remote OS Installation image. Place this image on the RIS server.

▶ To turn off 8.3 name creation

1. On the **Start** menu, click **Run**.
2. Type **regedt32.exe** or **regedit.exe**, and then click **OK**.
3. In the registry editor, navigate to:

   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\FileSystem
4. Select the **NtfsDisable8dot3NameCreation** entry.
5. To turn off 8.3 name creation, change the value of the **NtfsDisable8dot3NameCreation** registry entry from 0 (default) to 1. In Regedit.exe right-click the entry, and then click **Modify**.

   –Or–

   In Regedt32.exe, click the entry, click Edit, and then click the appropriate menu choice.

6. Close the registry editor.

Caution Do not use a registry editor to edit the registry directly unless you have no alternative. The registry editors bypass the standard safeguards provided by administrative tools. These safeguards prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your system. Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. To configure or customize Windows 2000, use the programs in Control Panel or Microsoft Management Console (MMC) whenever possible.

After this image is available, a user who receives a new computer that supports Remote OS Installation only has to connect the peripherals (keyboard, mouse, monitor), connect to the network, turn on the computer, and then proceed through the Client Installation wizard.

The computer finds the RIS server and then downloads the Windows 2000 operating system and the software. When the computer restarts after remotely installing Windows 2000, Software Installation detects that the virus protection software and Office 2000 are already on the computer and only updates the advertisement information, which takes a few seconds.

Note When the user logs on to the computer and selects the first Office 2000 application, they see Windows Installer start because it needs to complete a small amount of user configuration. This occurs whether Office 2000 is assigned to the user, assigned to the computer, published, or installed by using Remote OS Installation.

Although some scenarios lend themselves to assigning the software to the computer, and others to assigning it to users, avoid assigning the same software to both users and computers. This applies to assigning the software within one Group Policy object or assigning the software to users in one Group Policy object and to computers in another Group Policy object if both Group Policy objects can be applied to a single user when Group Policy is applied.

For more information about installing Microsoft Office 2000, see the Microsoft Office Resource Kit link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Installation Phase

The main task accomplished during the installation phase is managing the state of the software on the computer. Software can be installed, modified, updated to a new version, or removed.

- Installation involves the initial installation of the software, including copying the necessary files, configuring the registry, and creating any shortcuts that allow users to find and use the software.
- Modification involves adding or removing features after the initial installation. For example, after the initial installation of a word-processing application, a user might decide to install the spelling checker feature.
- Repair involves keeping the software in a working state without regard to what happens on the computer. For example, if a user deletes the executable file for a spreadsheet application, and then chooses the application from the **Start** menu, the executable file is reinstalled automatically thereby repairing the software.
- Removal involves completely and safely removing the software from the computer when it is no longer needed, including the removal of all the files, registry entries, and shortcuts.

Fundamentally, the Installation Phase occurs on Windows 2000 Professional desktops. In order to better understand the installation process, the Windows 2000 Professional components related to software installation and maintenance are listed in Table 23.10.

Table 23.10 Windows 2000 Professional Components

Component	Overall Function	Software Installation and Maintenance Function
Computer Start-up	Loads the operating system, shell, and other programs.	Applies computer Group Policy so that computer-assigned software is installed.
WinLogon	Allows a user to log on to his or her computer.	Applies user Group Policy so that assigned applications are advertised.

(continued)

Table 23.10 Windows 2000 Professional Components *(continued)*

Component	Overall Function	Software Installation and Maintenance Function
Application Management Extension (appmgmt.dll)	Client-side extension of software installation.	Communicates with Active Directory, Group Policy, and Windows Installer to assign or publish software.
Windows Installer	Enables the client for managing software.	Advertises, installs, repairs, and removes software.
Add/Remove Programs in Control Panel	Allows users to manage software on their computer.	Lists published and assigned applications so that users can install, modify, and remove software from their computers.

For more information about Windows 2000 Professional enhancements, see the *Microsoft® Windows® 2000 Professional Resource Kit.*

Note Windows 2000 software installation and maintenance uses Windows Installer during the installation phase, as it is the best method for deploying software and maintaining software in a managed state. Windows Installer is a base service of the Windows operating system; therefore, it is available on Windows 2000, Windows NT 4.0, Windows 98, and Windows 95.

Change Control Procedures

Good change control procedures assist in managing the impact of software installation and maintenance changes within the organization, and additional troubleshooting that might be required. It is recommended that you not assign, publish, patch, upgrade, or remove a large number of applications at any one time or session. For example, it might not be good practice to publish a large complex application such as Office, and another application at the same time. Small changes are more manageable.

Software State Information

At any time during the installation phase, you can view the current state of the software that you are managing by referring to the following information that is displayed in the Software Installation snap-in: Name, Version, Deployment state, Auto-install, Upgrade Type, Upgrading, Locale, Platform, Source, and Modifications.

Figure 23.8 shows the deployment state for the User Configuration, Software Settings for several applications.

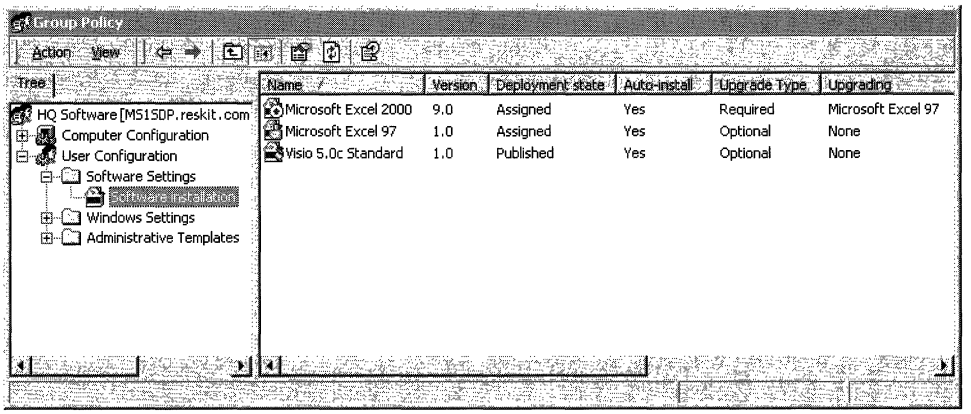


Figure 23.8 Software Settings

The deployment status is updated when a new application is deployed or an upgrade or service pack is released.

Note Systems Management Server supports other installation methods and clients. Other methods of software installation can leave computers in an unmanaged state.

Updating Software by Using Patches and Upgrades

From time to time, publishers of software provide patches or hot-fixes, service packs, and software upgrades.

Typically hot-fixes address a single problem, and not everyone has encountered the problem. The same is true of a patch, which also typically patches or fixes a single problem. You need to determine whether or not your organization needs the patch and if so, whether you are going to manage it. Usually, when you deploy patches, service packs, or minor software updates, you re-advertise the package to everyone who was granted access to the original application.

Service packs do not differ much from hot-fixes or patches. Typically service packs roll up several patches, and the patches have been tested together. Therefore, service packs are distributed less often than patches, but more often than full upgrades.

- If the service pack updates only a small number of files, it can be distributed and managed like a patch.
- If the service pack updates a large number of files, it can be distributed and managed like an upgrade.

Patching Applications

After you have tested a patch or service pack and decide to deploy it, replace the older files by copying the new files to the software distribution point.

The publisher of the software who distributed the patch or service pack either supplies a new Windows Installer package (.msi) or a Windows Installer patch (.msp). If they supply a new Windows Installer package, replace the existing package with the new one. If they supply a Windows Installer patch, use this file to update the existing package. The supplier of the Windows Installer patch includes instructions about how to use the Windows Installer patch to update the Windows Installer package.

▶ **To redeploy an assigned or published package**

1. Open the Software Installation snap-in.
2. Locate the Group Policy object that originally deployed the application.
3. Click the package name, or browse to locate the package.
4. Right-click **All Tasks**.
5. Click **Redeploy Application**.

The patched or updated files are copied to the users who have installed the software, and the software is re-advertised to everyone who was granted access to the original application.

Important Use caution when patching applications. If the product code in the Windows Installer package of the base application (the application that is already deployed) is the same as the product code of Windows Installer that is supplied with the patch, you can patch the software and then use the Software Installation snap-in to redeploy the software. If the product code in the Windows Installer package of the base application is not the same as the product code of Windows Installer that is supplied with the patch, it is recommended that you use the Software Installation snap-in to upgrade the base application with the patch.

Patches do not change the friendly name of an application. If you want the new friendly name to be displayed, you must perform an upgrade of the base application to apply the patch. For example, you have a patch that changes the friendly name of the application. The current friendly name is Microsoft Office and the new friendly name of the patch is Microsoft Office—Service Release One. If you patch an application by updating the files on the software distribution point, and then use the Software Installation snap-in and redeploy the application, the name in the Software Installation snap-in and in **Add/Remove Programs** is still the friendly name of Microsoft Office. To change the name to Microsoft Office—Service Release One, you need to perform an upgrade.

If you need to change the file name extensions that are to be associated with a managed application, upgrade the base application. For example, if you receive a patch to an application that changes the number of associated file name extensions, this includes adding new file name extensions or deleting some file name extensions, you need to upgrade the application and not only redeploy.

Upgrading Applications

Upgrades typically involve major changes to the software and usually have new version numbers. A substantial number of files might change for an upgrade. You can use the Software Installation snap-in to establish the procedure for upgrading from an existing applications to the current release.

In most instances, the publisher of the software supplies a Windows Installer package for the new version. That package defines what existing versions of the software the new package can upgrade. It also contains instructions about how to perform the upgrade—for example, which existing files can be left in place, which existing files must be deleted, and which new files need to be installed. The Windows Installer package schema and design accounts for a *declared upgrade relationship* in which one package knows which other packages it can upgrade. A declared upgrade relationship requires natively authored (rather than repackaged) applications. The Software Installation snap-in can use this declared upgrade relationship to manage upgrades.

Either a native Windows Installer package detects the upgrade candidate or you can choose the candidate, which allows, for example, one vendor's application to be replaced by another's. No special user action is required; when a user logs on next, the application is available as assigned or published.

However, initially most applications that are deployed with the Software Installation snap-in are repackaged applications, which impacts upgrades in the following ways:

- A repackaged application's Windows Installer package does not have declared upgrade relationships. Therefore, you have to manually create upgrade relationships, which the Software Installation snap-in support.
- Because the new application package (whether natively authored or repackaged) cannot detect how the existing repackaged application was installed, it is not always able to cleanly migrate from the existing application to the new application. To upgrade you have to remove the existing application, and then install (replace it) with the new application. As more and more applications become available with natively authored packages, upgrades are able to migrate from the existing application to the new application.
- During an upgrade, it might not be possible to completely remove a repackaged application. The removal of a repackaged application might leave components on the desktop, even if the component is neither shared nor needed.

Note In most cases, when a repackaged application is upgraded, the existing application is removed and the new version is installed resulting in the loss of user preferences and individual settings.

However, if you replace one native Windows Installer package with another, this problem does not occur because a native package only replaces files that need to be replaced and does not automatically remove all registry entries and other locations where user preferences might be stored.

It is recommended that you pilot or test an upgrade before putting it into production. In the pilot phase, users can be given the new version of an application and retain the original. You begin the upgrade by using the Software Installation snap-in to assign or publish the new version and define an upgrade relationship between the new and existing version. Regardless of whether the original application was assigned or published, there are two options for upgrading:

- **A required upgrade happens immediately.** Everyone who has installed the existing version of the application is upgraded to the new version. After the upgrade, users who have never installed the existing version of the application can install only the new version.
- **An optional upgrade does not happen immediately.** Everyone who has installed the software can continue to use the existing version. Users who want the new software immediately can install it from **Add/Remove Programs**. Users who never installed the existing application can install either the existing or the new version. At some point, you might decide that it is best that the optional upgrade become mandatory or required. The new version of the application works like a required upgrade from that point on.

Note At first, you might want to offer the upgrade as an option, giving users an opportunity to choose when the upgrade happens. Later, the upgrade can be required.

Figure 23.9 shows Excel 2000 as the required upgrade path for Excel 97.

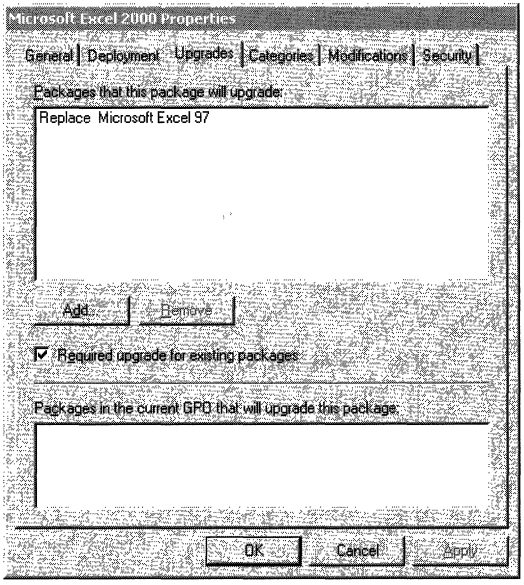


Figure 23.9 Microsoft Excel 2000 Properties to Upgrade Microsoft Excel 97

In Figure 23.9, Microsoft® Excel 2000 is the required upgrade path for Microsoft® Excel 97. If a future version of Excel is released and you add it to the Software Installation snap-in, make sure that you create an upgrade relationship between that future version and Excel 97, and between the future version and Excel 2000. If you only set the upgrade relationship between the future version and Excel 2000, users who still have Excel 97 installed must upgrade to Excel 2000 before they can upgrade to a future version. Because client computers can handle any number of active applications being chained together by required upgrades, the computer would sequentially request upgrades until it achieved the current release, which is not optimal.

Removing Software

At some point, users no longer require an application, so you decide to remove the application. The following choices, set within the Software Installation snap-in, might affect the removal of software:

- **Optional Removal.** You decide the existing software is not supported and that Help desk support is provided only for the latest version of the software.

  You can remove the software from management without forcing the (physical) removal of the software from the computers of users who are still using the software. Users can continue to use the software until they remove it themselves. In this scenario, no one is able to install the older version of the software from the **Start** menu, by using **Add/Remove Programs**, or by document invocation.
- **Forced Removal.** You decide that it is best that the software no longer be used. That is, users cannot install nor run the software. With forced removal, the software is automatically removed from a computer, either the next time the computer is turned on (when the software is assigned to the computer) or the next time the user logs on (when the software is assigned to the user).

Note When you originally deploy the software, if you want the application to be removed when a Group Policy object is no longer applicable, select the **Uninstall this application when it falls out of the scope of management** option. Use this option with caution. For more information about this option, see "Targeting Phase" earlier in this chapter.

Windows Installer Technology

Windows Installer is a client-side software deployment technology that you use to install, maintain, and remove software, including software deployed by using software installation and maintenance technology. Office 2000 is the first application that Microsoft produced with the power of Windows Installer. Additional applications from both Microsoft and other software publishers also use the Windows Installer technology. You can use Windows Installer to do the following.

Provide for complete and safe removal of software Windows Installer removes precisely the files that it should when an application is removed. It completely removes an application, including all of the files and related registry entries that are not necessary. Because it tracks applications on a systemwide basis, Windows Installer also makes software removal much safer. For example, the service does not delete a .dll file if that file is shared with another installed application.

Diagnose and repair software problems When you start an application, Windows Installer can quickly verify the existence of key files required for the application to run. If a key file has been removed or deleted, the service can repair the application. If necessary, it automatically returns to the installation source, retrieves a new copy of any deleted files, and repairs the application.

Install software in a variety of environments The Windows Installer package contains the necessary information to install the application in many different configurations, including the following:

- On different platforms
- With different sets of previously installed products and features
- With previous versions of a product
- With different default installation locations

Managing Windows Installer with Group Policy

You control many Windows Installer settings by using Group Policy so that you can configure the behavior of the Windows Installer centrally. Table 23.11 lists some of the settings that you can configure.

Table 23.11 Group Policy Settings for Managing the Behavior of Windows Installer

Group Policy Setting	Description
Disable Microsoft Windows Installer	Prevents users from using Windows Installer to install software that is not assigned or published to them. Prevents users from installing unauthorized application. However, it affects only those applications that use Windows Installer. Software that is not installed by using Windows Installer is not affected.
Always install with elevated privileges	Directs Windows Installer to use system permissions when it installs any program on the system. Extends elevated privileges to all programs and lets users install programs that require access to directories that the user might not have permission to view or change, including directories on highly restricted computers.
Disable local patching	Prevents user from using Windows Installer to install software updates or patches. Helps limit the potential harm caused by unauthorized updates that contain viruses or are not compatible with your existing software policy. Administrators can still deploy patches by updating the server.
Dimmed **Start** menu shortcuts	Allows you to use dimmed **Start** menu shortcuts for applications that are assigned to users but have not been installed. Provides a visual cue that selecting one of these **Start** menu shortcuts does not immediately open the application but starts the installation process.
Specify search order	Allows you to specify the order in which Windows Installer looks for installation files. By default, first Windows Installer searches for a network software distribution point and then for removable media, such as floppy disk drive, CD, or Digital Versatile Disc (DVD), and it then searches for an Internet location. You can use this policy to change the search order or to exclude a source location from the search order list.

(continued)

Table 23.11 Group Policy Settings for Managing the Behavior of Windows Installer *(continued)*

Group Policy Setting	Description
Specifying logging events	Allows you to specify which Windows Installer events are recorded in the Installation log (.msi.log in the temp folder). By changing the logon options, you can compile deployment related information.
Disable rollback	Prohibits Windows Installer from generating and saving the files it needs to reverse an interrupted or unsuccessful installation. Prevents Windows Installer from recording the original state of the system; as a result, Windows Installer cannot restore the system to its original state if the installation is not completed. Designed to reduce the temporary disk space required to install programs. Do not use this policy unless it is essential.

Windows Installer Package

The Windows Installer package is an .msi file that contains explicit instructions about installing and removing specific applications. The company or developer who produces the application provides the Windows Installer package .msi file and includes it with the application. If a Windows Installer package does not come with an application, you might need to create a Windows Installer package by using a third-party tool. Windows Installer technology provides the following:

- **Windows Installer Package format.** A relational database with all the information necessary to describe how to install an application. The package can contain guidelines for developers of in-house or corporate and commercial applications. The Windows Installer package file name extension is .msi.
- **Transforms and modifications.** Transforms can be used to customize the installation of a Windows Installer package at the time of assignment or publication. You can modify them at any time. Transforms are saved with the .mst file name extension.
- **Transaction-based installation service.** Service that helps to ensure that computers remain in a consistent state, even in the event of a failed or canceled installation.
- **Application programming interface (API).** The API allows applications to interact with Windows Installer to install or remove additional features of the application after the initial installation is complete.

Microsoft provides a redistributable installation program, instmsi.exe, that enables the Windows Installer installation service on Windows NT 4.0, Windows 98, and Windows 95. For more information about this installation program, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Note Only Windows 2000 Professional supports Group Policy–based managed applications.

Each package (.msi) file is a relational database that stores all of the instructions and data required to install and remove the program across various installation scenarios. For example, a package file can contain instructions for installing an application when a prior version of the application is already installed. The package file also contains instructions for installing the software on a computer where that application has never been present.

A Windows Installer package consists of multiple interrelated tables that together constitute a relational database of the information that is necessary to install a group of features.

The Windows Installer package database schema provides advantages over a text-based flat file solution as follows:

- High query performance at run time, with no data or string parsing required.
- Compact, persistent file size.

A Windows Installer .msi package contains standard summary information: Tables that describe the features that make up the application, pointers to the source files to install, and pointers to where to install the application on the client computer. The summary information stream contains some basic data about the package and the application that the package installs. This information includes the application name, the application publisher, and the version of the application as well as other basic information.

Developers can use the Windows Installer installation service and the .msi package schema to divide an application into available features and then further divide these features into the required components that make up the feature. This highly defined relationship between features and components is the architecture that allows applications to install features as they are needed. After the application is installed, a user can modify it by selecting additional features as needed; this is called "just-in-time" feature installation. Windows Installer also maintains the necessary key files and registry settings. If the necessary components for a feature become damaged, an intelligent reinstallation of only those features is possible. And the complete removal of the application is possible because of the known and tracked relationship between features and their base components.

You can store a Windows Installer package anywhere and on any media type that you want. As long as a client can get to the share, the package is available.

You can use the package format to compress all of the product's component files into one or more cabinet (.cab) files for installation from either a floppy disk, network share, or Internet location. The cabinet files can also be included in the package. When products reside in a folder hierarchy, the hierarchy is the same on the CD as it is at a distribution point and on the user's local hard disk drive by default unless the user makes changes. In such cases where the user has made changes, the package does not include product files themselves, but rather a reference to an installation point where the product files reside.

Natively Authoring Windows Installer Packages

For the best installation experience, use natively authored Windows Installer packages. Natively authored Windows Installer packages support all of the Windows Installer functions including just-in-time feature installation, feature repair, and installation with elevated privileges.

Typically, an organization can natively author a Windows Installer package for their in-house applications when they have access to all the source code, executable files, dynamic-link libraries (DLLs), and other resources. You need to have an understanding of the application, including registry entries, shortcuts, and other information that are needed for the application to run correctly.

For example, an organization has an in-house application that users can use to arrange their business travel. They have all the files for the application, and the developers understand how the application must be installed on user's computers. In this case, the organization can choose to natively author a Windows Installer package for the software. It is important that you fully understand all aspects of an application before modifying features and their relationship to other components.

For best results, create the installation package and the software simultaneously. That is, you should not create the setup as an afterthought at the end of the development cycle; instead, the installation of the software should be a part of the overall architecture of the application.

Windows 2000 can work with existing applications, but there are some distinct advantages to working with "Certified for Microsoft Windows" applications. A "Certified for Microsoft Windows" application requires more than just a Windows Installer based installation. For more information about these types of applications, see the appendix "'Certified for Microsoft Windows' Applications" in this book.

Additional Repackaging Programs

There are a few programs available to assist you with natively authoring a Windows Installer package or modifying and repackaging existing Windows Installer packages. Windows 2000 Server includes the VERITAS WinINSTALL LE (Limited Edition) repackaging tool. Other tools to support creation of packages include authoring tools from InstallShield, Wise Solutions, and Microsoft. For more information about these applications, see the following links on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources:

- VERITAS WinINSTALL LE
- InstallShield
- Wise Solutions
- Microsoft® Visual Studio® Installer

Use these Windows Installer authoring tools if you are a developer, you have access to the source code for the application, and you understand how to install the application. If you are an administrator, you have software with a non-Windows Installer based setup, and you want to use a program to repackage the application and create a Windows Installer package, you can use the third-party authoring tools.

Note The VERITAS Software WinINSTALL LE is included on the Windows 2000 Server CD in valueadd\3rdparty\mgmt\winstle.

You can use the VERITAS WinINSTALL LE program to view existing .msi packages and to see and modify many aspects of the package, including the following:

- General summary information
- Required files
- Shortcuts
- Registry settings
- Services
- INI edits
- Advertising

You can use this application to view the property table, the summary information table and which DLLs and program files are required for a specific application.

By using the VERITAS WINInstall LE program, you can make simple modifications to the applications general summary information that users see when working in the application. It is recommended, unless you are the author of the .msi package, that you only modify the general summary information area. Modifications to the required files, registry settings, and services can make the package unstable. If you need to make changes to the package, consider creating a transform.

Caution An erroneous package can disable the software on the server and client computers. Do not make changes to the existing .msi package if you are unsure of the possible effects.

Caution Do not use a registry editor to edit the registry directly unless you have no alternative. The registry editors bypass the standard safeguards provided by administrative tools. These safeguards prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your system. Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. To configure or customize Windows 2000, use the programs in Control Panel or Microsoft Management Console (MMC) whenever possible.

You can use WinINSTALL LE to take a snapshot of the current system state and store the configuration. You are then free to make systemwide changes, such as installing software, updating the registry, and configuring settings. After this is complete, you can then take another snapshot of the system. The changes between the two snapshots are compiled into a package known as an .msi file. (This tool can also assist with troubleshooting because it can compare the files and registry information before and after installation.)

It is best to repackage applications on the version of the operating system on which they are to be installed. For example, if you plan to install a package on Windows 2000 Professional, repackage the software on Windows 2000 Professional. If you plan to install the package on Windows NT 4.0, repackage the software on Windows NT 4.0.

For more information about Windows Installer, including the Windows Installer package schema, the details of the Windows Installer API, and sample packages and code, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Note Systems Management Server can distribute and install Windows Installer packages. A step-up tool is available from Microsoft to convert Systems Management Installer packages to Windows Installer packages.

Repackaged Applications

An organization can use repackaged applications for Windows Installer to begin to benefit from the Windows Installer technology and to use the software installation and maintenance feature of Windows 2000. Repackaged applications are not as granular as natively authored applications, that is typically they have one large feature that makes up the Windows Installer product defined by the package, whereas natively authorized packages can contain multiple, customizable features. Repackaged applications can be advertised and repaired and can install with elevated privileges when assigned or published.

Keep in mind that Windows Installer packages can be used for installing applications on computers that are running Microsoft Windows 2000, Windows NT version 4.0, Windows 98, and Windows 98. Systems Management Server can also manage Windows Installer packages.

Repackaging Process

Repackaging requires clean computers. A clean computer is one that has only the operating system installed and has not had any other software installed or removed. Begin the repackaging process by taking a snapshot of the computer without any applications on it. After you have the snapshot, run the existing installation program for the software you want to repackage, configuring the files and registry entries for any customizations you want. Then, when the computer is configured exactly how you want it, take another snapshot. The repackaging software compares the before and after snapshots and uses the differences to generate a Windows Installer package.

Of all the things that impact the success of repackaging, the most important is to start with a clean reference computer; that is, one that has not had other applications installed on it. If you do not use a clean computer, any files or registry entries needed by the application that are already on the computer are not detected by the repackager. This can potentially leave vital files out of the repackaged .msi file, and thus might break the application when you try to install it.

Consider using Windows 2000 Remote OS Installation to manage the clean computers in a repackaging lab.

Thoroughly test all repackaged Windows Installer packages before deploying them in a production environment. Repackaging is generally a safe process, but some applications that modify system settings can cause problems if installed in a repackaged format.

Delete any shortcuts placed in the Startup folder by the application between snapshots, or manually remove the shortcuts from Windows Installer package after it is created. Otherwise the shortcuts are created by the Windows Installer package rather than by the software installation and maintenance feature, and the application automatically installs itself when the users logs on if the application is assigned to them whether this is called for by the Group Policy object or not.

Delete any shortcuts that point to an uninstall program for the application. It is recommended that a repackaged application be removed by using **Add/Remove Programs** in Control Panel so that Windows Installer is triggered to perform the removal. The application might not be removed correctly if the user runs the earlier uninstall program.

In general, avoid restarting the computer between repackager snapshots. Even if the application states that the computer must be restarted after the setup is completed, it might be a good idea not to restart the computer. Windows might adjust certain registry settings or files during the restart process that might inadvertently end up in the resulting package. If your resulting package does not work, and restarting the computer is recommended, rebuild the package and restart the computer prior to taking the second snapshot.

Managing Software with Existing Setup Programs

If you have applications that do not have a Windows Installer package, and you have decided not to repackage the application for Windows Installer, use software installation and maintenance to publish existing setup programs using .zap files.

Note Windows Installer packages whether natively authored or repackaged can be installed automatically with elevated privileges if they are either assigned or published. Typically, existing setup programs would require the user to have local administrator privileges to run the existing setup, which is one reason to consider repackaging versus publishing existing setups using .zap files. Also, if the existing setup does not support clean and complete removal of the software, publishing the existing setup does not improve the software removal experience.

Using the .zap file format, you can publish applications using their existing setups. The .zap file is placed in the same folder on the software distribution point as the original setup program that the .zap file references. When an application is deployed by using a .zap file, the contents of the .zap file are parsed and the application and its associated file name extensions are added to Active Directory.

When you publish the existing setup using .zap files, the experience for the user installing the application is the same as using the existing setup. That is, if the person installing the application has to be an administrator to install the software, they also have to be an administrator to install the published version.

The .zap file resembles an .ini file. It must include an Application section [Application] and might optionally contain a file name extensions section [Ext]. Each section is made up of several parameters and their values. You can create or modify a .zap file by using a text editor such as Notepad.

Caution ZAP files created with Notepad might actually have a name of file name .zap.txt. Make sure that the extension of a zap file is .zap.

The following information is provided as an example for creating a .zap file. Do not copy this information exactly as it is presented. Modify the information to match the application that you are attempting to manage, as well as the location of your software distribution point.

The following three lines are all that is needed in a .zap file to publish Microsoft Excel 97 using the existing Excel 97 setup program:

```
[Application]
FriendlyName = "Microsoft Excel 97"
SetupCommand=""\\server\share\Excel 97\setup.exe""
```

The path and name of the .exe file must be quoted within the [Application] section. If there are no command line arguments, it needs to be quoted twice. For example:

Absolute path:

```
SetupCommand=""\\server\share\long folder\setup.exe""
SetupCommand="\\server\share\long folder\setup.exe" /argument
```

Relative path:

```
SetupCommand=""setup.exe""
SetupCommand="setup.exe" /argument
```

The following example shows a more extensive .zap file for publishing Microsoft Excel 97. This example includes optional information from the application documentation, file name extensions and includes LCID, CLSIDs, and progIDs.

- You only need the Application, FriendlyName, and SetupCommand information. The FriendlyName is a descriptive name that is displayed in the Software Installation snap-in and **Add/Remove Programs** in Control Panel. The SetupCommand provides the universal naming convention (UNC) path to the Setup program for the application. The path must be relative to the .zap file.
- The comments (any line that starts with a semicolon [;]) explain what each entry in the file is for.
- The underscore (_) is a continuation symbol; lines joined by an underscore must be together on one line.
- The additional information about progIDs, CLSIDs, and LCIDs requires customization based on the correct entries, which might involve installing the application, and then determining the correct information by working through the registry.

Note Many entries in this file are optional. The progID, CLSIDs, and LCIDs vary based on your system and the application with which you are working. A .zap file with only the required entries can be much shorter. Do not copy this file exactly as it is.

```
; ZAP file for Microsoft Excel 97

[Application]
; Only FriendlyName and SetupCommand are required,
; everything else is optional

; FriendlyName is the name of the application that
; appears in the Software Installation Snap-in
; and the add/remove programs control panel.
; REQUIRED
FriendlyName = "Microsoft Excel 97"

; SetupCommand is the command line to
; Run the applications setup. if it is a relative
; path, it is assumed to be relative to the
; location of the ZAP file.
; Long file name paths need to be quoted. For example:
; SetupCommand = "long folder\setup.exe" /unattend
; or
; SetupCommand = "\\server\share\long _folder\setup.exe" /unattend
; REQUIRED
```

```
SetupCommand = ""setup.exe""

; Version of the application that appears
; in the Software Installation Snap-in and the
; add/remove programs control panel.
; OPTIONAL
DisplayVersion = 8.0

; Publisher of the application that appears
; in the Software Installation Snap-in and the
; add/remove programs control panel.
; OPTIONAL
Publisher = Microsoft

; URL for application information that appears
; in the Software Installation Snap-in and the
; change/remove page of add/remove programs control panel.
; OPTIONAL
URL = http://www.microsoft.com/office

; Language for the app, in this case US English.
; OPTIONAL
LCID = 1033

; Architecture, in this case, intel.
; OPTIONAL
Architecture = intel

; the [ext] [CLSIDs] and [progIDs] sections are
; all optional

[ext]
; List the file name extensions this application supports
; "auto-install" for. This entire section is optional.

; When listing file name extensions you can place a dot in front of the
; extension or leave it off. For example you can use .XLS= or XLS=
; The text listed after the first = sign is optional and ignored.
; The first = is required (or the entire line will be ignored).
XLS=
XLA=
XLB=
XLC=
XLM=
XLV=
XLW=

[CLSIDs]
; CLSIDs that this application will "auto-install"
; for. This entire section is OPTIONAL.
```

```
; Format is CLSID with LocalServer32,
; InprocServer32, and/or InprocHandler32 (in a
; comma separated list) after the =.

{00024500-0000-0000-C000-000000000046}=LocalServer32
{00020821-0000-0000-C000-000000000046}=LocalServer32
{00020811-0000-0000-C000-000000000046}=LocalServer32
{00020810-0000-0000-C000-000000000046}=LocalServer32
{00020820-0000-0000-C000-000000000046}=LocalServer32
{00020820-0000-0000-C000-000000000046}=LocalServer32

[progIDs]
; progIDs that this application will "auto-install"
; for. This entire section is OPTIONAL.

; format is a CLSID, with the corresponding progid
; listed after the = sign
{00024500-0000-0000-C000- _
000000000046}=Excel.Application
{00024500-0000-0000-C000- _
000000000046}=Excel.Application.8
{00020821-0000-0000-C000-000000000046}=Excel.Chart
{00020811-0000-0000-C000-000000000046}=Excel.Chart.5
{00020821-0000-0000-C000-000000000046}=Excel.Chart.8
{00020810-0000-0000-C000-000000000046}=Excel.Sheet.5
{00020820-0000-0000-C000-000000000046}=Excel.Sheet.8
{00020820-0000-0000-C000-000000000046}=Excel.Sheet
{00020820-0000-0000-C000-000000000046}=Excel.Template
{00020820-0000-0000-C000-000000000046}=Excel.Workspace
```

Customizing Windows Installer Packages

You can customize Windows Installer applications by using transforms. Authoring and repackaging tools provide mechanisms that permit the creation of transforms, and applications might also provide wizards or templates that permit a user to create transforms.

For example, Office 2000 has the Office 2000 Custom Installation Wizard that builds transforms. By using the Office 2000 Custom Installation Wizard, you can create a transform so that you can manage the configuration of Office 2000 that is deployed to users.

By customizing the Windows Installer package with a transform, you can provide the key features that users need. In the case of Office 2000, Word might be a key feature, so it installs during the first installation. You might customize less popular features, such as revision support or document translators to be installed on first usage. You might decide that it is best that other features, such as clip art, not be installed at all. And again, you might create another transform that provides all of the features of Word and does not install PowerPoint. The exact mix of which features to install and when to install them varies based on the needs of your users.

You can transform or modify a Windows Installer package to handle a variety of customizations. For example, you can control which features are installed and where they are located: on the local computer's hard disk drive or used from a file server. Although they are optional, transforms can be used for a variety of purposes including the following:

- Encapsulating customizations for particular groups of users to be applied to a base package, which allows, for example, for a single administrative installation point with a base package and transforms for the Finance group, Support Staff group, and Legal group.
- Adding new features to an existing application's package. For example, adding custom corporate-developed Excel templates for the Finance group that do not change on a regular basis. (If the templates change frequently, a better practice would be to package them and assign or publish them on their own.)

Note It is recommended that transforms are stored at the same software distribution point (in the same shared folders) as the Windows Installer package that they customize.

A Windows Installer package might be associated with one or more transforms that get applied to the package at deployment time. Many of the customizations that you used to perform by repackaging applications are better handled by transforming the Windows Installer package.

Transforms are associated with the Windows Installer package at deployment time rather than when the Windows Installer is actually using the package to install or modify the application. Deployment time is when you are first working with the Windows Installer package in the Software Installation snap-in. The only way to associate transforms with a Windows Installer package is by selecting the **Advanced published or assigned** option from the **Deploy Software** dialog box when you are adding a package to the Software Installation snap-in.

It is best to create separate packages for each application. For example, if you have four separate applications you want to install, create four separate packages so that the Windows Installer service has the ability to better manage the installation state in the case of an installation failure. For example, if the installation of a package fails, Windows Installer can undo the installation. If four applications are one package, and the first three applications install, and then there is a failure with the fourth application, all four applications would be removed when the single package fails.

Note In the past, administrators often customized the software by repackaging the software. Do not try to repackage applications that use Windows Installer, instead use transforms for additional customization.

Note If a transform needs to change, make a new transform, and then remove the existing application and transform and reassign or publish the application with the new or changed transform. If you have to do this, consider making an upgrade relationship between the application and the old transform and the application and the new transform. Of course, it is recommended that this be tested to ensure you get the results that you want.

Important When you create a transform to customize a Windows Installer package, do not change the product code of the package. If you change the product code, you must verify the assignment and publishing of the package with the new product code works as if it were an upgrade or it might not be available to the users who already have the existing package (with the old product code).

Modifying Windows Installer Package Properties

You can customize the summary information screen of a Windows Installer package to fit the needs of your organization and your users. For example, you can include support information for your company to make it easier for your users to access support for the application being installed.

The information that you add is displayed when a user selects additional information in **Add/Remove Programs** in Control Panel.

To add modifications to an application package, use one of the Windows Installer authoring or repackaging tools. The process that you use to modify information and the information that you can modify varies according to the tool that you select. For example, you can use the VERITAS WinINSTALL LE application that is included with Windows 2000 Server to open the package and customize the summary information data.

The summary information screen contains basic data about the Windows Installer package and the application the package installs. It includes the application name, the application publisher, and the version of the application as well as other basic company support information.

You can safely modify the following items on the screen:

- Name of Application, Version, and Manufacturer—include product specific information.
- Online Support—include your own support information
- Phone—include your own support phone numbers

Caution You can damage the entire package if you modify any information outside the basic summary information listed in the previous paragraph. For example, you might unintentionally make the package unavailable to some users.

Distributing Additional Files

Many organizations need to distribute additional files with an application. For example, you might want to distribute a sales quote template with Word and a customized bidding spreadsheet with Excel.

These files can be bundled with the executable file and other source files for the application when you create the package, or you can include them as a transform.

However, these files are often more volatile than the application files. Therefore, you might want to package them in their own Windows Installer package. This package has its own product code, and therefore it can be upgraded on a different schedule than the application. If the files are included as part of the application transform, each time the templates are updated, the entire application transform needs to be deployed again.

When working with your own templates or custom files, it is easy to create new packages using one of the available native authoring tools for Windows Installer packages. You only have to define the files and the location where you want the files to installed, the target location. Build a package to distribute only the updated files, and if necessary, the files are compressed into their own Windows Installer package.

Terminal Services and Software Installation and Maintenance

You can set up a Terminal Services server to run in a remote administration mode or an application server mode:

- A Terminal Services server that is set up in remote administration mode allows a user to use the Terminal Server to perform administration functions only.
- A Terminal Services server that is configured as an application server provides multiple concurrent user connections to any number of applications.

Using Terminal Services, you can deploy and manage applications from a central location. After an application is deployed, many clients can connect to the Terminal Services server. Applications can be installed on client computers from Terminal Services servers configured as remote administration servers using Windows Installer. The underlying Terminal Services server code and Windows Installer code work together to deliver a seamless installation, in contrast to how applications were installed on a Terminal server in the past. The computer had to be put in an installation mode first; and after application installation, it was put back into an execution mode.

Table 23.12 shows when assigned and published software is supported on remote administration and application Terminal Services servers.

Table 23.12 Support for Managed Software on Terminal Services

Software Installation	Remote Administration	Application Server
User Assigned	Supported: Supported means that software installation and maintenance works in the same manner as it would on Windows 2000 Professional	Not applied, software is not installed.
Publish	Supported: Supported for both Windows Installer packages and existing setup programs defined in .zap files for publishing.	Not applied, software is not installed.

(continued)

Table 23.12 Support for Managed Software on Terminal Services *(continued)*

Software Installation	Remote Administration	Application Server
Computer Assigned	Supported	Supported: A domain user, with a roaming user profile might roam to an application server. Their application shortcuts follow them to the application server. If the server has the same application either installed (per computer) or assigned, and the user activates the shortcut, the shortcut works (either activating or installing the application). If the application is neither installed nor assigned, the shortcut does nothing.

On an application server, software installation must not jeopardize the running system, and it must be configured to allow concurrent usage of an application. For these reasons, software installation is only permitted by you, as an administrator. To accomplish this, Terminal Services servers configured as application servers restrict application deployment by only allowing Administrators to initiate a Windows Installer–based application installation and by not allowing any per-user application deployment. All applications deemed necessary must be fully installed by the system administrator before users are allowed to connect and use the computer through remote sessions. As such, it is your responsibility to decide what is required and to ensure that all bits are locally installed and available prior to allowing remote user connections.

Software Installation and Maintenance and Backing Up Data

Back up any software distribution points that you create to manage the deployment of software, packages, and customizations. Software installation and maintenance relies on Group Policy, and software installation and maintenance and Group Policy store information in both Active Directory and the sysvol.

Backup both Active Directory and sysvol immediately after making changes to software installation and maintenance to maintain version compatibility. For example, if you publish a new application without backing up and then you have to restore Active Directory and sysvol with a backup that does not have the published application, any users who installed the published application are installing the old application in an unmanaged state. The best way to prevent this is to keep current backups.

If you do not receive the data that you expected when restoring a backup, verify that you have the correct administrative permissions.

Important If you use the built-in administrators account on a domain controller, the backup and restore operation does not work correctly. Domain Administrators or equivalent access is required to back up. The built-in Administrators group does not have the same inherited permissions on Group Policy objects as it does for the rest of the domain server namespace. In a standard configuration, the access control list (ACL) set on the Policies container does not inherit from its parent.

For more information about backing up and restoring data in Active Directory, see "Active Directory Backup and Restore" in this book. For more information about the tools required for backing up data, see "Backup" in the *Microsoft® Windows® 2000 Server Resource Kit Server Operations Guide*.

Best Practices and Troubleshooting

You can use Group Policy, to specify your organization's specific requirements for managed desktop configurations for groups of computers and users. Microsoft® Windows® 2000 Server includes hundreds of Group Policy settings that you can configure. Because Group Policy is flexible and includes options for registry-based policy settings, security settings, software installation, scripts, computer startup and shutdown, user logon and logoff, and folder redirection, the Resource Kit provides tools for troubleshooting potential problems.

For detailed information about the tools that you use to troubleshoot software installation and maintenance, see "Troubleshooting Change and Configuration Management" in this book.

Assigning the same software to both users and computers

Although some scenarios lend themselves to assigning the software to the computer, and others to assigning it to users, avoid assigning the same software to both users and computers. This applies to assigning the software within one Group Policy object or assigning the software to users in one Group Policy object and to computers in another Group Policy object if both Group Policy objects can be applied to a single user when Group Policy is applied.

When an application does not immediately show up in Add/Remove Programs

When a new Group Policy object is created and edited, users will not see any new application assigned by this Group Policy object in **Add/Remove Programs** until the next time they restart their computers.

When you add a new Group Policy object with a new managed application the Software Installation Group Policy setting is applied when the user logs on. If the user clicks **Add/Remove Programs** in Control Panel and they do not see the new application (from the new Group Policy object), the user must log off and then log back on. **Add/Remove Programs** does not reflect the new Group Policy object until the Software Installation Group Policy setting is applied again.

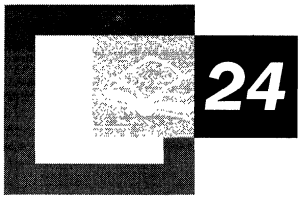
You cannot use the Group Policy refresh command, secedit/RefreshPolicy, to apply an updated Software Installation Group Policy setting. Secedit is ignored and has no effect on Software Installation. Software Installation also ignores the period refresh of Group Policy (default 90 minutes with a 30 minute offset).

Additional Resources

- For more information about Windows Installer, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.
- For more information about Windows Installer packages, see the Microsoft Knowledge Base link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. Search the Knowledge Base with the key words "Windows Installer" and "Repackaging applications."
- For more information about the client-side options that are available in Windows 2000 Professional, see the *Windows 2000 Professional Resource Kit* and the *Deployment Planning Guide.*

CHAPTER 24

Remote OS Installation



One of the most challenging and costly functions performed by information technology (IT) staff today is the deployment of a new operating system to clients. Microsoft® Windows® 2000 Remote OS Installation assists you in deploying an operating system throughout your organization. Remote OS Installation uses Preboot Execution Environment (PXE)–based remote boot technology to assist IT staff with centralized deployment of Windows 2000, thus eliminating the need to visit the desktop.

In This Chapter

Related Information in the Resource Kit

- For more information about using other methods for automating the installation of Windows 2000, see "Automating Server Installation and Upgrade" in the *Microsoft® Windows® 2000 Server Resource Kit Deployment Planning Guide*.
- For more information about the Windows 2000 directory service, Active Directory™, see "Active Directory Logical Structure" in this book.
- For more information about Group Policy, see "Group Policy" in this book.
- For more information about software installation, see "Software Installation and Maintenance" in this book.

Remote OS Installation Overview

Remote OS Installation and the IntelliMirror™ technologies are combined to form the change and configuration management features included in Microsoft® Windows® 2000 Server. Using Remote OS Installation you can customize and enable automated installation of Microsoft® Windows® 2000 Professional on new or replacement computers. You can experience better disaster recovery with easier operating system and application management by combining Remote OS Installation with the user data management, software installation and maintenance, and user settings management features that make up IntelliMirror. For more information about IntelliMirror, see "Introduction to Desktop Management" in this book.

This chapter focuses on implementation of Remote OS Installation by using Remote Installation Services (RIS) technology for the installation of Windows 2000 Professional on remote boot–enabled clients. RIS supports clients without an operating system or failed computers that need to have the operating system restored. When using Remote OS Installation with the default settings, everything is deleted from the hard disk and a new operating system is installed. Previous user profile and configuration settings are also removed when installing the operating system.

Note Remote OS Installation cannot be used to upgrade an existing operating system on computers currently running Microsoft® Windows® 95, Microsoft® Windows® 98, Microsoft® Windows NT® Workstation version 3.51, or Microsoft® Windows NT® Workstation version 4.0.

If you want to maintain your existing user profiles, there are a few additional methods that Windows 2000 supports for installing and upgrading an operating system on clients. These include the following:

- Performing disk duplication with the System Preparation (SysPrep.exe) tool
- Using Microsoft® Systems Management Server
- Using a bootable CD

These methods are not discussed in detail in this chapter. For more information about installing and upgrading an operating system by using these methods, see "Automating Server Installation and Upgrade," "Automating Client Installation and Upgrade," and "Using Systems Management Server to Deploy Windows 2000" in the *Microsoft® Windows® 2000 Server Resource Kit Deployment Planning Guide*. For more information about upgrading Windows 2000 on an existing Windows 2000 Professional client, see "Software Installation and Maintenance" in this book.

Remote OS Installation Requirements

Remote OS Installation requires other services and capabilities on both the server and client to run. Figure 24.1 shows the Remote OS Installation requirements.

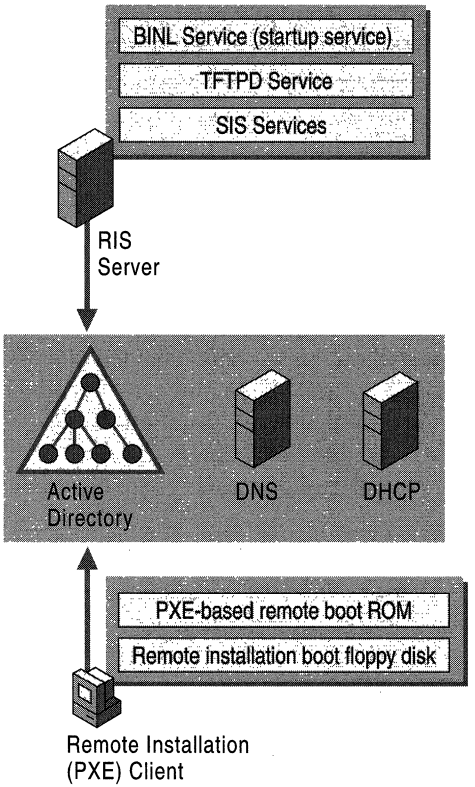


Figure 24.1 Remote OS Installation Server and Client Requirements

Server Software Requirements

Remote OS Installation requires you to install these server technologies included with Windows 2000 Server. You can install these services on individual servers or all on one server:

Remote Installation Services (RIS) RIS is an optional component of Windows 2000 Server that provides the services that allow you to install Windows 2000 Professional from a RIS server.

Domain Name System (DNS) RIS servers rely on DNS for locating Active Directory directory services and for completing domain operations. You can use Windows 2000 DNS and receive the benefit of dynamic updates for your DNS server. However, using the Windows 2000 version of DNS is not required for RIS to function. The DNS server you use must support the SRV RR (RFC 2052) and needs to support the dynamic update protocol (RFC 2136). For more information about DNS, see "Introduction to DNS" and "Windows 2000 DNS" in the *Microsoft® Windows® 2000 Server Resource Kit TCP/IP Core Networking Guide*.

Dynamic Host Configuration Protocol (DHCP) Server RIS servers require a DHCP server to be present and active on the network. Remote boot–enabled clients receive an IP address from the DHCP server prior to contacting a RIS server. You can install the version of DHCP that is included with Windows 2000 Server, or you can use an existing version of DHCP running on Microsoft® Windows NT® Server version 4.0. You can also use a third-party DHCP. For more information about DHCP, see "DHCP Options" and "DHCP Message Formats" in the *TCP/IP Core Networking Guide*.

Active Directory RIS servers must be installed on a Windows 2000 Server that has access to Active Directory. This can be a domain controller or a server that is a member of a domain with access to Active Directory. RIS uses Active Directory to locate existing clients and other RIS servers. You can administer RIS by using extension property pages that reside on specific computer objects within Active Directory. For more information about Active Directory, see "Active Directory Logical Structure" in this book.

For more information about installing and configuring a RIS server, installing and configuring the DNS service, and installing the DHCP service, see Windows 2000 Server Help.

Hardware Requirements

Ensure that both your server and client hardware meet the minimum installation hardware requirements for Windows 2000. For more information about Remote OS Installation hardware requirements and compatible computers or network adapters, see the Hardware Compatibility List link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

In addition, be sure that your server and client hardware meet the requirements in the following sections.

Server Hardware Requirements

Windows 2000 Remote OS Installation requires the following server hardware:

- Pentium or Pentium II 200 megahertz (MHz) or faster processor (Pentium 166 MHz minimum).
- 96 to 128 megabytes (MB) of random access memory (RAM) needed to run when services such as Active Directory, DHCP, and DNS are installed.
- 2-gigabyte (GB) disk drive for the Remote Installation Services servers folder tree.
- 10 megabytes per second (MBps) network adapter (100 MBps recommended).
- CD-ROM drive or access to a network share containing Windows 2000 Professional.

RIS requires a significant amount of disk space to store operating system images. Dedicate an entire hard disk drive partition specifically to the directory tree of the RIS server. Small computer system interface (SCSI)–based disk controllers or disks are preferred. Format the drive where you want to install RIS with NTFS. You cannot install RIS on the same drive as the system volume. RIS does not support the installation of images on to Encrypting File System (EFS) or the Distributed file system (Dfs) volumes.

Client Hardware Requirements

Windows 2000 Remote OS Installation requires the following client hardware:

- Pentium 166 MHz or faster processor Net PC PXE-based remote boot–enabled client.
- 32 MB of RAM minimum (64 MB recommended).
- 800 MB or larger disk drive.
- DHCP PXE–based boot ROM version .99c or later or a network adapter supported by the RIS boot floppy disk.

Note There are several ways to enable a computer with PXE remote boot services. All clients with Net PC and PC98 are required to have PXE support. The way that a hardware vendor supports PXE and the version of PXE that you have implemented is important for troubleshooting and diagnosing failures.

Network Adapter Containing a PXE-based Remote Boot ROM

Network adapter manufacturers can embed the PXE ROM code on a chip as part of the network adapter itself. The chip is usually a piece of silicon that you can upgrade independently of the system BIOS. There are several manufacturers that create a version of PXE-based remote boot code as part of the client system or as chips that reside on network adapters. A PXE-based remote boot ROM network adapter is included on computers that are Net PC or PC98–compliant systems. The client might also be an existing computer equipped with a PXE-based remote boot ROM that adheres to the PXE environment specifications. For more information about PXE protocol and process, see "Preboot Execution Environment" later in this chapter.

Network Adapter Supported by the Remote Installation Boot Floppy Disk

The remote installation boot floppy disk is used to start the process of remote operating system installation for computers that do not have a supported PXE-based remote boot ROM. The boot floppy disk is a PXE emulator that connects to the RIS server through PCI network adapters. The Rbfg.exe tool is used to generate a remote installation boot floppy disk. When you create a boot floppy disk, all of the network adapters listed in the **Adapter List** option are available on that floppy disk, and a specific network boot disk is not required. All of the adapters in this list are PCI-based network adapters. ISA, ISA (Plug and Play), and PCMCIA adapters are not supported. To see a list of supported network adapters, start the Rbfg.exe tool, and then click **Adapter List**. The Rbfg.exe tool does not allow you to add additional network adapters.

Note PXE remote boot can also occur when the BIOS is configured to start from the network adapter as the primary start device, as opposed to from the hard disk drive. Hardware vendors can embed PXE remote boot support in the computer's system BIOS. In this case, the remote boot ROM code is actually part of the computer's BIOS, which requires that you upgrade the system to upgrade the PXE ROM code.

Creating a Remote Installation Boot Floppy Disk

The remote installation boot floppy disk can be useful when you need to install an operating system on portable computer systems and systems that do not include a supported boot ROM using RIS. Because the current PXE specifications do not support PCMCIA adapters, you can place the portable computer in a docking station and use a RBFG-generated boot floppy disk to connect to the server, assuming that the docking station contains a supported PCI network adapter. Some docking stations have PXE ROMs built into the network adapter and do not need the floppy disk. (See your portable computer manual.)

The remote boot floppy disk generator tool, Rbfg.exe, can be found on every RIS server, in the Reminst share at *RISservername*\Reminst\Admin\i386 directory. The Rbfg.exe tool also exists in the Administrative Tools Windows Installer (Adminpak.msi) package that is included with Windows 2000 Server. You can run this tool from either the Reminst share, or on a computer that has the Administrative Tools package installed. The Rbfg.exe floppy disk generator gives you the option to create a remote boot disk to use with RIS on either floppy disk drive A or drive B.

▶ **To create a remote boot disk**

1. From the **Start** menu, click **Run**.
2. In the **Open** box, type the path of the Rbfg.exe tool, and then click **OK**. The default path is:

   ```
   \\RISservername\Reminst\Admin\i386\Rbfg.exe
   ```

3. Insert a formatted floppy disk into the disk drive.
4. Click the destination drive option, and then click **Create Disk**.
5. Click **Close**, and then remove the floppy disk from the disk drive.

Remote Installation Services

RIS provides software services that allow an administrator to set up new clients remotely, without having to visit each client.

There are several administrative options that you can control to configure how the RIS server services clients. You can accept the default configuration settings and begin offering users operating system installation images. You can also customize the settings to define your automatic computer naming policy, to define the Active Directory container in which computer accounts are created, and to define the operating system images to which users have access.

RIS servers can be set to respond only to service requests from clients that have already had their computer account objects created in Active Directory for a forest, or have been prestaged in Active Directory. RIS technology allows the coexistence of remote installation servers from multiple vendors on the same physical network. When set to ignore boot requests from unknown clients, RIS servers can be introduced into a network without interfering with pre-existing remote installation servers that use the same remote boot protocols.

Important The entire ROM sequence is not secure with packet type encryption, client, or server spoofing, or wire sniffer–based mechanisms. Use caution when using RIS on your corporate network. Ensure that you only allow authorized RIS servers on your network and that the number of administrators allowed to install or configure RIS servers is controlled.

RIS Components

The following are the various components that you use to install, configure, and implement RIS within your organization:

Remote Installation Services Setup (RISetup.exe) You can install RIS at the same time or after you install Windows 2000 Server by using **Add/Remove Programs** in **Control Panel**.

Remote Installation Preparation Wizard (RIPrep.exe) RIPrep allows you to create a customized image of a Windows 2000 Professional computer. *Imaging* Windows 2000 Professional means creating a replica of a hard disk that you can then install on other computers in your organization. You can use this wizard to prepare an existing Windows 2000 Professional installation image and to replicate that image to an available RIS server on the network. The image can include the operating system alone or it can be a preconfigured desktop image, including the operating system and standard locally-installed desktop applications.

Client Installation Wizard (OSChooser) OSChooser is the client-side (the part that is downloaded to the client) text-based program that communicates with the RIS server. The Client Installation Wizard is a default set of screens that the Boot Information Negotiation Layer (BINL), the server-side service, sends to the client to guide the user through the remote installation process. Users of remote boot–enabled clients use the Client Installation Wizard to log on and select from operating system installation options. You can customize the setup screens to meet the needs of your organization.

RIS Services

RIS comprises individual services that have been combined to enable the remote installation of Windows 2000 Professional. The Remote Installation Setup Wizard (RISetup) configures and starts the following services:

Boot Information Negotiation Layer (BINL) This service listens for and answers DHCP (PXE) requests. It also services Client Installation Wizard requests. BINL directs the client to the files needed to start the installation process. This service also checks Active Directory to verify credentials, determine whether a client needs service, and whether to create a new or to reset an existing computer account object on behalf of the client.

Trivial File Transfer Protocol Daemon (TFTPD) A RIS server uses TFTP to download the initial files needed to begin the remote installation process to the client. This includes the Client Installation Wizard and all files needed to start Windows 2000 Setup. The first file downloaded to the client using TFTP is Startrom.com. Startrom is a small bootstrap program that displays the **Press F12 for Network Service Boot** prompt. If F12 is pressed within three seconds, the Client Installation Wizard (OSChooser) is downloaded to begin the remote installation process. When it resides on the server side, it is called the Trivial File Transfer Protocol Daemon (TFTPD), and when it resides on the client, it is called Trivial File Transfer Protocol (TFTP).

Single Instance Store (SIS) SIS services consist of an NTFS file system filter and a service that acts on the volume on which the RIS images are kept. SIS services reduce the storage requirements needed to store these images by combining duplicate files.

Installing RIS

You can install Remote Installation Services either on a Windows 2000 server that is already on a network or on a stand-alone server that you want to add to the network.

▶ **To install the RIS component**

1. From the **Start** menu, point to **Settings**, and then point to **Control Panel**.
2. Double-click Add/Remove Programs, and then click Add/Remove Components.
3. Check the box for **Remote Installation Services**.
4. Click **Next**.
5. Click **Finish**, and then **Restart** your server. After the server has restarted, complete the Configure Remote Installation Services wizard.

RISetup locates the first NTFS volume that does not contain the system volume (boot.ini file) or the boot volume (%windir%). You then define image properties such as directory name, friendly description, and Help text for the end user. None of these properties can contain any non-ASCII characters. For more information about the Client Installation Wizard variables, see "OSCML and Client Installation Wizard Variables" in this book. This process also creates the RIS directory structure and copies the files required for a CD-based "flat" installation image of Windows 2000 Professional. This process also copies the default Client Installation Wizard screens and configures and starts all of the services required for RIS (BINL, SIS filter, SIS Groveler, and TFTP).

For more information about installing and configuring RIS servers and DNS, see Windows 2000 Server Help. Windows 2000 Server Help also provides instructions for promoting a stand-alone server to a domain controller and installing and configuring the DHCP service.

For more information about the Client Installation Wizard, see "Client Installation Wizard" later in this chapter.

Deploying RIS Servers

RIS servers are dependent on your network configuration. The way you deploy and manage your RIS servers on the corporate network determines how your RIS servers perform. By using RIS servers, you can have one operating system image that supports multiple sites, domains, and organizational units, or you can customize each image to meet the needs of the users and computers being served.

You might need multiple RIS servers to support your corporation, or only one RIS server if you are deploying Windows 2000 on a small localized network or network segment. As a general guideline, place a RIS server near the client computers that it services.

The amount of traffic the RIS server produces is similar to that of other servers performing as software distribution points on your network. Generally, the traffic for RIS servers is predictable. RIS-generated traffic is higher when many users are installing their initial operating system image, for example, during a deployment of new operating system images or when a group of new computers is being added to the network. After the operating systems are installed, the daily RIS server traffic will be lower.

Depending on the size of your network, you might need to adjust the distribution and management of client access to RIS servers to streamline access or support multiple operating system requirements. Determine your corporate needs before deploying automated customized versions of Windows 2000 and prior to imaging a standard desktop–configured computer. The number of RIS servers that you need to deploy is determined by the demand for new, upgraded, and customized operating system installations; the speed of your network; and the hardware you use to support your RIS images. Figure 24.2 shows one way to place your RIS servers and optional referral servers in relation to clients for a large organization deployment strategy.

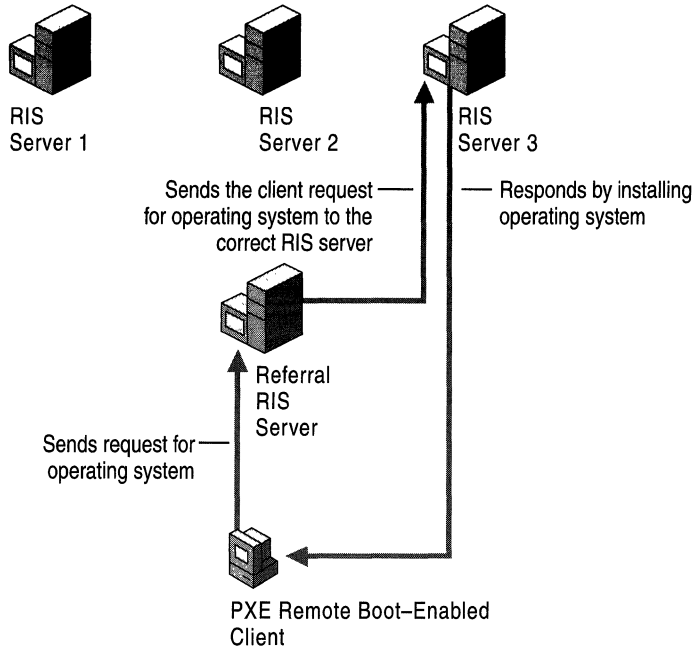


Figure 24.2 Sample RIS Server Layout Within a Large Organization

As illustrated in Figure 24.2, a new remote boot–enabled client requests a remote operating system installation. This request is passed to the RIS referral server, which has the **Do not respond to unknown client computers** option enabled. Clients that have been prestaged in Active Directory ahead of client servicing can access this RIS server. A prestaged client is a client that already has its computer account object created in Active Directory within a forest. The referral RIS server checks Active Directory to verify whether a computer account object exists for this client. In this example, the client was prestaged by the administrator, therefore it has a corresponding computer account object in Active Directory, and it is assigned to be serviced by RIS Server 3. The RIS referral server passes the request on to RIS server 3, from where the client then begins installing the operating system.

Figure 24.2 shows how one RIS server layout works in a large corporate setting. For this scenario there is close control on which clients can access which RIS servers. When the computer account object is created, the computer account object is assigned to a specific RIS server. Depending on your corporate environment, you can configure your RIS servers so that all RIS servers can respond to all clients. In the network design in Figure 24.2, the only purpose of RIS servers 1, 2, and 3 is to provide images of the operating system. These servers do not respond to initial client service requests. The referral RIS server does not provide image support, however, it does answer client service requests, checks Active Directory for the existence of a prestaged computer account object, and then refers the client to the specified RIS server.

By pre-staging clients to Active Directory and distributing various image files over different distribution points, you can control network traffic and speed up the installation process. Slow connections to your RIS servers can slow down the entire network if they are not designed and distributed appropriately or if the hardware utilized by the RIS server cannot support network demands. If your organization has branch offices, it is best to place a RIS server in each branch location and not attempt to install software over a slow network connection.

Note RIS does not detect slow links. RIS times out only if it does not receive a DHCP packet from the server.

For more information about optimizing performance of RIS servers, see "Automating Client Installation and Upgrade" in the *Deployment Planning Guide.*

Authorizing RIS Servers in Active Directory

To manage and control the way a RIS server interacts with existing and potential clients, you can configure the server properties, which allow you to determine how this RIS server responds to clients requesting service. You can administer the majority of the RIS configuration settings from a Windows 2000 Professional client by installing the Administrative Tools (Adminpak.msi) package that is included with Windows 2000 Server.

Note The Administrative Tools package (adminpak.msi), can be installed from the directory *%Windir%*\System32 where *%Windir%* is equal to the WINNT directory created during Windows 2000 Server installation. To install the Administrative Tools package on a workstation, run the file adminpak.msi that is located on the Windows 2000 Server operating system CD.

By using RIS you can designate which RIS servers can accept and process requests and designate which RIS servers can only service clients on the network. Before a RIS server can accept requests, it must be authorized to run in Active Directory.

Before a RIS server can accept requests, it must be authorized to run. To authorize a RIS server in Active Directory, you must log on with an account that is a member of the **Enterprise Admins** group in the forest where you want the RIS server to be authorized. By default, members of the **Enterprise Admins** group are the only users who can authorize DHCP/RIS servers. You can perform the procedure in this section on a domain controller, member server of the domain, or a Windows 2000 Professional–based computer that has the Administrative Tools package installed.

If the RIS server is not authorized in Active Directory, the RIS server cannot respond to clients requesting service. If you install RIS on a server that is not an authorized DHCP server, or if you add it to a DHCP server that is not authorized in Active Directory, perform the following procedure.

Note If RIS is installed on a DHCP server that is already authorized in Active Directory, you do not need to perform the following procedure.

▶ **To authorize a RIS server in Active Directory**

1. Log on to the domain where the RIS server resides. (The account used must be a member of the **Enterprise Admins** group.)
2. From the **Start** menu, point to **Programs** and **Administrative Tools**, click **DHCP** from the list. This starts the DHCP Management snap-in.
3. Right-click the **DHCP** root node in the scope pane, and then click **Manage Authorized Servers**.
4. Click **Authorize**, enter the IP address or name of the RIS server, and then click **OK**. When prompted, to ensure that this is the correct RIS server to authorize, click **Yes**.

Now your RIS server is authorized in Active Directory and can respond to clients requesting service. If your server is not responding to requests, the changes to Active Directory might not have taken effect. For these rights to apply immediately, on the domain controller on which you set the user rights, from the **Start** menu, click **Run**, type CMD, and at the command prompt, type:

```
secedit /refreshpolicy /MACHINE_POLICY
```

Or, if applicable in your network environment, you can restart the server so that Active Directory and Group Policy settings take effect.

Configuring RIS Servers

With RIS, you can designate which RIS servers can accept and process requests and also designate which RIS servers will only service clients on the network

After you have successfully installed RIS and authorized it in Active Directory, configure your RIS settings. These settings are required to service clients on your network. From within the Active Directory Users and Computers snap-in, use the RIS Administrative Tools to do the following:

- Reset and create computer account objects.
- Browse Active Directory.
- Search for computer accounts by name, globally unique identifiers (GUIDs), and dedicated servers.
- Configure the server.

These settings allow clients to locally install Windows 2000 Professional from RIS servers. RIS does not provide a mechanism for replicating operating system images from one RIS server to another, such as from RIS server 2 to RIS server 3 in Figure 24.2. However, you can use third-party replication tools for operating system image replication. Make sure that the replication mechanism supports the file maintenance attributes, extended attributes, and security settings of the source images.

Restricting Client Installation Options by Using Group Policy

Group Policy applies to sites, domains, and organizational units. It is important that you understand the effects of Group Policy in your organization before setting specific policies for your users or computers. You can determine which choices the Client Installation Wizard displays to a particular user or user group by using the Group Policy snap-in. For more information about Group Policy, see "Group Policy" in this book.

To restrict the Client Installation Wizard options for users of RIS in your organization, set the desired Group Policy settings for the RIS servers on your network by using the following procedure.

▶ **To set RIS policy to restrict the installation options for a particular user or security group**

1. Locate the Active Directory container where you want to set the RIS policy settings. By default, the RIS policy settings are applied in the **Default Domain Policy Object**, which is located at the root of your domain.
2. Right-click your domain root name, click **Properties**, and then click the **Group Policy** tab.
3. In the **Group Policy Object Links** window, select your Default Domain Policy object, and then click **Edit**.
4. Click **User Configuration**, double-click **Windows Settings**, and then click **Remote Installation Services**.
5. Double-click **Choice Options** in the right pane.
6. On the **Policy** tab, set the **Automatic Setup**, **Custom Setup**, **Restart Setup**, and **Tools** settings. Set the policy for the options available to users in the Client Installation Wizard from the following choices:
   - **Allow** Users who are affected by this policy are allowed to access that installation option in the Client Installation Wizard.
   - **Don't Care** Users receive the policy settings of the parent container. For example, if the administrator for the entire domain sets a RIS-specific policy, and the administrator of this container chooses the Don't Care option, the policy that is set on the domain is applied to all users who are affected by that policy.
   - **Deny** Users who are affected by this policy are not allowed to access that installation option in the Client Installation Wizard.

For more information about the Client Installation Wizard, see "Using Client Installation Wizard to Install Clients" later in this chapter.

Defining a Computer Naming Policy

The computer naming policy that is used during operating system installation provides the computer with a unique name. The computer name identifies the client on the network, similar to the NetBIOS name used in Microsoft® Windows NT® version 4.0. If you have an existing computer naming policy, you can set this format prior to users turning on their computer and requesting an operating system installation.

You can determine the computer naming format and the Active Directory container in which client accounts are created. In a large organization where multiple RIS servers are available, it is beneficial to define a computer naming policy to use to prestage clients and define which RIS servers that a client can access.

▶ To define computer naming policy

1. Start the Active Directory Users and Computers snap-in.
2. Right-click the RIS server.
3. Click **Properties**, and then click the **Remote Install** tab.
4. Click **Advanced Settings**.
5. Click **New Clients**.
6. Define computer naming and where the computer account object is created for new clients.

The **New Clients** page of the Advanced Settings property sheet allows you to control the name that the client is assigned when a user selects the Automatic Setup option within the Client Installation Wizard and where the computer account object is created in Active Directory. The naming format defaults to the user name of the account entered in the Client Installation Wizard with an incremental number (#) appended. You can customize this format. Table 24.1 lists the RIS computer naming options.

Table 24.1 RIS Computer Naming Options

Naming Options	Property
%first	User's first name
%last	User's last name
%Username (Default)	User's logon name
%MAC	Media access control (MAC) address of the network adapter
%#	Incremental number
%nField	Number of characters to be used in indicated field

Note You cannot use all Active Directory object attributes to create a naming format for use with the RIS automatic computer naming feature.

For example, if you create a name with the following format:

```
%5Username%3#
```

Where Username = JoeUser, %nField = %5, and %# = %3.

This yields the name: JoeUs123

For %5, it uses the first five characters of "JoeUser", which results in the "JoeUs" characters in the account. The "123" is determined by scanning Active Directory for existing computer account objects. The %3# specifies to use a three-digit number for the number. In this case, it had to go up to 123 to find a number opening, hence "JoeUs123". By changing the number in "#3", you can restrict or broaden the search from 0-9 to 0-999999999. It is best to keep your incremental number to as few digits as possible. The default is 2 if no specification is given.

Using the **New Client** page, you can also control the organizational unit in which the computer account objects are created. The default is the default account creation location as set in Active Directory. The following are your options:

Default directory service location This creates the computer account object for the client in an Active Directory location where all computer accounts are created by default during the domain join operation. The default Active Directory location is set to the **Computers** container in Active Directory. The client becomes a member of the same domain as the RIS server installing the client.

Same location as the user setting up the computer This creates the computer account object in the same Active Directory container as the user who is setting up the computer. For example, if you log on in the Client Installation Wizard and your user account currently resides in the **Users** Active Directory container, the client computer account object is created in the **Users** container in Active Directory.

A specific directory service location This creates the computer account object in a specific Active Directory container that you predetermine. It is assumed that most administrators will select this option to specify a container for all remote installation client computer account objects.

Client Response Options

The RIS settings on the **Properties** page control how the RIS server responds to remote boot–enabled clients requesting service. You can set the RIS server to **Respond to client computers requesting service** or only respond to known clients. When the RIS server is set to **Do not respond to unknown client computers**, it only responds to clients with a prestaged computer account object in Active Directory. This setting allows you to limit access to authorized clients that are prestaged in Active Directory, thereby increasing the security on your network. The **Do not respond to unknown client computers** setting also provides support for multiple third-party remote boot or installation servers on one physical network. For example, if your company already uses another vendor's remote boot or installation server, you cannot control which vendor's server answers the client's request. By setting the **Do not respond to unknown client computers** option in conjunction with pre-staging clients, you make sure that only those prestaged clients are serviced by authorized RIS servers.

Note If a user sets up the client, the user needs to have the appropriate rights to create the computer account in the domain or organizational unit chosen. For more information about granting computer account creation permissions to users, see Windows 2000 Server Help.

Pre-staging Clients in Active Directory Using GUID

You can also use the computer's GUID for pre-staging clients and making sure that each computer is uniquely identified. This unique ID is stored with the computer account object that is created when pre-staging the client. In most cases you can find the GUID for clients that are PC98 or Net PC–compliant in the system BIOS of the computer or on the outside of the computer case.

GUID Format

Valid characters for the client GUID are restricted to the hexadecimal characters 0-9 and A-F (uppercase or lowercase). You can enter the GUID in either "pretty print" or "raw byte order" format. However, combining the two formats causes RIS to not recognize the client.

Pretty Print

Pretty print format is as follows:

```
{dddddddd-dddd-dddd-dddd-dddddddddddd}
```

where *d* is a hexadecimal character. For example, {921FB974-ED42-11BE-BACD-00AA0057B223}. The dashes are optional and spaces are ignored.

Raw Byte Order

You can also enter GUIDs in "raw byte order," such as the byte order you get from a packet sniffer. In this case, do not include the curly brace and enter only the hexadecimal characters. The following GUIDs have exactly the same value:

- Pretty print:

  {12345678-1234-1234-1234-1234567890AB}

- Raw byte order:

  78563412341234112341234567890AB

Notice the first three parts of the pretty print GUID are in a different order than the raw byte format. This is how the computer stores the information internally and how it is sent on the network.

If you are having trouble with a prestaged client not being answered by a RIS server, make sure the GUID entered is either in pretty print format or raw byte order.

Clients Installing Operating System Images

Clients can also be granted permission to create their own computer account (non-prestaged) and install an image. This allows users to turn on their system, connect to the RIS sever, log on with their domain account, and be able to install an operating system image without assistance. To do this, the user needs the following permissions to the organizational unit that you have specified to hold the newly-created computer account:

- Read permissions
- Create computer objects

Users can also install an operating system image on their prestaged client if they have been granted the ability to read and write all properties on the specific computer object (not the container) that was created when the client was prestaged. The user also requires the ability to reset and change password rights on the computer object. (An administrator might need to reset the user account.)

Preboot Execution Environment

RIS uses the new Preboot Execution Environment (PXE) extensions to DHCP to initiate the installation of an operating system from a remote source to a client's local hard disk.

The PXE environment is built on a foundation of Internet protocols and services that are widely used in the computer industry. These include TCP/IP, DHCP, and TFTP. The PXE extensions to the DHCP protocol allows for information to be sent to network-bootable systems and allow these systems to find remote installation servers.

RIS Server PXE Environment

RIS uses DHCP as part of what is defined in the PXE architecture to initiate the process of remotely installing an operating system on a client. In other words, because PXE uses DHCP, so does RIS.

When a new DHCP PXE-based remote boot client is turned on for the first time, the client requests an IP address and the IP address of an active RIS server through the DHCP protocol and the PXE extensions to the DHCP protocol. As part of the initial request, as a DHCP option, the client sends out its GUID, which is used to identify the client in Active Directory. The client receives an IP address from the DHCP server and the IP address of the RIS server that services the client. In the RIS server's response, the client is given the name of a boot image that it must request when contacting the RIS server for initial service.

When the client makes its initial request for service, TFTP is used to download the boot image file to the client. In the case of RIS, this file is Startrom.com. Startrom.com prompts user to press the F12 key. If the user presses the F12 key, Startrom.com uses TFTP to download OSChooser, and presents the user with the Client Installation Wizard.

The process of initial communication between PXE clients and RIS servers can differ depending on how RIS is deployed in relation to DHCP services.

DHCP and RIS on Separate Servers

If you have DHCP and RIS on separate servers, the initial interaction between PXE clients and RIS/DHCP servers proceeds as follows:

1. DHCP discover from client (asking for IP address and PXE boot server).
2. DHCP offer from DHCP server (offers IP address and other network configuration settings).
3. DHCP offer from RIS server (offers PXE boot server).
4. DHCP request from client to DHCP server (requesting IP address).

5. DHCP acknowledge message from DHCP server (you can have this IP address).
6. DHCP request from client to RIS server (requesting the boot server).
7. DHCP acknowledge message from RIS server (this acknowledgment contains the address to the RIS server and the first file that the client needs to send a TFTP request to start the boot process).

Note If you configure the RIS server to respond only to known clients—that is, clients prestaged in Active Directory or previously installed computers—and the computer object is not located in Active Directory, the RIS server fails to respond to the client's DHCP request. If the RIS server is not on the same server as the DHCP server, and the server does not respond because the client is unknown, then the DHCP offer from the RIS server (in step 3 in the previous process) is not sent and therefore step 6 and step 7 do not occur.

DHCP and RIS on the Same Server

If the RIS server and the DHCP server are on the same computer, the conversation is as follows:

1. DHCP discover from client (asking for IP address and PXE boot server).
2. DHCP offer from DHCP/RIS server (offers IP address and PXE boot server).
3. DHCP request from client to DHCP server (requesting IP address, network configuration settings, and PXE boot server).
4. DHCP acknowledge from DHCP server (contains IP address and the RIS server IP and the first file to download).

Note If you configure the RIS server to respond only to known clients, and the computer object is not located in Active Directory, the RIS server fails to respond to the client's DHCP request. If the RIS server and DHCP server are on the same computer, the DHCP offer from the DHCP/RIS server (in step 2 in the previous process) only contains IP information and no information about any available servers to support the client's network boot process.

Verifying the Correct PXE ROM Version

When your Net PC or client containing a remote boot ROM starts, you see the PXE ROM message appear on the screen. The version of the PXE ROM code is displayed during the boot sequence of the client. RIS supports version .99c or later PXE ROMs. You might need to obtain a newer version of the PXE-based ROM code from your original equipment manufacturer (OEM) in case you do not succeed by using the existing ROM version.

Creating Operating System Images

Remote OS Installation allows the installation of operating system images on clients. You can create two types of images using RIS:

- CD-based images (using RISetup). CD-based images are similar to setting up a workstation directly from the Windows 2000 Professional compact disc; however, the source files reside across the network on available RIS servers.
- Remote Installation Preparation Wizard (RIPrep) images using RIPrep. RIPrep images allow a network administrator to clone a standard corporate desktop configuration, complete with operating system configurations, desktop customizations, and locally installed applications.

These two types of images are suitable for use in different situations. The benefits and limitations, as well as considerations and how to use RISetup and RIPrep are discussed in the following sections.

Note RIS only supports Windows 2000 Professional images, it does not support Windows 2000 Server or previous operating systems, including Windows 95 and Windows 98. RIS also does not support remote installation of the CD or RIPrep operating system images of Windows 2000 Server.

Using CD-based Images

A CD-based image is a copy of the contents of the Windows 2000 Professional operating system CD on the RIS server. Creating CD-based images is similar to creating a distribution share on a server that can be used to install the operating system over the network.

The benefits of using CD-based images are as follows:

- Easy to use.
- Enables standard, automated installation over a network.
- Can be customized by using answer files.
- Multiple answer files can be associated with a single CD-based image.
- Can be used for installation of all Windows 2000–compatible systems, regardless of hardware configurations.

The limitations of using RISetup are as follows:

- Images cannot contain preinstalled standard desktop applications.
- Takes longer to install on equivalent hardware than RIPrep images.

Creating New CD-Based Images

An initial CD-based image for Windows 2000 is created during the installation of RIS on a RIS server.

▶ **To add a new CD-based image to a RIS server**

1. You must be on the RIS server, where you want to add the CD-based image. In **Active Directory Users and Computers**, locate your RIS server by selecting the appropriate container, like the **Domain Controller** container.
2. In the right pane, right-click your RIS server, click properties, and then click the **Remote Install** tab.
3. Click the **Advanced Settings** button.
4. Click the **Images** tab, and then click **Add**.
5. Click **Add new installation image**. This option copies a Windows 2000 Professional CD-based image. Click **Next.**
6. In the **Remote Installation Services Setup Wizard**, click **Next**, then enter the installation source files location and proceed through the Wizard.

Or you can run the **risetup -add** command.

Figure 24.3 shows the directory structure where CD-based images are stored. You can define the name of the root folder (*imagename*) where the CD-based images are stored.

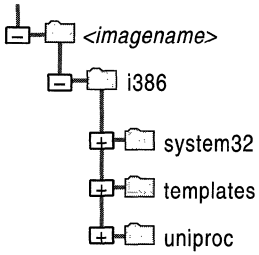


Figure 24.3 CD-based Image Directory Structure

Note CD-based images support additional directories. For example, you can add an \i386\OEM directory so that you can locate additional device driver files if you add additional hardware to a RIS client. For more information about creating an OEM subfolder, see "Automating Client Installation and Upgrade" in the *Deployment Planning Guide.*

For more information about creating a CD-based image, see Windows 2000 Server Help.

The Templates folder under each image contains the answer files for that image. By default, RIS creates a standard unattended answer file called Ristndrd.sif for each CD-based image, but multiple answer files can be associated with a single CD-based image. The default answer file can also be customized. For more information about working with answer files for CD-based images, see "Working with Answer Files" later in this chapter.

Note The \i386\lang directory is not copied to the RIS server as part of a CD-based image. This directory is needed by any client system that attempts to add the ability to read or write files in a language that is different than the default language that is installed. To allow client systems to be able to install language packs, copy \i386\lang and all subdirectories from the Windows 2000 Professional operating system CD to *RISServername*\Reminst\setup*clientlanguage*\images*imagename**i386**lang*

Modifying Properties of a CD-based Image

You can associate one or more answer files (.sif) to an existing CD-based image on the RIS server. This allows you to offer a variety of unattended Windows 2000 installation types from the same source image on the RIS server. You can also modify the settings within the answer file by using the setupmgr.exe tool. For example, you can change regional settings, video resolution, and network settings with answer files. For more information about modifying answer files, see "Working with Answer Files" later in this chapter.

Important If you purchased Windows 2000 from an authorized reseller, you must automate the process of entering the CD key. See "Specifying a CD Key in the Answer File" later in this chapter.

Using RIPrep Images

RIPrep images allow a network administrator to clone a standard corporate desktop configuration, complete with operating system configurations, desktop customizations, and locally-installed applications. After installing and configuring the Windows 2000 Professional operating system, its services, and any standard applications on a computer, the network administrator runs a wizard that prepares the installation image, and replicates it to an available RIS server on the network for installation on other clients.

The benefits of using RIPrep.exe are as follows:

- Allows for the creation of standard environment images including the operating system, standard desktop applications, and settings.
- Faster overall install times than CD-based images.

The limitations of using RIPrep.exe are as follows:

- Does not support multiple disks or multiple partitions on the source computer used to create the RIPrep image.
- Works only with the Windows 2000 Professional operating system.
- The destination client must have a hard disk that is the same size or larger than the system partition on the hard disk of the source computer used to create the RIPrep image.
- Target system must also have the same hardware abstraction layer (HAL).
- A CD-based image of the same version and language as the RIPrep source computer must also be installed on the RIS server.

RIPrep Considerations

If you plan to use RIPrep to create operating systems in your organization, keep the following considerations in mind:

- RIPrep supports replication of a single disk, single partition (the boot partition, which is usually drive C) of Windows 2000 Professional. Because of this, the operating system and all of the applications that make up the standard installation must reside on the boot partition prior to running the RIPrep wizard. By creating a RIPrep image, you can install and configure the operating system, locally-installed applications and configuration settings once for deployment to many clients.
- RIPrep-based images generally use more hard disk space than CD-based images because they contain an uncompressed copy of the client system's hard disk stored on the server. CD-based flat images still contain the compressed installation files that the CD contains.
- To store a RIPrep image on a server, you must also have a CD-based image that is the same version and default language stored on the same RIS server. This is because the answer file used for the RIPrep image also refers the client to the CD-based image for access to network adapter and text-mode boot drivers, in case the drivers required for the client installing the RIPrep image are different from those in the system used to create the RIPrep image. The text-mode setup then does an advanced *Xcopy* of the client's image to the client's hard disk drive. (Text-mode setup is the normal blue screen you see when installing Windows 2000 Professional that moves or copies all the files over before the graphical user interface setup begins.)

- One of the beneficial features of the RIPrep wizard is that the destination client, that is, the computer that is installing the RIPrep image, does not need to contain identical hardware as that of the source computer used to create the image. However, the hardware abstraction layer (HAL) drivers must be the same. The RIPrep wizard uses the new Plug and Play support that is included with Windows 2000 for detecting any differences between the source and the client hardware during image installation.
- To create a RIPrep image, a source computer is required. The source computer contains the Windows 2000 Professional operating system, locally-installed applications, and any configured system settings that represent a standard client configuration you want to deploy to the specific clients. Carefully configure and test this image before running the RIPrep wizard to create the RIPrep image. After the image is replicated to the RIS server, you cannot alter its configuration without rerunning the RIPrep wizard against the existing image.

Configuring a RIPrep Source Computer

To prepare and configure a source computer for a RIPrep image, use the following general steps:

1. Install Windows 2000 Professional on the boot drive of a newly formatted computer. Any of the common methods of performing a standard installation can be used.
2. During setup, create a single partition, and then set the partition to the minimum size required for support of the base operating system and any required applications. The size of the partition used on the RIPrep source computer determines the minimum disk size required on computers installing the resulting RIPrep image.
3. Configure all components and settings that represent the standard client configuration for this image, including:
   - Network settings
   - Security settings
   - User settings
   - Desktop settings

   Configure the source computer to adhere to any company configuration policies. For example, you might choose to define specific screen colors, set the background bitmap to a company-based logo, and set intranet proxy server settings in Microsoft® Internet Explorer 5.

4. Install all applications that you want to be a part of this image. If you have applications that are Windows Installer (.msi) packages and you want the applications to be installed in a managed state, see the following section. Install all the applications from the location that should be used when looking for updates or additional files, rather than a temporary location such as a local CD-ROM drive that is not available on clients installing the RIPrep image. For example, you can install Microsoft® Office 2000 and virus protection software for all users who require these applications on the computer.
5. Test the source computer to ensure that the configuration is exactly how you want it to be for the group of users who will access this image. RIPrep images cannot be modified after they are created, so if your image fails your test process, you must recreate the image or restore the existing RIPrep image, make the necessary adjustments, and run the RIPrep wizard again to create a new image that contains the additional changes. If it is appropriate, you can overwrite the existing image on the RIS server when you create the new image
6. Run the RIPrep wizard to create the RIPrep image on the server.
7. Configure user access to the image by setting permissions in the ristnrd.sif file in the Templates folder of the new RIPrep image. For more information, see "Setting Security Permissions in Answer Files" later in this chapter.

Using Software Installation and Maintenance with RIPrep

By using the Windows 2000 Software Installation and Maintenance features, you can install and manage key software in a RIPrep image by using the same methods you use to install the software on other computers in the organization.

Consider an organization that wants to bring in new computers and customize both the Windows 2000 operating system and the Office 2000 suite of applications. The organization has existing Group Policy objects to manage the computers in the organization, and the administrator has assigned Office 2000 to the computers in the appropriate Group Policy objects.

Note Be sure you configure the RIPrep source computer with applications from the same Group Policy objects that apply to the destination computers (those that install the RIPrep image) when they are deployed. The applications might be removed, or removed and reinstalled, if a different policy is applied to the computer when it is deployed.

The administrator installs the Windows 2000 operating system on a computer (that has the same HAL as the wanted target systems), and then configures the operating system the way that they want it. When Windows 2000 is installed and configured, the administrator adds it to the same Active Directory container where it stays after it is deployed. This container has a Group Policy object with Office 2000 assigned to the computer.

Note When you install Office 2000 as part of an RIPrep image, you must turn off 8.3 name creation. Change the value of the **NtfsDisable8dot3NameCreation** registry entry from 0 (default) to 1 in order to turn off 8.3 name creation. **NtfsDisable8dot3NameCreation** is located in HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\FileSystem. See the following procedure.

▶ **To turn off 8.3 name creation**

1. From the **Start** menu, click **Run**.
2. Type **regedt32.exe** or **regedit.exe**, and then click **OK**.
3. In the registry editor, navigate to:

   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\FileSystem
4. Select the **NtfsDisable8dot3NameCreation** entry.
5. To turn off 8.3 name creation, change the value of the **NtfsDisable8dot3NameCreation** registry entry from 0 (default) to 1. In Regedit.exe, right-click the entry, and then click **Modify**.

   -Or-

   In Regedt32.exe, click the entry, click **Edit**, and then click the appropriate menu choice.

Caution Do not use a registry editor to edit the registry directly unless you have no alternative. The registry editors bypass the standard safeguards provided by administrative tools. These safeguards prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your system. Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. To configure or customize Windows 2000, use the programs in Control Panel or Microsoft Management Console (MMC) whenever possible.

The administrator restarts the computer, and Software Installation and Maintenance installs Office 2000 (applications assigned to a computer install when the computer starts). After Office 2000 installation is complete, the administrator can take the computer running Windows 2000 with Office 2000 installed, and use RIPrep to build a Remote OS Installation image and put the image on a RIS server.

When the resulting RIPrep image is installed on destination clients, as long as the same Group Policy objects are applied to the destination computers, the applications remain in a managed state and can be managed, updated, or patched using the Software Installation and Maintenance features. It is recommended that you use Software Installation and Maintenance to install, update, and manage all applications that you install in RIPrep images.

For more information about software installation, see "Software Installation and Maintenance" in this book.

RIPrep and User Profiles

When creating RIPrep images, it is important to understand the relationship of user profiles, the changes made to a RIPrep source computer, and the wanted result for users who log on to computers that are installed by using the RIPrep image. Windows 2000 Logo–compliant applications properly separate user-specific and computer-specific configuration settings and data. Installing such applications for all users of the computer as part of a RIPrep source computer allows the applications to then be available to all users of clients that have the resulting RIPrep image installed later. Non-Windows 2000–compliant applications might perform or rely on per-user configurations that are specific to the profile of the user actually installing the application prior to running RIPrep (typically a local administrator), rather than to all users of the client. Such configurations remain specific to that user, which can result in the application or configuration setting not being available or not functioning properly for users of computers installed with the RIPrep image. In addition, some non-application configuration changes, such as the wallpaper specified for the user desktop, are applied only to the current user's profile by default, and are not applied to users of systems installed with the RIPrep image.

Thoroughly test any applications or configuration settings that you want to use in a RIPrep image to ensure that they will work properly with your organization's implementation of user profiles. To perform the test, make the change as one user (typically a local administrator of the computer), log off, and log on as a user account that is representative of your organization. If the changes you made are applied to the second user, the changes should also apply to users who log on to systems installed with an RIPrep image that contains the same change. To complete the test, create an RIPrep image, restore it to a different computer, and log on as a different representative user. Verify that the changes are applied and fully functional.

Some configuration settings can be copied directly from the profile to which they were applied (the local administrator in the previous example, for instance) to the All Users profile (such as the desktop wallpaper) some **Start** menu options, and shortcuts. However, all such changes must be tested carefully to verify that the profile copy process does not affect their functionality.

▶ **To copy the Administrator's profile to the Default User profile on the source computer**

1. Log on to the source computer as Administrator. Right-click **My Computer**, and then click **Properties**.
2. In the **System Properties** property page, click the **User Profiles** tab, select the local Administrator profile from the list of profiles on the computer, and then click **Copy To**.
3. In the **Copy To** dialog box, enter the path to the All Users profile folder (typically C:\Documents and Settings\All Users), and then click **Change**.
4. Select the appropriate group from the **User or Group** dialog box, typically the **Everyone** group, and then click **OK**.
5. Click **OK**, and then click **OK** again to exit the **System Properties** property page.

Running the RIPrep Wizard

After the client source computer is configured and fully tested, you are ready to run the Remote Installation Preparation Wizard (RIPrep.exe) from the RIS server that you want to receive this RIPrep image.

To run the RIPrep wizard, from the source client computer's **Start** menu, click **Run**, and then type:

\\<*RISservername*>\Reminst\Admin\i386\RIPrep.exe

Then click **OK**.

The RIPrep wizard does the following:

- Asks for the name of the RIS server on which to store the image. If none is selected, it defaults to the server name from which RIPrep.exe was run. If RIPrep.exe is run from *Server*\Reminst\Admin\i386\Riprep, server becomes the default location.
- Prompts for a subdirectory name to which the new image is created. The directory specified is created under the \RemoteInstall\Setup*OS Language*\Images directory on the specified RIS server.
- Prompts for a description and Help display text. These values are written to the answer file for the RIPrep image and used for display to users in the Client Installation Wizard.

- Prompts you to stop services and close applications on the client that RIPrep does not recognize. (This page does not prevent you from continuing, but you need to attempt to stop any services that RIPrep does not recognize as well as close any open applications.)
- Removes unique security identifier and other unique registry settings to prevent conflict between clients.
- Replicates the source client's system partition to a folder on the RIS server.
- Creates a default answer file named Riprep.sif. This associates a specialized unattended answer file with the image to ensure that the user is not prompted during image installation.

To create an image on the server, you must be logged on to the source client with an account that has backup privileges on the source computer. (If you are a member of the domain administrators group, you have this privilege.) Otherwise you need to log on as the local administrator.

Figure 24.4 shows the directory structure that exists after the RIPrep image is copied to the server. You can define the name of the root folder (*imagename*) where the RIPrep images are stored.

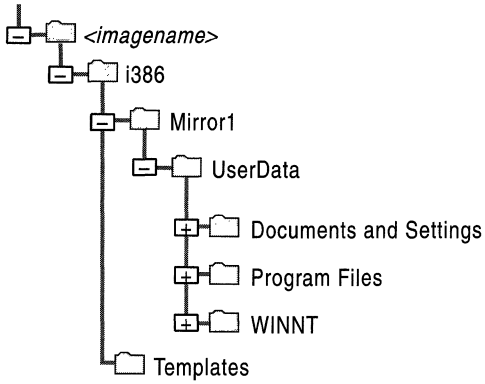


Figure 24.4 RIPrep Image Directory Structure

After creating the new image, the important files to note are RIPrep.log, Bootcode.dat, and Imirror.dat. These files contain the following information about the RIPrep image:

RIPrep.log This file contains log information about RIPrep.exe. RIPrep.log contains any errors that occur, such as encrypted files encountered or files in use. It also notes other information such as server name and description. This file resides in the \i386 directory on the server.

Bootcode.dat This file contains the boot sector for the system. Bootcode.dat resides in the \i386\Mirror1 directory.

Imirror.dat This file contains information, such as the drive letter, installation directory, and hardware abstraction layer (HAL) type, about the system that was cloned by using RIPrep. Imirror.dat resides in the \i386\Mirror1 directory.

These files contain some binary data, and therefore cannot be completely read by using a simple text editor or word processing application. After RIPrep is run, the source computer is shut down. If the source client is restarted, a mini-wizard runs and prompts the user for the unique configuration items that were removed when RIPrep was run.

After the replication of the image is complete, any DHCP PXE–based remote boot–enabled client, including those clients using the RIS boot floppy disk, can be used to access this image through RIS.

Relationship of SysPrep to Remote OS Installation

The System Preparation (SysPrep) tool works in conjunction with a third-party disk imaging process to prepare a source computer's hard disk for duplication to other computers. SysPrep is used to remove configurations unique to the computer, such as its computer name and security identifier (SID), so that the resulting computer image can be safely reused for installation on other computers.

SysPrep cannot be used with Remote OS Installation. However, the Remote OS Installation RIPrep wizard performs similar functions as SysPrep, which are combined with the other Remote OS Installation features to provide a full desktop imaging solution.

A comparison of using Remote OS Installation and SysPrep for creating system images follows.

The Benefits of Using Remote OS Installation with RIPrep

- Supports use of different hardware between the source and destination computers (both computers must have the same hardware abstraction layer).
- Does not require additional imaging software.
- Provides full support for replicating the source image to a server and restoring it onto destination computers.

The Limitations of Using Remote OS Installation with RIPrep

- Does not support replication of RIPrep images to sources other than RIS servers, such as removable media.
- Does not support creating Windows 2000 Server images.

The Benefits of Using SysPrep

- Supports creating both Windows 2000 Professional and Windows 2000 Server images.
- Supports use of third-party disk imaging and replication products.

The Limitations of Using SysPrep

- Requires the same hardware abstraction layer and mass storage controller on both the source and destination computers.
- Does not provide support for drive imaging, additional third-party tools are required for the imaging and replication processes.

Removing RIS Server Operating System Images

The **Remove** option, found on the **Images** tab of the **Advanced Settings** property page, only removes the unattended setup answer file (.sif) that is associated with the operating system image.

To completely remove the operating system image from the RIS server, use Windows Explorer to delete the directory and subdirectories containing the image that you want to remove. You can do this from the console of the server or by browsing the Reminst share on the RIS server.

Note Do not remove the physical operating system image completely until all clients have upgraded to the new version of the operating system. Be sure that you back up the unattended setup answer file (.sif) prior to removing any setup answer files.

Working with Answer Files

Unattended installation of Windows 2000 involves running Windows 2000 Setup with a customized script, called an answer file (.sif). These small text files contain the information that Windows 2000 otherwise asks the user to enter when installing or upgrading. Answer files do not always eliminate the need for user input. The information stored in answer files determines the level of input required during installation. If you want to create an unattended answer file, set the parameters so that user interaction is not required.

The process of working with answer files for use in Remote OS Installation includes the following steps:

- Creating a setup answer file (.sif)
- Associating an answer file with an operating system image
- Controlling which users have access to the different setup answer files

Creating and Modifying Setup Answer Files

Windows 2000 includes a tool called Setup Manager to create unattended setup answer files for use with standard Windows 2000 setup and with Remote OS Installation. You can use the Setup Manager Wizard or create your own answer files manually by using a text editor such as Notepad. You can also combine using Setup Manager to generate the basic answer file and then use a text editor to further customize the answer file to meet your needs.

For more information about general Windows 2000 setup operations and how setup uses answer files, see "Automating Client Installation and Upgrade" in the *Deployment Planning Guide*.

For more information about manually working with answer files and sample answer files, see "Sample Answer Files for Unattended Setup" in the *Deployment Planning Guide*.

For the Setup Manager tool, information about how to use it, and a complete list of answer file parameters and their syntax, see the deploy.cab file in the \Support\Tools folder on the Windows 2000 Professional operating system CD.

Modifying Remote Installation Answer Files

A standard answer file named ristndrd.sif is created in the Templates folder of each operating system image installed on a RIS server.

The standard answer file performs an unattended installation, and also contains the description that is displayed during the Client Installation Wizard for selecting an operating system image to install.

You can have more than one of these files present in the Templates folder of an operating system image. For example, you might have different installation options for your marketing, sales, and accounting departments that each require a different answer file. Associated multiple answer files with a single operating system image allows you to use one set of Windows 2000 Professional source files to provide multiple installation options instead of creating more images. Remote OS Installation answer files must end with the .sif extension.

For more information about the answer file parameters specific to Remote OS Installation, see the deploy.cab file in the \Support\Tools folder on the Windows 2000 Professional operating system CD.

In addition, there is a set of answer file settings specific to Remote OS Installation that must be included with every answer file to be used with RIS, and that cannot be modified. When creating answer files using Setup Manager these settings are added automatically. When working with answer files manually, ensure that these parameters are included exactly as shown here:

```
[data]
floppyless = "1"
msdosinitiated = "1"
OriSrc = \\%SERVERNAME%\RemInst\%INSTALLPATH%\%MACHINETYPE%
OriTyp = "4"
LocalSourceOnCD = 1

[SetupData]
OsLoadOptions = "/noguiboot /fastdetect"
SetupSourceDevice =
"\Device\LanmanRedirector\%SERVERNAME%\RemInst\%INSTALLPATH%"
```

Associating an Answer File with an Image

After you have created your answer file, you need to associate it with an existing CD-based image. You cannot associate additional answer files with RIPrep images.

▶ **To associate an answer file with an existing CD-based image**

1. From the **Start** menu, point to **Programs** and **Administrative Tools**, and then click **Active Directory Users and Computers**.
2. Right-click the name of the RIS server, and then click **Properties**.
3. In the **Properties** dialog box, click the **Remote Install** tab, and then click **Advanced Settings**.
4. Click the **Images** tab, and click **Add**. The **Add** wizard is displayed.
5. Click **the Associate a New Answer File to an Existing Image** option.

6. On the **Location of Answer File** page, browse to the location and answer file that you want to associate with the image, and click **Next**.
7. On the **Select an Installation Image** page, select the image with which you want to associate the new answer file, and then click **Next**.
8. On the **Friendly Description and Help Text** page, enter the description and Help text you want displayed to users in the Client Installation Wizard for this image. Click **Next**.
9. On the **Review Setting** page, confirm your selections, and then click **Finish** to complete the process and copy the answer file to the selected image's Templates folder.

The friendly description and Help text are used to describe the available operating systems within the Client Installation Wizard. For more information about the Client Installation Wizard, see "Client Installation Wizard" later in this chapter.

Setting Security Permissions in Answer Files

By setting explicit user or group security permissions on the answer file (.sif) for a particular operating system image, you can determine which operating system options are displayed to users in the Client Installation Wizard. You can allow a user of Remote Installation Services to choose from all of the operating system images available on a particular RIS server, or you can restrict the user to only the images appropriate for that user.

▶ **To restrict access to the available operating system images**

1. Locate the Templates folder of the desired image on the RIS server, typically:

   `\\`*`servername`*`\RemInst\Setup\`*`OSLanguage`*`\Images\`*`OSImageName`*`\i386\Templates`

   Note Each CD-based operating system image that you add to a RIS server has an associated Templates directory that contains the unattended setup answer file or files associated with that image.

   Set specific access permissions on the individual unattended setup answer files that are contained in this directory. If you have not associated additional unattended answer files to the base operating system image, you only see the default answer file (Ristndrd.sif) in the \Templates directory.

2. Right-click the answer file, and then click **Properties**.

3. In the **Properties** window, click the **Security** tab.

   Warning The default security permissions allow the **Everyone** group (which includes all users) access to this operating system image from within the Client Installation Wizard. To restrict access to this operating system image, select the **Everyone** group, and then click **Remove**.

4. Click **Add**, and then select the security group or individual users who you want to receive access to the selected operating system image. Click **Add**, and then click **OK** twice. The default permissions that are set for each user or security group are sufficient for use with Remote Installation Services (the user requires only read permissions).

   Unless you specify users or groups of users to be given access, no users have access to the selected available operating system image.

Because selecting individual users for specific access can greatly increase the administrative workload, it is recommended that you group your users by security group and apply the security group to the answer file when restricting access to operation system images. If you add users to the security group, they will have access to the correct operating system image. For example, if all users in the sales department need access to the same list of operating system choices, you can group all sales staff users in a sales security group in Active Directory. You can then allow the sales security group read access to the operating system choices they need by granting the sales security group access to the appropriate answer files. When a new member of your company joins the sales department, add that user to the sales security group so that they can see the same operating system choice listings as the other members of the sales security group.

How Answer Files Are Used During Remote Installation

Even though only CD-based images can have multiple answer files associated with them, both CD-based and RIPrep images use answer files during the installation process, and the handling of an answer file for a specific installation is handled the same for both image types.

Using answer files with RIS also allows OSChooser (OSC) variables to be embedded within them. (For more information about OSC variables, see "OSCML and Client Installation Wizard Variables" in this book.) OSC variables are replaced with user, server, installation, or client-specific data, and then a completely configured answer file specific to the individual installation attempt is written to a temporary location (the \Reminst\Tmp folder) on the RIS server. The file name of the temporary answer file is the raw GUID in hexadecimal with .sif appended. For example, if the client has a GUID of {78563412-3412-3412-1234-123456789AB}, the raw format is 12345678123412341234123456789AB and a generated answer file name is "12345678123412341234123456789AB.sif." The client is then instructed to use this answer file to control the installation of the image. At the end of text-mode setup, setup deletes the temporary answer file.

Note Because the location where the temporary answer file is created is open for anyone to read, embedding passwords (such as the administrator's password) into an answer file is a security risk.

Specifying a CD Key in the Answer File

Users might be prompted to enter a CD key during setup, depending on the type of Windows 2000 product you use to create Remote OS Installation images (retail, select, or OEM).

Note The CD key, even though it is specified as ProductID in the answer file, is different from a system's product identification number (PID). A unique PID is always generated for clients during installation, adding the appropriate CD key to an image's answer file will not result in system's being installed with duplicate PIDs.

The default answer files created by Remote OS Installation do not include a CD key. If CD-based images created from the retail version of the Windows 2000 Professional operating system CD or images created by using the Remote Installation Preparation Wizard (RIPrep.exe) on systems originally installed by using a retail version of Windows 2000 Professional are used, during operating system installation the user is prompted to enter a CD key. To avoid the user being prompted for a CD key, locate the CD key for the copy of Windows 2000 Professional in use and add it as the ProductID value in the [UserData] section of the answer file.

When working with the select or OEM versions of the Windows 2000 Professional operating system CD, setup does not prompt the user for the CD key.

Important All copies of Windows 2000 Professional and any additional software installed by using Remote Installation Services must be properly licensed, and it is the licensee's obligation to ensure that it is licensed to make such copies.

Client Installation Wizard

The Client Installation Wizard is a text-based tool that guides the user through the remote operating system installation process. The Client Installation Wizard is the first image a user sees after pressing F12 and successfully connecting to the RIS server for a network service startup.

When the client makes its initial request, TFTP is used to download the file requested by the client. In the case of RIS, this file is Startrom.com. Startrom.com prompts users to press F12. If the user presses F12, Startrom.com uses TFTP to access OSChooser, and presents the user with the Client Installation Wizard.

Using Client Installation Wizard to Install Clients

The Client Installation Wizard supports two types of operating system images: CD-based flat images and RIPrep image formats. When a new remote boot-enabled client is started for the first time, the user selects the F12 key to initiate the download of the Client Installation Wizard.

Both the CD-based and RIPrep image formats can be displayed in the list of available installation options on the screen. The installation process is initiated when the user selects one of the operating system image options.

The actual screens displayed to users during the Client Installation Wizard can be completely customized to reflect an organization's internal requirements, such as Help desk contact information or other settings related to operating system installations. You can configure RIS servers to guide users through a successful operating system installation without requiring them to select a specific operating system image, or you can give them the flexibility to select the operating system of their choice.

The following sections describe the default implementation of the Client Installation Wizard by RIS. Details about the installation options that an administrator can choose to present to users during the initial network service request are also included.

Default Client Installation Wizard Process

When you install RIS and run RISetup.exe, a default set of the Client Installation Wizard screens are installed. Figure 24.5 illustrates the sequence that a user might follow when he or she installs a remote operating system by using the default Client Installation Wizard screens.

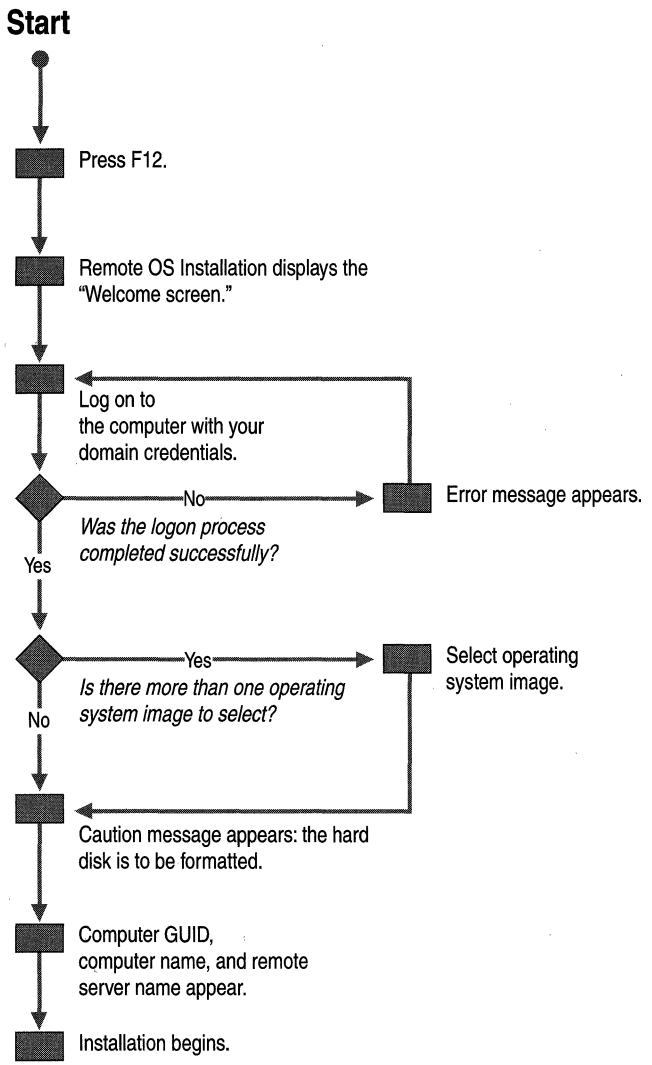


Figure 24.5 Default Client Installation Wizard Screens Displayed

Client Installation Wizard Screens

Table 24.2 describes the Client Installation Wizard screens that are downloaded when a new remote boot–enabled client is started for the first time. The file names and any customization information are listed as well.

Table 24.2 Client Installation Wizard Screens

Screen	Filename	Description and Customization
Welcome screen	Welcome.osc	The first screen displayed to the user. You can customize this welcome screen to display a specific company message, or to provide information to users prior to installing the operating system. You can also modify the welcome screen to support multilanguage choices. For more information about multilanguage choices see "Multilanguage RIS Servers" later in this chapter.
Logon screen	Login.osc	Requires a user to log on. The user logs on to the network using their existing user account, password, and domain. After the user successfully logs on, RIS uses the credentials entered by the user to determine which installation options should be displayed to the user on the Setup Options screen, based on the RIS Group Policy settings that apply to the user. If the process is not successful, and the logon account, password, or domain is not recognized, the user is prompted to log on again.
Setup Options screen	Choice.osc	Displays setup options to the user: Automatic, Custom, Restart, Maintenance and Troubleshooting. The degree to which this screen and its options are displayed is controlled through RIS Group Policy settings. The other possible setup options available to users through this screen are described in "Controlling Client Setup Options" later in this chapter.
	OSAuto.osc	This screen is not displayed to users. Performs a check to see if a computer account object already exists in Active Directory with the same GUID as the computer running the Client Installation Wizard. If a duplicate GUID is found, DupAuto.osc is displayed. If no duplicate GUID is found, OSChoice.osc is displayed.

(continued)

Table 24.2 Client Installation Wizard Screens ***(continued)***

Screen	Filename	Description and Customization
Error screen	DupAuto.osc	If a duplicate GUID is found in Active Directory, the user is presented with this screen. This screen instructs the user to contact the network administrator.
Operating system choice screen	OSChoice.osc	Displays the list of operating system images on the RIS server that are available to the logged-on user. If there is only one possible image for the user to install, then that image is selected and the user does not see this screen. (When RIS is initially installed, the user only has one choice on the screen.) When an operating system image choice is added to a RIS server it is available for all users who receive service from that server. For information about controlling access to operating system images, see "Setting Security Permissions in Answer Files" earlier in this chapter. Depending on how you choose to implement Remote OS Installation in your organization, you might prefer to automate users' selection of operating system images or to allow users to select the operating system image that best fits their role in the company. Consider carefully how these different methods suit your organization's desktop management philosophies prior to implementing the Remote OS Installation feature.
Caution screen	Warning.osc	The caution screen displays a warning message that the hard disk will be formatted. The user is cautioned that an operating system is about to be installed on this computer, which requires the local hard disk to be repartitioned and formatted, thus erasing all data that currently reside on the disk.
Summary screen	Install.osc	The summary screen displays information about the computer including: computer name, computer GUID, and the RIS server to be used for downloading the image. Pressing any key begins the installation process. At this point, the RIS server has created a computer account object in Active Directory for this computer. The RIS server is now able to look up this computer and associate the computer name and other settings if this computer is reinstalled. If you were running Client Installation Wizard to pre-stage the computer for another user, you can now shut down the computer and give the computer back to the end user. The end user needs to reset password permissions on the newly created computer account object in Active Directory.

After the summary screen appears, the user exits the Client Installation Wizard and proceeds to the automated installation process. This is similar to the installation process of installing from a CD, but it is automated and occurs while the operating system files are stored on the RIS server. Depending on the speed of your network and the load on the RIS server, this process can be much faster than an installation from a CD-ROM drive. (This process can also be customized to present the user with additional options.)

Warning All data is erased from your hard disk drive when you install a new operating system.

Controlling Client Setup Options

There are four installation options available to a user of the Client Installation Wizard on the setup options screen (Choice.osc). They are:

- Automatic setup
- Custom setup
- Restart a previous setup attempt
- Maintenance and troubleshooting

A user's available installation options are determined by RIS Group Policy settings in conjunction with the user's specific security group or user account. For example, you might choose to provide all members of the Help desk security group with access to all of the installation options, but restrict general network users to only the Automatic Setup and Maintenance and Troubleshooting options. This prevents confusion and helps to guide users to the correct choices for operating system installation.

Automatic Setup

This option provides the easiest operating system installation path. If there is already a computer account object in Active Directory with a GUID that matches the client's GUID, the client is named the same as the existing computer account, and the existing computer account is reused. If a matching GUID is not found in Active Directory, the client is named based on the automatic naming format configured in the properties of the RIS server, and a new computer account is created in the location specified by the RIS server. For more information about changing the automatic naming format or the location in which to create automatic computer accounts, see "Configuring RIS Servers" earlier in this chapter.

Custom Setup

This option allows users to override the automatic computer naming process, as well as the default location within Active Directory where client computer account objects are created. Because the default installation of RIS is configured to generate computer names based on which user logs on to the client, it is not appropriate to use the Automatic Setup option for setting up another user's computer. The Custom Setup option is similar to the automatic option, but it gives you the ability to set up a client for other users. For example, Help desk personnel can use this option to preinstall an operating system on a client within the enterprise, prior to delivery to the user. If either the computer name or computer location box is left blank on the Custom Setup screen, then the automatic name or location is used.

Warning If the name and location entered in the Custom Setup screen and the GUID of the client match the name, location, and GUID of an existing computer account object, the existing computer account object is reused. If only one of the fields between GUID, name, and location match, a duplicate name or duplicate GUID error screen is presented. However, because the user can bypass the error screen, do not use the Custom Setup option when you are creating prestaged computer accounts for use in installations performed directly by users.

Restart a Previous Setup Attempt

If the installation process fails or network connectivity is disrupted during the initial text-mode portion of setup (prior to completion of the file copy stage), a Restart Setup command is available for optional display to the user the next time the computer is started.

If this option is selected, it restarts the operating system installation process by using the information entered during the previous attempt. This option is controlled by the existence of the client-specific temporary answer file generated by RIS, which is deleted prior to running the graphical user interface portion of setup.

Maintenance and Troubleshooting

This option provides access to maintenance and troubleshooting tools, such as system flash BIOS updates and computer diagnostic tools, that can be used prior to operating system installation.

Client Installation Wizard Error Screens

There are additional screens in the \Oschooser subdirectory that are displayed when errors occur, such as when an incorrect user account or password is entered.

When RIS encounters an error, the error code (such as 20008 for the logon error) is changed into a hexadecimal value (00004e28) and ".osc" is appended to produce a error screen file name ("00004e28.osc"). RIS then checks the language directory in use for a screen file of that name. If an error screen is not found, an internally generated error screen is displayed. The internally generated screen is not customizable.

Error screens use the same OSCML syntax as non-error screens, and can be customized using the same methods.

Customizing Client Installation Wizard Screens

You can customize the Client Installation Wizard screens to meet the needs of your organization. You can add or remove entire screens or individual options within screens, and you can modify the text displayed to users to include information such as Help desk phone numbers or corporate policies and procedures. You can also build new, custom screens that prompt the user for information, which you can then use to control the installation of the image.

When Remote Installation Setup Wizard (RISetup.exe) is run, the default set of the Client Installation Wizard screens are installed at \RemoteInstall\Oschooser*<language>*. The files for these screens (sometimes referred to as OSC screens) are simple text files with the file name extension .osc, and are in the format of the OSChooser Markup Language (OSCML). OSCML is modeled after Hypertext Markup Language (HTML) version 2.0 specifications. For more information about OSCML, see "OSCML and Client Installation Wizard Variables" in this book.

Adding Screens and Working With OSC Variables

One way that you can customize Client Installation Wizard screens is by creating additional screens that prompt the user for custom information required to perform the installation. Any settings that you can specify in an answer file you can create prompts for and set with OSC variables in the Client Installation Wizard. Remote OS Installation replaces the variables in the answer file with the values entered in the Client Installation Wizard screens when it creates the temporary answer file for the client installation. In most cases, this allows you to have greater flexibility when automating a setup than having to create an answer file for each possible scenario.

OSC variables are status variables and variables collected through user input in Client Installation Wizard screens. You can add variables and have them used in other Client Installation Wizard screens, or have them inserted in the generated answer file that the client uses when installing the image. You can use up to 64 unique variables per client session; however, there are some variables that are reserved and are set or used internally by RIS.

OSC variables cannot be defined prior to the user logging on. The only exception to this rule is the *%language%* variable, which can only be set (not replaced in outgoing screens) prior to a successful logon process. This exception allows users to select the desired language to proceed in. For more information about supporting multilanguage users, see "Multilanguage RIS Servers" in this chapter. After the user successfully logs on, all communication from the client to the RIS server is encoded, making it safe to send passwords and other confidential information across the network. At this point, any OSC variable can be entered and replaced in outgoing screens.

The following sections describe how you can modify the default Client Installation Wizard screens to add an additional screen that prompts users to enter settings for their video display. To do this, you need to modify one existing screen, OSChoice.osc, and create one new screen, Display.osc. After creating and modifying these files you can then modify the answer files to work with the new OSC variables.

Warning As soon as a modified or new OSC file is placed on the server, users can see it. Be sure to test your changes carefully on a nonproduction RIS server before making changes to RIS servers that are supporting users in the production environment. In addition, changes to OSC files must be performed on all wanted RIS servers individually. There are no capabilities provided to synchronize changes in OSC files across multiple RIS servers.

Modifying the OSChoice.osc File

To enable the display of the new screen, modify the OSChoice.osc screen to display the new display settings screen (Display.osc) when ENTER is pressed, instead of the default Warning.osc screen. No changes are required to the text or options displayed by OSChoice.osc, only the screen that is displayed next must be changed.

▶ To modify the OSChoice.osc file

1. Open the OSChoice.osc file in Notepad.
2. Change the existing tag:

   ```
   <FORM ACTION="WARNING">
   ```

   to

   ```
   <FORM ACTION="DISPLAY">
   ```

After this change is made, the Display.osc screen is displayed as the next screen. The following is the OSCML text for the modified OSChoice.osc file that you create in Notepad:

```
<OSCML>
<META KEY=F3 ACTION="REBOOT">
<META KEY=ESC HREF="CHOICE">
<META SERVER ACTION="ENUM IMAGES">
<TITLE>  Client Installation Wizard
OS Choices</TITLE>
<FOOTER>  [ENTER] continue            [ESC] go back             [F3]
restart computer</FOOTER>
<BODY left=5 right=75>
<BR>
<BR>
Use the arrow keys to select one of the following operating systems:
<P left=8>
<FORM ACTION="DISPLAY">
<SELECT NAME="SIF" SIZE=12>
%OPTIONS%
</SELECT>
</FORM>
</P>
<BOLD>Description:</BOLD>  
<TIPAREA>
</BODY>
</OSCML>
```

Creating a New Display.osc Screen

The Display.osc file allows the user to change the default video display settings to be used on the client. Each input control is preset with a default value, and example values are displayed on the screen. The result of each input is stored in an OSC variable. The following procedure shows a way that you can modify a Display.osc screen.

▶ To create a new Display.osc screen

1. Use Notepad to create a new file named Display.osc.
2. Add the tags and display text as shown, and save the file:

```
<OSCML>
<META KEY=F3 ACTION="REBOOT">
<META KEY=ESC HREF="CHOICE">
<TITLE>  Client Installation Wizard
Custom Setup</TITLE>
<FOOTER>  [ENTER] continue            [ESC] go back      [F1] help
[F3] restart computer</FOOTER>
<BODY left=5 right=75>
<BR>
<BR>
Please enter your desired Display settings or accept the defaults
below.
<BR>
<BR>
<FORM ACTION="WARNING">
X resolution: <INPUT NAME="X-res" VALUE="640" SIZE=4 MAXLENGTH=4><BR>
Y resolution: <INPUT NAME="Y-res" VALUE="480" SIZE=4 MAXLENGTH=4><BR>
Refresh rate: <INPUT NAME="Refresh" VALUE="60" SIZE=3
MAXLENGTH=3><BR>
<BR>
</FORM>
<BR>
<BOLD>Examples:</BOLD><BR>
X resolution: 1024<BR>
Y resolution: 768<BR>
Refresh rate: 70<BR>
</BODY>
</OSCML>
```

Figure 24.6 shows how the display.osc screen appears in the Client Installation Wizard.

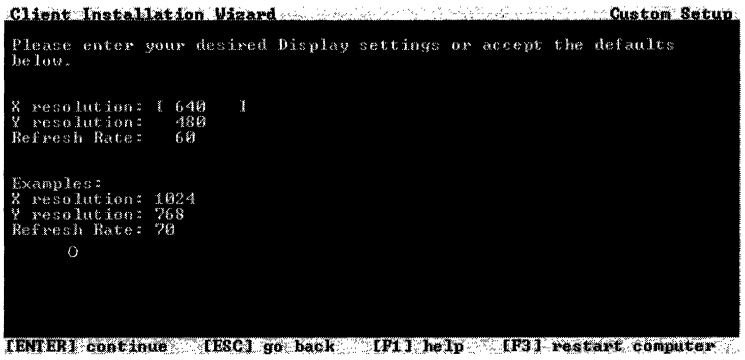


Figure 24.6 Client Installation Wizard Results for Display.osc

Modifying the Answer Files to Work with the New OSC Variables

When you modify screens and create new OSC variables, you also need to update the answer files to accept the new variables. Modify each answer file that can be selected by the user while using the custom screens. The following example creates variables for the X and Y resolution of the display ("X-Res" and "Y-Res") and the screen refresh rate ("Refresh").

OSC variables in the answer file are identified by % signs around the value. For example, Refresh is referenced as *%Refresh%* in the answer file. Use the following procedure to modify the related entries in the default RIS answer file to allow use of the values entered during the Client Installation Wizard instead of static values specified in the answer file.

▶ **To modify files to work with the new OSC variables**

1. Use Notepad to open the default RIS answer file (located in the i386\Templates folder of each image on the RIS server).
2. Change the values as shown for the following entries:

```
[Display]
XResolution = %X-Res%
YResoultion = %Y-Res%
VRefresh = %Refresh%
```

With these modified files in place, the user is prompted to adjust the default video display settings. The user's input is incorporated into the answer file that is used during installation.

Note When creating OSC variables, be sure that you do not use variable names reserved for use by RIS internally. For a list of those used internally by RIS, see the following section.

Reserved OSC Variables

Table 24.3 lists the reserved OSC variables and a description of what each one does. Some variables can be useful for you to use within your own custom screens and answer files. Some of the values do not have defined defaults. Some are only generated when the variable was not previously set and is required for the installation process to continue. In these cases, the RIS server generates the variables prior to starting the installation (they are only available in the Install.osc screen and in the answer file).

Table 24.3 Reserved OSC Variables

Variable	Description
Language	The only OSC variable that can be set prior to the user successfully logging on. This variable controls which folders are used to obtain Client Installation Wizard screens, to generate the list of available operating system images, and when performing any ENUM actions. The default value of this variable matches the default language of the server. For more information about language support, see "Multilanguage RIS Servers" later in this chapter.
SubError	The server sets this variable internally for any errors it encounters. This can be added to an error screen to help administrators diagnose internal server failures.
MachineOU	Indicates to the server where the new computer account is to be generated. If this variable is not set prior to installation, the setting specified in the properties of the RIS server is used.
MachineName	Indicates to the server the name of the new computer. If this variable is not set prior to installation, a name is generated based on the computer naming policy specified in the properties of the RIS server. By default, this variable is used to set the ComputerName value in the answer file.
ServerName	Set to the name of the server to which the Client Installation Wizard is connected. By default, this variable is used to set the OriSrc value in the answer file.

(continued)

Table 24.3 Reserved OSC Variables *(continued)*

Variable	Description
Language	The only OSC variable that can be set prior to the user successfully logging on. This variable controls which folders are used to obtain Client Installation Wizard screens, to generate the list of available operating system images, and when performing any ENUM actions. The default value of this variable matches the default language of the server. For more information about language support, see "Multilanguage RIS Servers" later in this chapter.
ServerDomain	Set to the domain name of the server to which the Client Installation Wizard is connected.
BootFile	Set when a tool is about to be opened.
NetBiosName	The generated NetBIOS name of the new computer.
Siffile	The path of the answer file (.sif) for the operating system image that the user selects, in the form of X:\RemoteInstall\Setup\English\Images\Win2000.pro\i386 \Templates\Ristndrd.sif
Options	This variable is filled with the results of an ENUM action performed by the RIS server. It contains OSCML and is typically placed between a <SELECT> tag and a </SELECT> tag. (See Tools.osc for an example.)
CheckDomain	Used internally. Set to 1 or 0 to indicate verification that the user domain has been processed.
MachineDomain	The domain that the new computer attempts to join during setup. By default this variable is used to set the JoinDomain value in the answer file.
SysPrepPath	Used for RIPrep images to indicate the path to the image source files. Example: Setup\English\Images\Win2000.prep\i386
InstallPath	The TFTP relative path to the installation image. Example: Setup\English\Images\win2000.pro. By default, this value is used to set the OriSrc, SetupSourceDevice, LaunchFile, and InstallFilesPath values in the answer file.
SysPrepDrivers	The path used to find Plug and Play drivers, determined by the RIS server as the best fit for a RIPrep image. If the RIS server cannot make a determination, it is set to blank.
MAC	Sent by the Client Installation Wizard to indicate the media access control (MAC) address of the client. This is sent right after the user logs on and before the next screen is viewed.

(continued)

Table 24.3 Reserved OSC Variables *(continued)*

Variable	Description
GUID	Sent by the Client Installation Wizard to indicate the GUID address of the client. This is sent right after the user logs on and before the next screen is viewed.
MachineType	Sent by the Client Installation Wizard to indicate the type of hardware on which OSChooser is running. This is sent right after the user logs on and before the next screen is viewed. By default, this variable is used to set the OriSrc, LaunchFile, and InstallFilesPath values in the answer file.
UserName, Password, UserDomain	The values entered by the user in Login.osc. These values are used by the RIS server when performing functions specific to the logged-on user, such as determining Group Policy settings and available operating system images. Password is not available to OSC or SIF files.
UserFullName	Name generated by the BINL service that is the user's first name followed by the user's last name. By default, this variable is used to set the FullName value in the answer file.
UserFirstName	The first name of the user who has entered their credentials.
UserLastName	The last name of the user who has entered their credentials.
TimeZone	Set by the server to the server's time zone setting. By default, this variable is used to set the Timezone value in the answer file.
OrgName	Defaults to the organization name set on the RIS server. By default, this variable is used to set the OrgName value in the answer file.

Multilanguage RIS Servers

To assist administrators supporting multiple languages, it is possible for a single RIS server to service clients that install different languages versions of the Windows 2000 operating system.

The Welcome.osc screen in the root of the \Oschooser directory in the RIS directory tree is the first file sent to the Client Installation Wizard. By modifying this screen, you can provide a list of languages that you want the server to support. The Multilng.osc file (in that same directory) is a sample OSC file that prompts the user to select their desired language. Subsequent Client Installation Wizard screens are obtained from the specified language folder in the \Oschooser directory, and the list of operating system images presented to the user is restricted to those images in the language selected.

For each language made available to users, a set of Client Installation Wizard screens must also be made available in a separate folder in the \Oschooser folder (\Oschooser\English is supplied by default).

Language Restrictions

The Client Installation Wizard does not support non-101 key keyboards, non-OEM fonts, or Multi-Byte Character Set (MBCS)/Unicode character sets. These limitations apply to data used within the Client Installation Wizard, such as computer names, domain names, and directory and file names (such as answer files). It also applies to what can be displayed to users, such as example or descriptive text, and what users can input to the Client Installation Wizard, such as user names, passwords, and domain names.

Because of these restrictions, ensure that the common strings of users accessing RIS servers, such as domain names, user names, and user passwords, do not contain characters that cannot be used within the Client Installation Wizard.

In addition, even though a different set of Client Installation Wizard screen files can be used depending on the language selected, in many cases creating screens that are fully localized to the selected language is not possible using the available character sets.

▶ **To enable language selection in the Client Installation Wizard**

1. Using a text editor such as Notepad, open the Multilng.osc file, located in the \Oschooser directory.
2. Modify the <OPTION> tags to match the list of the languages supported by the server.
3. Save the modified Multilng.osc over the Welcome.osc file. (A copy of the original Welcome.osc file is found in the language-specific directory. For example, the English language directory is \Oschooser\English.)
4. Create folders for each supported language in the \Oschooser folder.
5. Create the Client Installation Wizard screens required for each language. If you want, these can be copied directly from the supplied \Oschooser\English folder.

When you complete these steps, users who perform a network service boot will receive the multilanguage welcome screen. Subsequent Client Installation Wizard screens will be obtained from the \Oschooser\<*language*> folder that matches the selected language. The selected language limits the list of image options displayed to the user.

Note To support RIPrep images of different languages, a CD image of the language must also exist on the RIS server. This is required to supply files such as device drivers that are required during installation of the RIPrep image but that were not a part of the original RIPrep source computer.

Single Instance Store

Single Instance Store (SIS) allows multiple images on the same RIS server to consume less disk space by combining duplicate files. This is the case for any type of operating system image files whether they are CD-based or RIPrep files on the RIS volume. This is especially useful when you store multiple RIPrep images on a RIS server. Usually, only 10 percent of the files (excluding installed applications) are different from the installed standard CD-based image. This saves a significant amount of disk space.

RISetup installs the two pieces that make up SIS: the SIS filter driver and the SIS Groveler. SIS is not necessary for RIS to work properly, and SIS can work independently of RIS. SIS can operate on any local volume except boot and system partitions.

Note The SIS filter driver is not removed when the rest of Remote Installation Services is removed. There might be files on the system that have been converted to SIS reparse points, and removing the driver causes those files to be inaccessible.

Single Instance Store Groveler

When the SIS Groveler starts, it searches the root of each NTFS volume in the system to see if it contains the SIS directory **SIS Common Store** and a file called **MaxIndex** within that directory. If it finds these items, and the SIS filter driver is installed on the system, the Groveler knows to search for and consolidate duplicate files on the volume.

The SIS Groveler does most of its work when the system is not busy. It uses the same technology that the Indexing Service (a service that indexes your volumes for quick search capabilities) uses to not consume CPU time when the system cannot afford it. If the disk space on the volume drops below a specified value, the Groveler increases its CPU usage regardless of system activity to help prevent the possibility of running out of space on the volume.

A side effect of the Groveler's intelligent CPU use is that the service does not run at full speed during the first several hours after installation, even if the system is idle. This is because the service attempts to determine how much CPU and input/output (I/O) bandwidth it can use without causing problems to other system components. If you want the Groveler to run at maximum capacity do the following:

▶ To make SIS Groveler run at maximum capacity

1. Expand grovctrl.ex_ from the Windows 2000 Server operating system CD. This file is located in the \i386 directory on the CD.
2. Run **grovctrl f** to force the Groveler into foreground mode for all drives.

After the Groveler completes its work in foreground mode, it resumes normal operation where it intelligently uses the CPU cycles.

When scanning a volume, the Groveler marks files that are 32 KB or larger in size and identical to one or more files on the volume. It then checks the file in more detail to verify that the content is identical. After the file is verified, it is copied into the \SIS Common Store folder, renamed with a unique GUID, and given the .sis file name extension. The identical files on the volume are then changed to reparse points. When an application tries to open the original file, the file system redirects any file input or output to the *<guid>*.sis file in \SIS Common Store. Figure 24.7 shows an example of two files that were combined with a reparse pointing to the SIS common store.

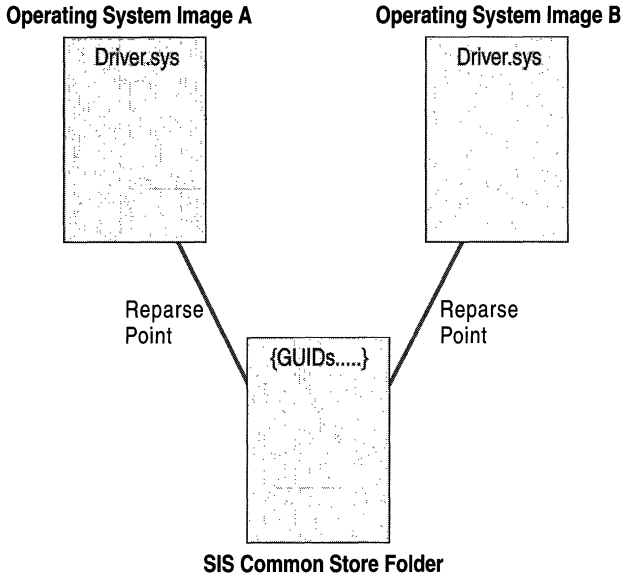


Figure 24.7 A File with a Reparse Pointing to the Common Store

The server has two operating system images, A and B. Both contain the Driver.sys file. The files in both directories are identical, so the data has been placed in the SIS common store and the original files are changed to reparse points with referrals to the <*guid*>.sis file.

Note Even though the files are combined and space is being saved on the disk, for disk quota purposes the users who owned the original files are still charged as if the file had not been combined.

When a file that has been consolidated by SIS is modified or its contents replaced, such as when you copy over the file or modify it in some way, the reparse point is removed and replaced with a copy of the <*guid*>.sis file. The changes are then applied to the fresh copy of the original file. The results of such an operation are shown in Figure 24.8

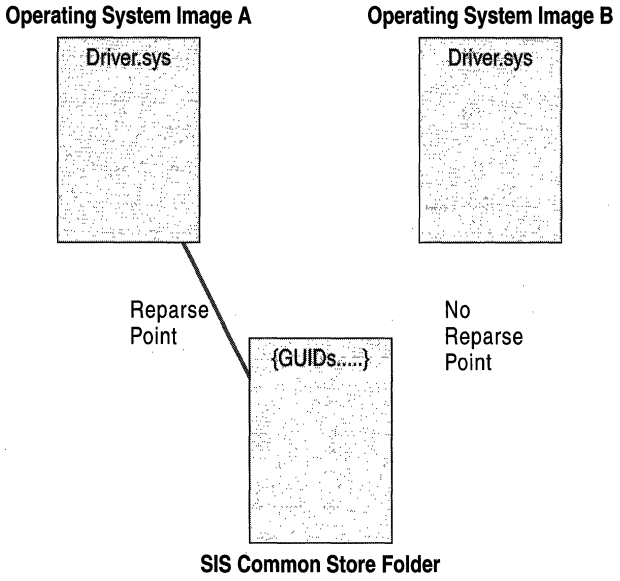


Figure 24.8 A File with No Reparse Point After Modification

The other reparse point or points for the original file are not changed, even if only one reparse point remains pointing to the <*guid*>.sis file. After the final instance of the original file is modified or deleted, the <*guid*>.sys file in the SIS Common Store folder is deleted.

Backing Up a SIS Volume

Because SIS creates reparse points out of files that are identical, your backup program must understand this capability of the NTFS file system. The backup tool that is included with Windows 2000, Ntbackup.exe, handles these files properly.

If you are using a third-party backup solution, verify that it supports reparse points on an NTFS partition. If your backup program is not aware of SIS files, it might make duplicate copies of only the SIS files in the actual backup data file, so that when you restore these backup data files, they create normal files instead of the reparse points.

SIS Groveler Configuration Parameters

Table 24.4 is a list of registry entries for SIS Groveler configuration. SIS is configured by default for optimum performance and minimal system disruption. The HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Groveler\Parameters key is created by default. None of the optional values are created by default. Modify these parameters with caution.

Caution Do not use a registry editor to edit the registry directly unless you have no alternative. The registry editors bypass the standard safeguards provided by administrative tools. These safeguards prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your system. Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. To configure or customize Windows 2000, use the programs in **Control Panel** or the Microsoft Management Console (MMC) whenever possible.

The SIS Groveler configuration entries are located in the following location:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Groveler\Parameters

Table 24.4 SIS Groveler Registry Entries

Parameter	Description
base grovel interval	The base amount of time (in milliseconds) between successive operations when the service has paused.
low-disk-space grovel interval	The maximum amount of time (in milliseconds) between successive operations when the service has paused and disk space is low.

(continued)

Table 24.4 SIS Groveler Registry Entries *(continued)*

Parameter	Description
working grovel interval	The amount of time (in milliseconds) between successive operations during normal operation.
grovel duration	The amount of time (in milliseconds) that the Groveler processes before examining the performance metrics and determining whether to pause the service.
base regrovel interval	The base amount of time (in milliseconds) between successive attempts to process a file that the Groveler cannot access.
max regrovel interval	The maximum amount of time (in milliseconds) between successive attempts to process a file that the Groveler cannot access.
base CPU load threshold	The base value of the CPU load threshold, above which the Groveler pauses.
max CPU load threshold	The maximum value of the CPU load threshold, above which the Groveler pauses.

Optional Registry Parameters for RIS BINL

RIS includes the BINL service, which is responsible for answering clients, validating users and computer objects in Active Directory, and passing information to the client from the RIS server.

The BINL service is configured with default parameters sufficient for optimal performance in most organizations. The HKLM\System\CurrentControlSet\Services\BINLSVC\Parameters key is created by default. ResponseDelay is the only value created by default and is set to zero.

Caution Do not use a registry editor to edit the registry directly unless you have no alternative. The registry editors bypass the standard safeguards provided by administrative tools. These safeguards prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your system. Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. To configure or customize Windows 2000, use the programs in **Control Panel** or the Microsoft Management Console (MMC) whenever possible.

The root registry subkey for the RIS BINL service is located in the following location:

HKEY_LOCAL_MACHINE\SYSTEM\CurrrentControlSet\Services\Binlsvc\Parameters

---

Note RIS does not add these entries to the registry. To add these entries, you must edit the registry or use a program that edits the registry. For more information, see "Using Regedt32.exe and Regedit.exe" in Windows 2000 Server Help.

Table 24.5 contains the optional registry parameters for RIS. All values are of data type REG_DWORD unless otherwise noted.

Table 24.5 Optional Registry Parameters for RIS

Registry Parameter	Specifications (if any)	Description
Debug		Directs RIS to log debugging information from BINL to a log file kept in %windir%\debug\binlsvc.log. To enable this, set the value to 0x80FFFFFF.
LdapOptReferrals	0 is FALSE, 1 is TRUE, default is 0.	Directs BINL to allow referrals when connecting to a non-Global Catalog server. Referrals are always allowed when connecting to a Global Catalog server.
ResponseDelay	Specified in seconds, default is 0.	Specifies how long BINL waits before responding to a client's initial request for a remote boot server.
LdapTimeout	Specified in seconds, default is 30.	The time-out value specified in Lightweight Directory Access Protocol (LDAP) commands. Note that a value of 0 is actually treated as 500 milliseconds.
DefaultServer	Fully qualified domain name (FQDN) of the server.	Specifies the server that RIS uses to initialize the connection to the domain. Normally, the server is not specified and BINL uses any of the available Active Directory servers on the network. Use this entry if you need to force BINL to get its domain information from a specific server.
DefaultGCServer	FQDN of the server.	Specifies the server that RIS uses to initialize a connection to the Active Directory Global Catalog. Normally, the server is not specified and BINL uses any of the available Active Directory servers on the network.
ClientTimeout	Specified in seconds, default is 900.	Determines how long the system waits before displaying time-out errors in the Client Installation Wizard.
UpdateParameterPoll	Specified in seconds. Default is 14,400, which equates to 4 hours.	Determines how often BINL updates all parameters (including registry settings and domain server information) during normal operation.
MaxDSErrorsToLog	Default is 10.	Determines the maximum number of LDAP errors that RIS records in the system log within the error interval. (The error interval is specified by the value of DSErrorInterval.) If more errors are encountered within the interval, they are not individually logged. Instead, a generic error listing the number of errors is logged at the end of the interval. Increasing this value can be helpful in troubleshooting LDAP errors.

(continued)

Table 24.5 Optional Registry Parameters for RIS

Registry Parameter	Specifications (if any)	Description
DSErrorInterval	Specified in minutes, default is 10.	Specifies the error interval. At this time, individual errors are combined into a single event recorded in the system log. The interval size for logging errors is as described by the entry **MaxDSErrorsToLog**.
AssignNewClientsToServer	0 is FALSE, 1 is TRUE, default is 0.	Determines whether new clients are assigned to the server. Setting the value of this entry to 1 (TRUE) causes the server to be set as the client's specified server when creating new computer account objects, or when answering clients whose computer account objects do not contain a specified server. This setting has no effect if the client's computer account object already contains a specified server.

Troubleshooting RIS

The following information will help you troubleshoot problems with RIS. For more information about troubleshooting, see "Troubleshooting Change and Configuration Management" in this book.

Troubleshooting: No Response From a RIS Server

The following sequence occurs with most PXE ROM–enabled computers.

1. When you start the client, you see the PXE boot ROM begin to load and initialize.
2. The computer displays the message "BootP." This message indicates that the computer is broadcasting a request for service to the network.
3. The computer displays the message "DHCP." This message indicates that the computer has received a DHCP response. This might be either the BINL response or the response from the DHCP server.
4. The computer displays the message "BINL." This indicates that it also received a valid BINL response. Sometimes this happens quickly and you do not see it.
5. The computer displays the message "TFTP" followed by a series of dots to indicate progress while it downloads Startrom.com.

6. The computer displays the message "Press F12 for network service boot." This indicates a successful download of Startrom.com. Pressing F12 downloads the Client Installation Wizard, which then requests the Welcome.osc screen from the RIS server.

Note Compaq computer systems provide the ability to press F12 during start up on PC98 or Net PC–based systems. In this case, the user is required to press F12 on the Compaq welcome screen, then press F12 again when prompted by the RIS server. RIS includes a "no F12" version of the first boot image, Startrom.n12. If all of your clients support the F12 key or some capability to request a network service boot through a welcome screen, you can replace the existing Startrom.com in the \Oschooser\i386 directory with the Startrom.n12 boot image. Rename Startrom.com to Startrom.old and then rename the Startrom.n12 boot image to Startrom.com. This allows you to use the F12 key on the client's welcome or "splash" screen to request the Client Installation Wizard and eliminate the need to press F12 a second time.

If the computer does not get past the "BootP" message, it is not receiving any information from the network. Check the following conditions:

- Is the DHCP server available? Has the DHCP service been started on the server?
- Have the DHCP and RIS servers been authorized in Active Directory? RIS and Windows 2000 DHCP servers must be authorized in Active Directory for their services to start.
- Does the DHCP server have a defined IP address scope and is it activated? A DHCP server that does not have any scopes defined cannot service any clients.
- Is there a router between the computer and the DHCP server that is not configured as a DHCP proxy? Verify DHCP packets are being forwarded through your router. Make sure that all routers between the client and the DHCP and RIS servers have been DHCP proxy–enabled. Also make sure that DHCP proxy tables contain both the DHCP and RIS servers. If the RIS server is located on the same network segment as the clients, no modifications are required to the router.
- Are there any error messages in the event log under System Log for DHCP or BINLSVC?
- Can other computers receive an IP address from DHCP on this network segment? Check computers already running an operating system to see if you can successfully renew their IP leases.

If the computer does not get past the "DHCP" message, the client might not be receiving a response from the RIS server. Check the following conditions:

- Are the RIS and Windows 2000 DHCP servers authorized to run on the network? RIS and Windows 2000 DHCP servers must be authorized in Active Directory for their services to start.
- Are other remote boot–enabled clients on the network able to download Startrom.com and see the "Press F12 for network service boot" prompt? If so, this computer might not be supported or is having problems related to PXE ROM. Most systems work with PXE ROM version .99c or later, however certain systems require version .99l. Check the version of the PXE ROM on the computer.
- Is there a router between the computer and the remote installation server that does not allow the DHCP-based requests or responses through? The RIS server communicates through the DHCP packets during the initial service request and response sequence. Blocking these packets does not allow the computer to contact the RIS server.
- Are there any error messages in the event log under the System or Application logs specific to BINLSVC, TFTPD, DNS, or Active Directory services?
- Make sure the **Respond to client computer requesting service** setting is checked in the properties of the RIS server. If this setting is not enabled, the server does not respond to any client requests.
- If the **Do not respond to unknown client computers** setting in the properties of the RIS server is checked, the server only answers clients with computer account objects in Active Directory.
- Run **Verify Server**. This checks to see if all of the necessary services are started and that the settings are correct. Anything it finds wrong it attempts to fix. It might prompt the user for settings that are missing or set incorrectly. It might also prompt the user for the source CD to install files that might have been accidentally erased.
- Check the event log on the RIS server for any errors relating to DHCP, DNS, TFTPD, or BINLSVC.

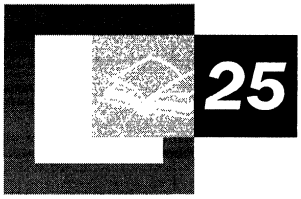
Troubleshooting: Working with Routers

Because Remote OS Installation client service requests are based on the DHCP discovery process, configuring your network to support Remote OS Installation across routers has the same requirements as supporting DHCP across routers. If your environment contains routers between clients and your RIS servers, ensure that DHCP broadcasts from clients (that are used to initiate a remote boot session with RIS servers) are used properly by the routers.

Routers that are configured to forward DHCP broadcasts also forward Remote OS Installation client service requests, however you must ensure that these requests are forwarded to the proper RIS servers in addition to any DHCP servers. Depending on the router models in use, the specific router configuration of DHCP broadcast forwarding might be supported to either a subnet (or router interface), or to a specific host. If your DHCP servers and RIS servers are separate computers, ensure that the routers forwarding DHCP broadcasts are designed so that both the DHCP and RIS servers receive the client broadcasts, otherwise the client does not receive a reply to its remote boot request.

CHAPTER 25

Troubleshooting Change and Configuration Management



When using Change and Configuration Management, you might come across some problems that need troubleshooting. You might encounter Group Policy issues where all or part of the expected Group Policy settings are not applied or did not have the expected effect. You might encounter User Data Management issues such as problems with offline files or redirection of folders to network locations. You could have Software Installation and Maintenance issues such as problems with assignment, publication, or upgrading of applications. Or you might have User Settings Management issues, such as difficulties with roaming user profiles. You might also encounter Remote OS Installation issues where installation of Windows 2000 to a remote computer over the network is not functioning. All of these different types of problems are discussed in this chapter along with possible causes and the troubleshooting steps you need to follow to solve them. It is recommended that you familiarize yourself with the diagnostic tools provided in the *Microsoft® Windows® 2000 Resource Kits*. Some of the recommended tools for troubleshooting Change and Configuration Management problems are Gpotool.exe, Gpresult.exe, Netdiag.exe, Replmon.exe, and other tools.

In This Chapter

Related Information In the Resource Kit

- For information about troubleshooting Active Directory, See "Active Directory Diagnostics, Troubleshooting, and Recovery" in this book.

Best Practices

There are some best practices you should use to avoid the need for troubleshooting.

Examine Active Directory and Group Policy To predict how the Windows 2000 Change and Configuration Management features affect the desktops they manage, you need to examine your organization's Active Directory structure, review how Group Policy works, and understand precedence among the Group Policy objects that are in effect. For more information, see "Group Policy" and "Active Directory Logical Structure" in this book. For information about troubleshooting Active Directory, See "Active Directory Diagnostics, Troubleshooting, and Recovery" in this book.

Check the Configuration of the Network Before you can use the Change and Configuration Management features, the domain controllers, Distributed Host Configuration Protocol (DHCP), and Domain Name System (DNS) servers and services must be correctly configured and operating properly. Fix any of these problems first.

Use Test Labs If you work in a large organization, use test labs to determine the effect of enabling or disabling various change and configuration management features. You are using tools that swiftly affect their entire scope—which might be your entire organization—so you will need to make sure your proposed course of action works as you expect in a test lab.

Change Only One Item at a Time When you do this, you are assured that the change in behavior is due to a known action on your part.

Don't Ignore Error Messages Write error messages down if you suspect they might disappear if the system fails. Pay attention if they change slightly.

Examine Event Logs Certain Windows components are dependent on other components, and this is reflected in the chronological order of failures recorded in event logs. This is useful information. For specific instructions on how to enable verbose logging of certain events, see "Verbose Logging" later in this chapter.

Roaming User Profiles Because roaming user profiles keep a copy of the profile local and only resynchronize this profile during logoff, some care must be taken to prevent awkward behavior.

After users have their roaming user profile enabled, they should first log on from their main workstation and then log off to have a correct roaming user profile stored. For daily usage of multiple computers, users should always log on to their main computer first, and then log on to other computers that they use.

When logging off, log off of the computers in reverse order—that is, log off of the computer you logged on to last first and work backward. For example, if the user first logs on to Computer 1 and then logs on to Computer 2, he or she should log off of Computer 2 first, and then log off of Computer 1.

Troubleshooting Tools

There are many tools you can use to troubleshoot Change and Configuration Management issues. Some of these tools are recommended in the troubleshooting issues later in this chapter. Before you use these tools, it is best to read the Help files on the *Windows 2000 Server Resource Kit* companion CD or the Windows 2000 CD to familiarize yourself with the procedures for using each tool. The following tools are recommended in this chapter.

Tools from the Windows 2000 Server Resource Kit CD

- Gpotool.exe
- Gpresult.exe
- Replmon.exe
- Addiag.exe

For more information about these tools, see Tools Help on the *Windows 2000 Server Resource Kit* companion CD.

Tools from the Windows 2000 Operating System CD

- Netdiag.exe
- Replmon.exe

For more information about these tools, see Windows 2000 Support Tools Help. For information about installing and using the Windows 2000 Support Tools and Support Tools Help, see the file's readme.doc in the Support\Tools folder of the Windows 2000 operating system CD.

Verbose Logging

Verbose logging tracks all changes and settings applied using Group Policy and its extension to the local computer and to users who log on to the computer. The log file is located at *SystemDrive*\Debug. This folder is a hidden folder. Enabling verbose logging involves adding the registry key for verbose logging. By performing the following steps, you can enable verbose logging for Group Policy and its extensions.

▶ **To enable verbose logging (Userenv.log)**

1. Log on as the local administrator.
2. Click **Start**, and then click **Run**.
3. In the **Open** text box, type **regedit**, and then click **OK**.
4. Open the HKEY_LOCAL_MACHINE/Software/Microsoft/Windows NT /Current Version/Winlogon key, and then click **Edit** on the toolbar.
5. Click **Add Value**, and enter the value name **UserenvDebugLevel**.
6. Set the **Data Type** to REG_DWORD, click **OK**, and then under **Value**, enter the number **30002**. Set the option button to **Hex**.
7. Log off and then log on again.

Note 30002 is for verbose logging, 30001 is for errors and warnings only, and 30000 logs nothing. The log will be found at %windir%\debug.

▶ **To enable verbose logging using a different method**

1. Log on as local administrator.
2. Copy the file Envdebug.ini to the root of the system drive (C:\Temp)
3. Click **Run**, type **Regini C:\envdebug.ini**, and then click **OK**.

   You must refer to Regini.exe in the path.

You can remove the registry key for verbose logging by taking the following steps.

▶ **To disable verbose logging**

1. Log on as the local administrator.
2. Click **Start**, click **Run**, and in the **Open** text box, type **regedit.exe** and then click **OK**.
3. Navigate to the following registry key:

   HKEY_LOCAL_MACHINE/Software/Microsoft/Windows NT/
   Current Version/Winlogon
4. Click the value **UserenvDebugLevel** in the right pane of the Regedit window.

5. On the Regedit toolbar, click **Edit**.
6. On the **Edit** drop-down menu, click **Delete**.
7. Click **Yes**.

Software Installation and Maintenance Logging

Follow the steps mentioned earlier to use Regedit to add the following key:

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\
Current Version\Diagnostics\Appmgmtdebuglevel

REG_DWORD = F

Windows Installer Logging

Follow the steps listed earlier to use Regedit to add the following key:

HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\
Installer\Logging

REG_SZ = vicewarmup

HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\
Installer\Debug

REG_DWORD = 3

Caution Do not use a registry editor to edit the registry directly unless you have no alternative. The registry editors bypass the standard safeguards provided by administrative tools. These safeguards prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your system. Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. To configure or customize Windows 2000, use the programs in Control Panel or Microsoft Management Console (MMC) whenever possible.

Group Policy Issues

In these troubleshooting cases, all or part of the expected Group Policy settings are not applied, or do not have the expected effect.

Scripts Do Not Run

You configure scripts (Log on, Log off, Startup, and Shutdown) in a Group Policy object, and the user who is managed by the Group Policy object logs on and later logs off, or starts and then shuts down his or her computer. One or more scripts that were configured to run are not run.

Possible Causes:

- Group Policy is not applied.
- Active Directory replication has not yet completed on the domain controller in use.
- File Replication Service (FRS) replication has not yet completed on the domain controller in use.

Diagnostic Tests:

Confirm that the Group Policy Scripts extension is applied to the user's computer. Run Gpresult.exe in verbose mode, and examine the output under the heading **User received "Scripts" settings from these GPOs**.

Check to see if the user can successfully connect to a domain controller.

Run Gpotool.exe and confirm that the user can successfully connect to a domain controller on the network. If the result of this test does not show a domain controller, discontinue troubleshooting Group Policy, and investigate network connectivity to your domain controller. You can use Netdiag.exe to investigate your network connectivity further. If you can connect to a domain controller, check to see if other computers in the same organizational unit are successfully receiving the Group Policy Scripts extension Group Policy settings. If other computers are successfully receiving these Group Policy settings, it is probably a replication issue.

Run Gpotool.exe in verbose mode to verify that the Group Policy object that was not applied has been successfully replicated to all domain controllers.

It might be that the Group Policy Scripts extension Group Policy settings were applied, but the actual scripts did not run. The user might have received the Group Policy successfully, but failed to receive the actual script due to a synchronization problem with the Sysvol on the domain controller in use.

Run Gpotool.exe in verbose mode to find out which domain controllers have an unsynchronized Sysvol. After you have identified these domain controllers, copy a text file to the Sysvol of one of the domain controllers in question. Confirm that this text file is successfully replicated to the other domain controllers. If it has replicated, increase the version number of Group Policy. Then, at the next logon, the Group Policy is re-applied and the user receives the script. If the text file does not replicate, check the network connectivity between the servers.

Registry Settings Using Administrative Templates Are Not Applied

You configure Administrative Templates to apply registry-based Group Policy settings in a Group Policy object. The user who is managed by the Group Policy object logs on, and the administrative templates are not applied.

Possible Causes:

- Group Policy is not applied.
- Active Directory replication has not yet completed on the domain controller in use. Therefore the .pol file is not present and cannot be applied.

Diagnostic Tests:

To confirm that Administrative Template Group Policy settings are not applied to the user's computer, run Gpresult.exe in verbose mode on the client. Look for the following text in the output:

```
The user received "Registry" settings from these GPOs:
```

If this section does not appear, no Administrative Template Group Policy settings are applied.

If some Administrative Template Group Policy settings are applied, confirm that the specific ones of interest are applied to the client computer and user. Look at the listing of the Group Policy objects that are applied, and also the registry settings that these Group Policy objects contain. Look for the following text in the output:

```
The user received "Registry" settings from these Group Policy objects:
    Local Group Policy
        Revision Number:    40
        Unique Name:    Local Group Policy
        Domain Name:
        Source:        Local computer
```

```
    EU-AppDeploymentPolicy
        Revision Number:    528
        Unique Name:    {ESD6ECF2-A8C9-11D2-B575-0008C7457B4E}
        Domain Name:    ntdev.microsoft.com
        Source:         Domain

    EU-AppDeploymentTest
        Revision Number:    11
        Unique Name:    {DFSDAC2B-FA94-11D2-9332-00C04FA3787A}
        Domain Name:    ntdev.microsoft.com
        Source:         Domain

The following registry settings were applied:

    KeyName:
Software\Microsoft\Windows\CurrentVersion\Policies\Comdlg32
    ValueName:    NoBackButton
    ValueType:    REG_DWORD
    Value:    0x00000001

    KeyName:
Software\Microsoft\Windows\CurrentVersion\Policies\Comdlg32
    ValueName:    NoFileMru
    ValueType:    REG_DWORD
    Value:    0x00000001
```

If the Group Policy objects that you think should apply are not applied, then check the other possible causes that could lead to Group Policy objects not applying successfully. Confirm your Group Policy object configuration in Active Directory.

Use Netdiag.exe to check network configuration and that DNS is working and configured.

Check to see that the user can successfully connect to a domain controller. Without a Windows 2000 domain controller, Group Policy cannot be applied.

Run Gpotool.exe in verbose mode on the client and check to see if any domain controllers are available.

Confirm that domain controller replication (Active Directory and Sysvol) is successful between all domain controllers. You can check replication status using Gpotool.exe. If domain controller replication is working, check the Sysvol file share on the user's log on domain controller to confirm that the .pol file is present.

If you can connect to the domain controller:

- Check to see that other computers in the same organizational unit are successfully receiving the Administrative Template Group Policy settings. Then verify user or computer group membership and check for filtering.
- If no other computers are receiving these Group Policy settings, then there might be a configuration issue. Confirm your Group Policy object configuration and design.
- Check to see if loopback processing is enabled.

For more diagnostic detail, you can enable Group Policy verbose logging.

Group Policy Object Does Not Open

You attempt to either open or edit a Group Policy object, and receive an error indicating that the Group Policy object cannot be accessed or opened.

Possible Causes:

- There is no write access to the Group Policy object.
- The domain controller you are focused on is down.
- You have attempted to edit a Group Policy object in another domain and the trust relationship broken.
- Network connectivity or dropped packets is causing intermittent failures.

Diagnostic Tests:

▶ **To check that you have Read and Write access to the Group Policy object**

1. Select the Group Policy object to open or edit on the Group Policy property page of the site, domain, or organizational unit.
2. Right-click the Group Policy object, and then click **Properties**.
3. Click the **Security** tab.
4. Verify that you have both Read and Write access. You must have both; you cannot open a Group Policy object in read-only mode.

▶ **To confirm that you can successfully connect to the domain controller that you are focused on**

1. Run Netdiag.exe to check network connectivity and to confirm that DNS is configured and working.
2. Locate the name of the domain controller that you are focused on. At the command prompt, type **SET**. Look for the variable *Logonserver* in the output and make a note of the name of the domain controller that it points to. (LOGONSERVER=\\NTDSDCB)
3. Run Gpotool.exe in verbose mode and confirm that the domain controller that you are focused on is available and online.

If you are editing a Group Policy object in another domain, confirm that the trust relationship is not broken.

Check for group membership that gives the user permissions to edit the Group Policy object in the remote domain.

Check for media sense issues, that is, problems with Windows not detecting peripherals or the network.. If you are logged on locally to the domain controller that contains the Group Policy object you want to edit, even if the domain controller is not replicated, confirm that the following conditions are met.

- There is a network cable plugged into the network adapter on the domain controller.
- This network cable is connected to a powered network hub.

Active Directory and Sysvol Are Unsynchronized

You configure a logon script with a Group Policy object. In a multiple domain controller environment, this change requires that Active Directory and the Sysvol replicate this change to all the domain controllers. Before both Active Directory and Sysvol are fully replicated, a user logs on to the system and is authenticated by a domain controller that is not fully replicated, and the user experiences unexpected behavior.

Possible Causes:

- In a multiple domain controller environment, changes to Active Directory have not yet completed replication.
- In a multiple domain controller environment, changes to the Sysvol have not yet completed replication.

Diagnostic Tests:

Run Netdiag.exe to check client network configuration and that DNS is configured and working correctly.

If the user has a roaming user profile, verify that he or she correctly receives the roaming user profile at logon.

Run Gpresult.exe to see if any Group Policy Settings are applied. If no Group Policy settings are applied, see "No Group Policy Objects Are Applied" later in this chapter.

▶ **To check the status of Active Directory and Sysvol replication on each server**

1. Run Gpotool.exe to check the number of unique Group Policy objects available on the network, and the status of each of these Group Policy objects on each domain controller. The status output from Gpotool.exe indicates all necessary information to diagnose if Active Directory and Sysvol are synchronized for each domain controller that you can connect to.
2. If you find that Sysvol is not synchronized between two domain controllers, place any text file on the Sysvol of one of the domain controllers. Confirm that it is replicated to the other domain controllers. If this fails, check the network connectivity between the domain controllers.
3. If Active Directory is not synchronized between domain controllers, run Active Directory Replication Monitor (Replmon.exe), which can provide additional information about the state of Active Directory synchronization, and provide assistance in resolving the problem.

More Than 1,000 Group Policy Objects are present and Group Policy fails

More than 1000 Group Policy objects are stored in a domain, and Group Policy fails. This failure can affect any extension of Group Policy

Possible Causes:

The API that gathers the list of readable Group Policy objects can only return 1000 Group Policy objects at one time. If a user or computer has Read access to more than 1000 Group Policy objects, Group Policy fails. This failure occurs regardless of whether Apply Group Policy is enabled on the readable Group Policy objects. If both Read and Apply Group Policy are enabled on dozens or hundreds of Group Policy objects, Group Policy processing is very slow.

Note If you make sure no user or computer has Read access to that many Group Policy objects, the problem does not occur. It is recommended that you always turn off (sometimes explicitly Deny, but in most cases don't Allow) Read access whenever Apply Group Policy is turned off.

Diagnostic Tests:

This failure disables Group Policy from applying to the intended recipients until it is resolved.

The event log should contain entries similar to the following:

```
Event Type:    Error
Event Source:    Userenv
Event Category:    None
Event ID:    1000
Date:        10/13/1999
Time:        2:17:33 PM
User:        JEFFSMITH
Computer:    RESKIT1
Description:
Windows cannot search for Group Policy objects with (3).
```

Event Type: Error
Event Source: Userenv
Event Category: None
Event ID: 1000
Date: 10/13/1999
Time: 2:19:37 PM
User: JEFFSMITH
Computer: RESKIT1
Description:
Windows cannot search for Group Policy objects with (4).

When you observe the %systemroot%\debug\usermode\userenv.log you should find entries similar to the following:

```
USERENV(ac.2d0) 14:17:33:272 EvaluteDeferredGPOs:  Failed to search with
error 0x3
USERENV(ac.2d0) 14:17:33:602 GetGPOInfo:  EvaluateDeferredGPOs failed.
Exiting
USERENV(ac.2d0) 14:17:33:632 ProcessGPOs: GetGPOInfo failed.
USERENV(ac.13c) 14:19:37:143 EvaluteDeferredGPOs:  Failed to search with
error 0x4
USERENV(ac.13c) 14:19:37:313 GetGPOInfo:  EvaluateDeferredGPOs failed.
Exiting
USERENV(ac.13c) 14:19:37:343 ProcessGPOs: GetGPOInfo failed.
```

No Group Policy Objects Are Applied

A user managed by Group Policy logs on, but does not receive Group Policy. Other computers and users are successfully receiving Group Policy.

Possible Causes:

- The user is using a non-Windows 2000 system.
- Migration problems occur because the user logs on to a Windows NT 4.0 domain.
- The network is down.
- DNS does not work or is not configured.
- There is no available domain controller.
- The available domain controller has replication problems.
- Security group membership causes Group Policy objects to be ignored. This amounts to inadvertent use of the Group Policy scope-filtering mechanism.
- Loopback is in effect, so that user Group Policy is replaced by computer Group Policy.
- Disk quotas are enforced on a user, and there is not enough disk space allocated for the user to carry out the application of Group Policy or some extension of it. In particular, Software Installation not only places the application in the user's disk quota, but it also creates temporary files that require some of the user's quota; so the quota must be large enough to allow for the user to carry out the application of Group Policy.

Diagnostic Tests:

Confirm that the client is running Windows 2000. Group Policy does not apply to Windows NT 4.0 and Windows 95 and Windows 98 computers even if they are Active Directory clients.

Verify that the user and computer accounts are managed by Windows 2000 domain controllers or by Windows NT 4.0 domain controllers.

For Group Policy settings under the User Configuration node to be applied, the user account object must be in Active Directory—that is, it must be handled by a Windows 2000 domain controller, not a Windows NT 4.0 domain controller.

For Group Policy settings under the Computer Configuration node to be applied, the computer account object must be in Active Directory, that is, it must be handled by a Windows 2000 domain controller, not a Windows NT 4.0 domain controller.

If the user and computer accounts are both handled by a Windows 2000 domain controller, then the Windows 2000 client computer receives Group Policy computer settings at startup, and the user receives Group Policy settings at logon.

During migration from Windows NT 4.0 to Windows 2000, Windows 2000 domain controllers and Windows NT 4.0 domain controllers can coexist. For details on the Group Policy behavior of such a mixed environment, see "Group Policy" in this book.

Check client network connectivity and confirm that DNS is working and configured. Netdiag.exe can assist in determining this.

Run Gpresult.exe to confirm that no Group Policy objects were applied to the client computer.

Check to see if the user is subject to excessively rigid disk quotas.

▶ **To check that the client computer has access to an available domain controller using Gpotool.exe**

1. Confirm (by running Gpotool.exe) that the Group Policy objects that you want to apply actually exist on your domain controllers.
2. If running in a multiple Group Policy object environment, confirm that all Active Directory and Sysvol information has replicated successfully on the domain controllers.

▶ **To confirm that Group Membership does not cause the Group Policy objects to not apply because of a filter**

1. Run Gpresult.exe to find security group membership for the user who you are troubleshooting.
2. Check the property sheet of the Group Policy object that did not apply, to confirm that the correct security groups are configured to have Group Policy apply. The user must have Apply Group Policy and Read permissions for the Group Policy object on the basis of his or her membership in Windows 2000 security groups.
3. Verify that loopback is enabled.

Enable verbose Userenv.log logging to check for any other errors.

Inheritance Issues with Group Policy Objects Cause Unexpected Results

A user managed by Group Policy logs on and the Group Policy applied is different from what you expect.

Possible Causes:

- The order in which Group Policy objects are applied is not what you expect. Policy applied later usually takes precedence.
- There are Group Policy object **No Override** issues.
- There are **Block Inheritance** issues.
- The entire Group Policy object is disabled.
- The user or computer parts of the Group Policy object are disabled.

Diagnostic Tests:

Write down a list of the Group Policy settings that you expect to be applied to your client user and computer. Using Gpresult.exe, generate the list of the Group Policy settings that actually applied to the client user and computer. Compare the two lists to find out which Group Policy settings are not applied.

Starting at the top of the directory tree, search down to locate the client user object. As you search down the tree, make a note of all organizational units and any associated Group Policy objects applied to each of these organizational units. Repeat this procedure for the client computer object.

Group Policy objects are applied in the following order: local, site, domain, organizational unit. If there is more then one Group Policy object that is associated to a specific site, domain, or organizational unit, these are applied in the specific order that you set.

To check the order in which multiple Group Policy objects are applied, right-click the site, domain, or organizational unit in question, and then select **Properties**. Then click the **Group Policy** tab.

Look for Group Policy objects that have special settings applied to them. Right-click the site, domain, or organizational unit that contains the Group Policy objects that you are investigating. Then click the **Group Policy** tab.

For each Group Policy object, check whether the **No Override** or **Block Inheritance** options are enabled. If these options are enabled on any Group Policy object, you should start from the top of the tree and work down to locate any **No Override** issues. Then start from the bottom of the tree and work upward to locate any **Block Inheritance** configuration issues.

Note You should use the **No Override** and **Block Inheritance** features sparingly because they make troubleshooting Group Policy more difficult.

To confirm that the Group Policy object that you want to apply is not disabled, look at the Group Policy tab, and click **Options**.

To confirm that the Computer Configuration or User Configuration portion of the Group Policy object is not disabled, click the **Properties** button on the **Group Policy** tab, and check the **General** tab.

Only Some IP Security and User Rights Policy Settings Are Applied

You configure IP Security and Users Rights Assignment Group Policy settings in various Group Policy objects, and when a user managed by some or all of these Group Policy objects logs on, only some of the expected IP Security and Users Rights Assignment Group Policy settings are applied.

Possible Causes:

Some Group Policy extensions have different application rules. These extensions do not merge Group Policy settings from various Group Policy objects, but always use the last written setting.

Diagnostic Tests:

In most situations, if a Group Policy extension has been configured in more than one Group Policy object for a user or computer, the configuration settings are merged from all the Group Policy objects that contain settings for that Group Policy extension.

For example, if you apply two Group Policy objects to a user, and the first Group Policy object enables Software Deployment and assigns an application, and the second Group Policy object assigns another application, the user receives both applications.

IP Security and Users Rights Assignment are Group Policy extensions that are exceptions to this rule. They always use the last written setting. Therefore, when troubleshooting the resulting set of Group Policy, identify Group Policy objects that contain settings within these two extensions. You can check to see if either of these extensions is enabled or configured on your client by running Gpresult.exe in verbose mode.

Security Settings on Group Policy Object Cause Unexpected Results

You configure security on various Group Policy objects in Active Directory by adjusting group membership and changing security descriptors (access control lists, or ACLs) on the Group Policy objects. When a user managed by these Group Policy objects logs on, the Group Policy applied to this user is different from what you expect.

Possible Causes:

- Problems with the security groups used to filter whom the Group Policy object applies to.
- Problems with the ACLs set on Group Policy objects.

Diagnostic Tests:

Write down a list of the Group Policy settings that you expect to apply to the client user and computer. Using Gpresult.exe, generate the list of the Group Policy settings that actually applied to the client user and computer. Compare the lists to determine which Group Policy objects are not applied.

▶ **To check security filtering and ACLs for Group Policy objects in question**

1. Right-click a site, domain, or organizational unit in which the Group Policy object is referenced.
2. Click **Properties** on the shortcut menu.
3. Click the **Group Policy** tab and select the Group Policy object that you want to investigate.
4. Click the **Properties** button, and then click the **Security** tab.

For this Group Policy object to apply successfully, at least one security group to which the user belongs must be listed with the Read or Apply Group Policy permissions set to **Allow**. If either of these two permissions is not selected, the Group Policy object will not apply.

Note Access control lists (ACLs) on Group Policy objects should only be configured and diagnosed using the user interface as explained earlier. Do not try to manually configure ACLs on Group Policy objects located in the Sysvol. This leads to Group Policy objects not applying, or to other unexpected behavior.

User Data Management Issues

You might encounter User Data Management problems when using offline files, or when redirecting folders to network locations.

Files Do Not Synchronize

A user cannot synchronize certain files or folders.

Possible Causes:

- Files with the extensions .mdb, .ldb, .mdw, .mde, .db are not synchronized by default.
- Network connection problems accessing the files to be synchronized.
- Insufficient disk space on the client computer to synchronize files.
- Insufficient user rights to read or write the files to be synchronized.
- You have configured a Group Policy setting to specify additional file name extensions that cannot be synchronized.

Diagnostic Tests:

Check the extensions of the files that were not synchronized to confirm that they are not on the list of files to exclude.

Check user rights on the unsynchronized files.

▶ **To verify that the user can connect to the file share containing the files you want to synchronize**

1. Ping the computer using the computer's IP address to test base-level IP connectivity.
2. Ping the computer using the server name to test Domain Name System (DNS) name resolution.
3. Use the **net view** command to view the computer using the server name to see the available shares. This also verifies that the user has rights to access the shares.

Check the amount of free disk space on the client to make sure there is sufficient disk space to synchronize the missing files.

Check to see if you have applied any Group Policy settings that restrict other extensions from being synchronized.

You should be sure that you have enabled the following Group Policy setting:

- Computer Configuration\Administrative Templates\Network\Offline Files\Files not cached

Using this Group Policy setting, you can designate additional file name extensions that cannot be synchronized.

You can also check this on your client by running the Gpresult.exe tool and looking for the following in the output:

```
KeyName:    Software\Policies\Microsoft\Windows\NetCache
ValueName:    ExcludeExtensions
ValueType:    REG_SZ
Value:        *.xls
```

Any file name extensions listed in the **Value** line of the sample output listed earlier will not be synchronized. In this sample output, any files with the extension .xls will not be synchronized. The user cannot override this Group Policy setting.

User Cannot Make Files and Folders Available Offline

A user would like to make certain files or folders available for offline use. The user right-clicks on a file or folder, but **Make Available Offline** does not appear on the shortcut menu.

Possible Causes:

- Terminal Services is enabled. Offline Files and Terminal Services are mutually exclusive.
- The file or folder selected is actually a local file or folder and not on a network file share.
- Offline Files is not enabled.
- A Group Policy setting was applied to disable Offline Files.
- The user is trying to make his or her redirected My Documents folder available offline but does not have access to the file share.

Diagnostic Tests:

Run Gpresult.exe to check if Terminal Services is enabled.

Validate that the file or folder is on a network file share and not a local share.

▶ To verify that Offline Files is enabled

1. Click **My Computer**.
2. Click **Tools**, and then select **Folder Options**.
3. Click the **Offline File**s tab.
4. Select the check box beside **Enable Offline Files**.

If this procedure does not enable Offline Files, there might be a Group Policy setting that prevents Offline Files from being enabled. The Group Policy setting that controls this is:

Computer Configuration\Administrative Templates\Network\Offline Files\Enable

To see if this Group Policy setting is applied, run Gpresult.exe in verbose mode on the client computer. Look at the output of this tool and compare its output to the following sample:

```
KeyName:    Software\Policies\Microsoft\Windows\NetCache
    ValueName:    **del.Enabled
    ValueType:    REG_SZ
    Value:
```

If the output of Gpresult.exe on your client contains this text, this Group Policy setting is applied and you have disabled Offline Files. In this case, it is not possible to enable Offline Files until this Group Policy setting is changed.

If files and folders have the **Make Available Offline** shortcut menu option, but a redirected **My Documents** folder does not, then you should check that the My Documents folder is actually redirected successfully and is not local. Then verify that the user has appropriate file security to read and write to the location where the **My Documents** folder is redirected.

Note: When you have configured Group Policy **Enable Offline Files** with a setting of **Disable**, the Offline Files feature is disabled. Although the user can turn on and turn off the **Enable Offline Files** check box, as explained earlier, the Group Policy setting takes priority.

Files Available When Online Are Not Available When Offline

Documents and programs that are accessible when connected to the network are not synchronized with the local cache for offline use.

Possible Causes:

- The files reside on a computer that is not running Windows 2000 Professional. Computers running previous versions of Windows do not support automatic caching of files and folders.
- Offline files are not enabled on the client computer.
- **Allow caching of files in this shared folder** is not enabled on the file share where the documents are being accessed.
- **Allow caching of files in this shared folder** is enabled but is not set to Automatic Caching.

Diagnostic Tests:

Check to see if the server containing the file share is running Windows 2000 Server. You need Windows 2000 Server and Windows 2000 Professional to use this feature.

▶ **To see if Offline Files is enabled on the client computer**

- Navigate to a network file share, right-click a file or folder and check to see if there is a **Make Available Offline** shortcut menu.

▶ **To check the configuration of the file share**

1. On the file server containing the file share, click **My Computer.**
2. Navigate to the folder that is shared, right-click the folder, and then select **Properties**.
3. Click the **Sharing** tab, and then click **Caching**.
4. Make sure the check box for **Allow caching of files in this shared folder** is selected.
5. In the drop-down box, select one of the following:
   - **Automatic Caching for Documents** if this share contains documents.
   - **Automatic Caching for Programs** if this share contains application files.

My Documents Icon Is Missing

The user attempts to use the My Documents folder and the **My Documents** folder is not available on the desktop or in Windows Explorer.

Possible Causes:

- The My Documents icon has been deleted.
- There is a Group Policy setting that disables the My Documents icon.

Diagnostic Tests:

If the My Documents icon is deleted or disabled by the user, it can be restored by using the following procedure:

▶ **To restore the My Documents icon**

1. Open **My Computer**.
2. Click **Tools,** click **Folder Options**, and then click the **View** tab.
3. Click the **Show My Documents on the Desktop** check box to select it.
4. Log off and then log back on.

You can apply a Group Policy setting that restricts access to the My Documents folder. The name of the Group Policy setting that controls this is:

User Settings\Administrative Templates\Start Menu and Taskbar\Hide My Documents icon on Desktop

To see if this Group Policy setting is applied, run Gpresult.exe in verbose mode on the client computer. Look at the output of this tool to see if the following registry setting is applied:

```
KeyName:    Software\Microsoft\Windows\CurrentVersion\Policies\NonEnum
ValueName:    {450D8FBA-AD25-11D0-98A8-0800361B1103}
ValueType:    REG_DWORD
Value:        0x00000001
```

If this is present, the **Hide My Documents folder on the Desktop** Group Policy setting is enabled.

Note When you apply the **Hide My Documents icon on the Desktop** Group Policy setting to a user, it is impossible for a user to override it. Although the user can turn the My Documents icon on and off as shown in the procedure listed earlier, the Group Policy setting still takes priority.

Folders Are Not Redirected

You configure a user managed by a Group Policy object to have his or her My Documents folder redirected to the server share *ServerName*\MyDocs*Username*. When the user logs on to the network, his or her My Documents folder is not redirected to this server.

Using Group Policy you can redirect the My Documents, Application Data, Desktop, and Start menu folders from a user's profile to a server-based Universal Naming Convention (UNC) location.

Possible Causes:

- The client computer is running Windows NT 4.0, Windows 95, or Windows 98.
- Group Policy is not applied.
- The network share is unavailable, and Offline Files is not enabled.
- The user does not have appropriate access rights to the share on which he or she wants to redirect the folder.
- There is a disk quota on the target folder.
- You use a mapped drive for the target path rather than a UNC path.

Diagnostic Tests:

Confirm that the client computer is running Windows 2000 Professional. Group Policy does not work on earlier versions of the Windows operating system.

Check to see if any Group Policy settings are applied that include Folder Redirection configuration information.

Run Gpresult.exe in verbose mode. You should see a section in the output that starts with the heading:

```
The user received "Folder Redirection" settings from these Group Policy
objects:
```

If this is not present, check to see why no Folder Redirection Group Policy is applied.

Check to see that the correct Group Policy objects containing Folder Redirection configuration information are applied and that the expected folders are redirected.

Run Gpresult.exe in verbose mode. Refer to the Gpresult.exe documentation to analyze which Group Policy objects contain Folder Redirection information, and their locations.

The following is sample output that contains this type of information:

```
The user received "Folder Redirection" settings from these Group Policy
objects:

    EU-RedirectedDesktop-Marketing
        Revision Number:    16
        Unique Name:    {C19SADC-A8E8-11D2-9BEB-00A024070A22}
        Domain Name:    ntdev.microsoft.com
        Source:        Domain

    EU-FolderRedirection-Building26
        Revision Number:    11
        Unique Name:    {FBEE2508-BCAA-11D2-B3EE-00C04FA3787A}
        Domain Name:    ntdev.microsoft.com
        Source:        Domain

    Desktop is redirected to \\policy1\desktop\%username%
    My Documents is redirected to \\policy1\mydocs1\%username%
    My Pictures is redirected to \\policy1\mydocs1\%username%\My
Pictures
```

▶ **To check for network connectivity with the computer holding the redirected folder**

1. Ping this computer with the computer's IP address to test base-level IP connectivity.
2. Ping this computer with the server name to test Domain Name System (DNS) name resolution.

If the server that contains redirected folders is offline, and Offline Files is disabled, users cannot gain access to their data. To enable Offline Files, see "User Cannot Make Files And Folders Available Offline" earlier in this chapter.

If the server that contains the redirected folders is offline, and Offline Files is enabled, then users should have access to their data as long as those files are accessed when the users were previously online. If these files and folders are not available, see "Files Do Not Synchronize" earlier in this chapter.

Verify that the user has enough file security to access folders to which his or her data is redirected. You should assign a user Full Control security access on the access control lists (ACLs) of the root of the share where he or she will redirect data.

Check to see if there is a disk quota enabled on the volume of the computer that contains the redirected folder. If there is a quota enabled, make sure that this quota is not exceeded. If it is exceeded, increase the quota or have the user delete some files.

Redirection Is Successful But Files and Folders Are Unavailable

You configure a user managed by a Group Policy object to have his or her My Documents folder redirected to the server share *ServerName*\MyDocs*Username*. And when the user logs on, the folders are successfully redirected, but are not available to the user on this redirected share.

Possible Causes:

- Network connectivity problems.
- The share is not available and items are not available in the local cache.
- When using applications, open and save operations have hard coded locations and do not use the redirected path.

Diagnostic Tests:

Ping the server that stores the redirected folder to test the network connectivity.

Ping *\\ServerName\Fileshare*

Check user rights on the redirected folder to see if the user has the correct NTFS security file access permissions. The user should have Read and Write access if he or she is saving and retrieving documents.

Check the list of applications that the user is using. Older applications might not be able to recognize the redirected folders.

Software Installation and Maintenance Issues

Assigning, publishing, or upgrading applications doesn't work.

Published Application Does Not Appear

You publish applications in a Group Policy object, and the user who is managed by the Group Policy object logs on. He or she clicks **Add/Remove Programs** in Control Panel, but no applications appear in the **Add New Programs** list box.

Possible Causes:

- The user is running Terminal Services on the desktop.
- The Group Policy object does not apply to this user. For example, the user might be filtered out of the Group Policy object by a security group.
- The user has not logged on since the Group Policy object was created.
- Group Policy did not run.
- The Appmgmt extension does not run.
- The user cannot access Active Directory.
- The user cannot access the software distribution point.

Diagnostic Tests:

Use Addiag.exe to see if Terminal Services is running.

Open the Group Policy object with the Group Policy snap-in, and make sure the Group Policy object manages the user.

Have the user log off and then log back on. Verify that the user does not log on with cached credentials—that is, that the user is authenticated by a domain controller.

Run Gpresult.exe to verify that Group Policy runs.

Run Addiag.exe to verify that the Appmgmt extension runs.

Check to see that the user can access Active Directory. If they are validated by a Windows 2000 domain controller, they can access Active Directory.

▶ **To test basic connectivity and permissions**

1. Ping the domain controller by IP address to test base-level connectivity.
2. Ping the domain controller by server name to test the DNS server.
3. Use the Net view command to view *ServerName*\Sysvol to see if the user has rights to read the Sysvol.
4. Check to see if the user can access the software distribution point.
5. Ping the software distribution point by IP address to test base-level connectivity.
6. Ping the software distribution point by server name to test the DNS server.
7. Use the **net view** command to view *ServerName**AppShares* to see if the user has rights to read the software distribution point.

Published Application Does Not Auto-install

You publish an application with a Group Policy object, and a user who is managed by the Group Policy object logs on. The user has never installed the published application, and when he or she double-clicks a document associated with the published application, the **Open with** dialog box displays.

Possible Causes:

- You did not set the package to **Auto-Install** in the Software Installation snap-in.
- The user is running Terminal Services on the desktop.
- The Group Policy object does not apply to the user. For example, the user is filtered out of the Group Policy object by a security group.
- The user has not logged on since the Group Policy object was created.
- Group Policy or the Appmgmt extension did not run.
- The user cannot access Active Directory.
- The user cannot access the software distribution point.

Diagnostic Tests:

If the application appears in the **Add New Programs** list box of **Add/Remove Programs** in Control Panel, then, typically, the cause is that you did not select the **Auto-Install** check box in the Software Installation snap-in.

Open the Software Installation snap-in for the Group Policy object and check to see that the **Auto-install** check box is selected.

Run Addiag.exe to see if Terminal Services is running.

Open the Group Policy object with the Group Policy snap-in, and make sure the user is not being filtered out because he or she does not have Read or Apply Group Policy permissions.

Instruct the user to log off and then log back on. Verify that the user does not log on with cached credentials—that is, that the user is authenticated by a domain controller.

Run Gpresult.exe, and verify that the Appmgmt extension runs.

Unexpected Application Automatically Installs

You publish an application with a Group Policy object, and a user who is managed by the Group Policy object logs on. The user has never installed the published application, and when he or she double-clicks a document associated with the application, a different application than the expected one installs. For example, the user double-clicks a document with an .htm file name extension, expecting Microsoft® Word to open it, and instead, the Windows Installer installs Microsoft® FrontPage® and then opens the document.

Possible Causes:

You deployed multiple applications that want to own the file name extension but did not adjust the precedence of file name extensions in the Software Installation snap-in.

Diagnostic Tests:

Check to see that the precedence of the file name extensions for the Group Policy object with the Software Installation snap-in is set the way that you want it to work for users managed by the Group Policy object.

Note Checking Event Viewer for the application log does not provide any useful information for this problem because the unexpected application installs correctly. It is not an installation error, but rather a case of correctly installing an unexpected application.

Installation Error Messages

You assign or publish an application with a Group Policy object, and a user has never installed the managed application. The user selects the application to install. The installation begins and one of many error messages appears. This happens without regard to how the user selects the application for installation. It might be from **Add/Remove Programs** in Control Panel, from a shortcut on the desktop or the **Start** menu, or by Auto-install.

Possible Causes:

- There are problems with the Windows Installer package.
- The user does not have the appropriate permissions to read the Windows Installer package from the software distribution point, nor to install the application to the installation target folder as defined in the Windows Installer package.

Diagnostic Tests:

Copy the file to the computer, and see if the package installs when the Windows Installer starts from the command line. Test the package to see if it installs on a computer with a similar configuration.

Verify that the user has read permissions to the software distribution point, and Write access to the installation target directory as defined in the Windows Installer package.

Verify that the Windows Installer service is running.

Note If the managed application is advertised on the computer, or is visible in **Add/Remove Programs** in Control Panel, Group Policy and the Appmgmt extension run correctly. Therefore, the error messages come from Windows Installer.

Feature Is Not Found

You assign or publish an application with a Group Policy object, and a user attempts to install the application but receives the following error message:

The feature you are trying to install cannot be found in the source directory.

Possible Causes:

- The user cannot access the software distribution point because there is a network problem.
- The user cannot access the software distribution point because he or she has incorrect access rights.

Diagnostic Tests:

Check that the user has read access to the software distribution point.

Verify that the folder on the software distribution point with the managed application is shared.

Verify that the folder on the software distribution point with the managed application allows users who are managed by the Group Policy object to read the contents of the folders.

▶ **To test the network connectivity to the software distribution point:**

1. Ping the software distribution point by IP address to test base-level IP connectivity.
2. Ping the software distribution point by server name to test Domain Name System (DNS) name resolution.
3. Use the **net view** command to view the software distribution point by server name to see the available shares. This also confirms whether or not the user has rights to access the share.

Computer-Assigned Applications Do Not Install

You assign an application to a computer managed by a Group Policy object, and a user restarts the computer. When he or she logs on to the computer, the computer's assigned application does not install.

Possible Causes:

- Computer (machine) Group Policy is not applied—that is, the computer has not been restarted since the application was assigned to the computer.
- The Group Policy object does not apply to the computer. For example, the computer is in a different organizational unit, or the computer is filtered out of the Group Policy object by a security group.
- Group Policy did not run.
- The Appmgmt extension did not run.
- The computer is not able to access Active Directory.
- The computer is not able to access the software distribution point.

Diagnostic Tests:

Restart the computer and open the Group Policy object with the Group Policy snap-in to make sure the Group Policy object manages the computer.

Run the Gpresult.exe tool to see if Group Policy took effect.

Check to see if the Appmgmt extension ran, by running the Addiag.exe tool.

▶ **To check that the computer can access Active Directory**

1. Ping the domain controller by IP address to test base-level connectivity.
2. Ping the domain controller by server name to test DNS.
3. Use the **net view** command to view the *ServerName*\Sysvol share to test that the computer can read the Sysvol.

▶ **To check that the computer can access the software distribution point**

1. Ping the software distribution point by IP address to test base level connectivity.
2. Ping the software distribution point by server name to test DNS.
3. Use the **net view** command to view the *\\ServerName\AppShare* share to test that the computer can read the software distribution point.

Note There are cases where installation of an assigned application can cause the computer to stop responding during restart. If the application package requires user intervention, but the Windows Installer cannot display the necessary user interface, the Windows Installer is effectively blocked. If a computer with an assigned application continually stops during startup for no apparent reason, remove the assigned application. Then, either reassign the application, allowing the user access to the interface, or deploy the application in such a way that user input is not needed. It is always recommended that you test installations on another computer before deployment.

Installed Application Is Unexpectedly Removed

You assign applications using Group Policy and a user installs the assigned applications. When the user logs on (in the case of user-assigned applications), or the user restarts the computer (in the case of computer-assigned applications), the applications are removed.

Possible Causes:

- You selected the **Remove when policy does not apply** check box, and the scope of management is changed—that is, the Group Policy object that contains the setting is no longer in effect.
- The software is managed by site, and the computer moved to a new site.

Diagnostic Tests:

Check to see if the Group Policy object with the managed application still applies to the user or computer.

Check to see if the computer is in a new site, and the Group Policy object with that site does not have the application.

Note If a user or computer is moved from one organizational unit to another, or from one site to another, and if all aspects of the managed application do not match (for example, the package and the transform as well as the software distribution point), then you can remove and reinstall the application. This works even if the application appears to be managed by both Group Policy objects.

Opening Application Installs New Application

You assign or publish an application with a Group Policy object, and a user installs the managed application. When the user opens the application, a new application installs. For example, opening Microsoft Word starts Microsoft FrontPage installation.

Possible Causes:

- You set **Auto-Install** when configuring the application in the Software Installation snap-in.
- You created an upgrade relationship between the existing (base) application and the new (upgrading) application.
- The **Download missing CLSID from Active Directory** Group Policy is enabled.
- The application that starts shares a class identifier (CLSID) with the published application.

Diagnostic Tests:

Check the file name extension priorities for the Group Policy object.

Check for an upgrade relationship to the base application.

Start the Group Policy snap-in and check for the **Download missing CLSID from Active Directory** Group Policy.

Check the CLSIDs for the two applications for duplicate CLSIDs.

Note Checking Event Viewer for the application log does not provide any useful information for this problem because the new application installs correctly. It is not an installation error; rather, it is a case of correctly installing the wrong application.

Shortcuts Still Appear for Removed Application

You assign or publish an application with a Group Policy object, and a user installs the application. The user then removes the application or Software Installation and Maintenance removes the application. After the software is removed, the shortcuts for the application still appear on the desktop or on the **Start** menu.

Possible Causes:

- A user creates shortcuts for the application that the application's Windows Installer package is not aware of, and therefore cannot remove.
- Automatic upgrade of the application leaves shortcuts for the base application —that is, the application being upgraded.
- Another Group Policy object is advertising the application.

Diagnostic Tests:

If you click the shortcuts and they no longer point to an application—that is, they neither start the application nor start the installation of the application—then they are broken links. Delete the shortcuts and see if they reappear or if they stay deleted.

Check to see if there is a new version of the application. For example, the broken links might point to Microsoft® Word 97, and there is a working shortcut to Microsoft® Word 2000. Delete the shortcuts and see if they reappear or if they remain deleted.

If you delete the broken links, and they reappear, then a Group Policy object is advertising the links. If they do not install the application, see "Computer-Assigned Applications Do Not Install" earlier in this chapter.

Unexpected Applications Appear in Add/Remove Programs

You publish applications with a Group Policy object, and a user who is managed by the Group Policy object logs on. The user opens **Add/Remove Programs** in Control Panel, and then clicks the **Add New Programs** list box. The application that he or she expects to see does not appear in the list of applications that the user can install.

Possible Causes:

- The user is filtering the display by categories, and the application he or she is looking for is not in the current view.
- The application that the user is looking for is the base (existing) application in an upgrade relationship—that is, you want users to install only the upgrading application; therefore, the base application no longer appears.
- The Group Policy object that manages the user (the Group Policy object with the published application) is not being changed or overridden by another Group Policy object.

Diagnostic Tests:

Change the category and see if the application is visible.

Install the upgrading application that does appear in the list box.

Make sure that the resulting set of Group Policy that applies to the user still publishes this application.

Upgrading Base Application Does Not Complete

You create an upgrade relationship between a base (existing) managed application and a new (upgrading) application, and a user who is managed by the Group Policy object logs on. The user opens the base application, triggering the upgrade, and the base application is removed. Then the installation of the upgrading application does not complete.

Possible Causes:

- There are problems with the Windows Installer package of the upgrading application.
- The user does not have the appropriate permissions to read the Windows Installer package of the upgrading application from the Software Distribution Point, nor to install the application to the installation target folder as defined in the Windows Installer package.

Diagnostic Tests:

Copy the upgrading application Windows Installer package to the computer, and see if the package installs when Windows Installer is started from the command line. Test the package, and see if it will install on a computer with a similar configuration.

Verify that the user has read permissions to the software distribution point, and Write access to the installation target directory as defined in the Windows Installer package.

Note If the base application is removed, this indicates that Group Policy and the Appmgmt extension run correctly. The upgrading application is probably the problem. There might be useful entries in the Event Viewer application log about the Windows Installer attempting to install the upgrading application.

"Another Install in Progress" Error Message

You assign or publish applications with a Group Policy object, and a user attempts to install the application, but receives the following error message:

Another install in progress...

Possible Causes:

- Windows Installer is already running another installation.

Diagnostic Tests:

Wait for the installation to complete, and then try again.

Opening Application Starts Windows Installer

You assign or publish a managed application with a Group Policy object, which a user installs. When the user opens the application, the Windows Installer starts.

Possible Causes:

- The application is being repaired.
- The application was assigned to the computer, and this user has not used the application, so there is user configuration that needs to complete.
- A feature is being added to the installation.
- The application is the base application of an upgrade relationship, so it is being upgraded.

Diagnostic Tests:

In all cases, allow Windows Installer to complete.

Check Event Viewer for the application log to see why Windows Installer starts.

Cannot Prepare Package for Deployment

You open a Group Policy object to manage software with the Software Installation snap-in, and browse to the software distribution point and select the package you want to manage. When you click the **Open** button you receive the following error message:

Cannot prepare the package for deployment. Please check the package for possible corruption.

Possible Causes:

- The Software Installation snap-in is communicating with the Sysvol share.
- The package is corrupted.

Diagnostic Tests:

▶ **To check to see that the Sysvol can be connected to**

1. Ping the computer with the Sysvol by the computer's IP address to test base-level IP connectivity.
2. Ping the computer with the Sysvol by server name to test Domain Name System (DNS) name resolution.
3. Use the **net view** command to view the computer with the Sysvol by server name to see the available shares. This also confirms if the user has rights to access the share.

▶ **To check the package**

1. Install the package on another computer and make sure that the package can be opened.
2. Install the package by starting the Windows Installer from the command line. **C:>msiexec /i** *PackageName*
3. Test the package to see if it installs on a computer with a similar configuration.

Active Directory Does Not Allow Package to Deploy

You open a Group Policy object to manage software with the Software Installation snap-in, you browse to the software distribution point, select the package you want to manage, and open the package. When you click **OK** to manage (deploy) the package, you receive the following error message:

Active Directory will not allow the package to be deployed.

Possible Causes:

- The Software Installation snap-in is communicating with Active Directory.
- The package is corrupted.

Diagnostic Tests:

▶ **To check to see that Active Directory can be connected to**

1. Ping the computer with Active Directory by the computer's IP address to test base-level IP connectivity.
2. Ping the computer with Active Directory by server name to test Domain Name System (DNS) name resolution.
3. Use the **net view** command to view the computer with Active Directory by server name to see the available shares. This also confirms if the user has rights to access the share.

▶ **To check the package on another computer and make sure that the package can be opened**

1. Install the package by starting the Windows Installer from the command line. **C:>msiexec /I** *PackageName*
2. Test the package to see if it will install on a computer with a similar configuration.

Note For complete information about the Msiexec.exe command-line parameters, see the Windows Installer information on the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

User Settings Management Issues

You might encounter difficulties with roaming user profiles, which should allow a uniform experience for a user who logs on to different computers.

Roaming User Profile Does Not Roam Correctly Using Multiple Computers

You configure a user to use roaming user profile, and the user logs on to several computers in varied sequence and logs off in a random order from these computers. The user's roaming user profile does not roam across the multiple computers as the user anticipates.

Possible Cause:

A user logs on to multiple computers at the same time. When logging off from these computers, if the user logs off in a different order, this can cause odd behavior with the roaming user profile. The user fails to receive his or her roaming user profile on one or more of the computers.

Diagnostic Tests:

▶ **To check to see if a user receives his or her roaming user profile**

Perform the following steps on each of the computers on which the user logs on:

1. Right-click **My Computer** on the desktop.
2. Click **Properties** on the shortcut menu.
3. Click the **User Profiles** tab.

   Depending on the number of users who have logged on to each specific computer, there might be a different number of users listed on this page.
4. Locate the user entry on this list below **Profiles stored on this computer**.
5. If the Type is **Roaming** on all the computers, the user is successfully receiving his or her roaming user profile.
6. If the Type is **Local,** the user is not receiving his or her roaming user profile.

▶ **To find out why the roaming user profile is not available**

1. Ping the file server that contains the roaming user profile to confirm network connectivity.
2. Check to see that the user has enough file security to write his or her profile to this network location.
3. Check for a registry leak. A registry leak is a fault caused when the registry is not properly closed after it is accessed. When a registry leak occurs, the user profile cannot roam. Look for Event Viewer entries similar to the following:

   ```
   Windows cannot unload your registry class file. If you have a roaming
   profile, your settings are not replicated. Contact your
   administrator.
   ```

4. Open the Event Viewer snap-in on the user's computer (Eventwr.msc).
5. Look for the following event in the application log:

   ```
   Event ID: 1000
   Category: None
   Windows cannot unload your registry file.  If you have a roaming
   profile, your settings are not replicated. Contact your
   administrator.
   ```

Roaming User Profile Lost and User Left with Temporary Profile

A user logs on and his or her configuration and settings from his or her roaming user profile are not present. Instead, the user is left with a temporary profile.

Possible Causes:

- The user registry hive has been corrupted.
- It is the first time the user has logged on to a computer, and the roaming user profile is not available.

Diagnostic Tests:

Check to see if the user registry hive is corrupted. If this is the case, the user's local profile folder is renamed *Username*.bak. There are events written in Event Viewer when the user's registry hive is corrupted. Open the Event Viewer snap-in on the user's computer (Eventvwr.msc), and check the system log for the following events:

```
Event ID:    1000
Category:    None
Text:    Windows failed to load your profile because it may be
corrupted. Contact your Administrator.

Event ID:    1000
Category:    None
Text:    Windows cannot find the profile and is logging you on with a
temporary profile. Changes you make to this profile will be lost when
you log off.
```

▶ **To check to see that the user can connect to the server that contains the appropriate roaming user profile file share**

1. Ping this computer by the server's IP address to test base-level IP connectivity.
2. Ping this computer by server name to test Domain Name System (DNS) name resolution.
3. Use the **net view** command to view this computer to see the available shares (this also confirms if the user has rights to access). You should be able to see the share name that stores the roaming user profile.

Not All Settings Roam

A user is configured to use a roaming user profile. The user logs on from different computers and notices that most settings and configurations follow him or her, but some application settings, such as the personal dictionary for Microsoft® Office 95, do not roam.

Possible Causes:

- Applications are not storing user settings and configuration information to the roaming user profile.
- Applications are storing user settings and configuration to HKEY_LOCAL_MACHINE in the registry instead of correctly storing this information to HKEY_CURRENT_USER.

Diagnostic Tests:

Identify the user settings and configurations that are not roaming with the user from computer to computer.

Identify where this information is stored—in the registry, in a folder, in the roaming user profile, or in any other location.

If the application is indeed storing user settings and configuration information to places other than the roaming user profile or HKEY_CURRENT_USER, contact your application vendor to correct this.

User Profile Does Not Roam

A user with a roaming user profile regularly uses several computers. As the user moves from computer to computer his or her configuration and settings generally follow the user correctly. But on one occasion, the user logs on and some configuration information or settings are missing because the roaming user profile did not roam.

Possible Causes:

- Network connectivity problems make the server storing the roaming user profile unavailable.
- Network connectivity is lost while the roaming user profile is downloading.
- There are insufficient file access privileges to the roaming user profile share and folder.
- Roaming user profiles does not support roaming with encrypted files.
- There is a registry leak.
- The user exceeded his or her disk quota for the roaming user profile.
- The roaming user profile size is limited by an Administrative Template Group Policy setting.

Diagnostic Tests:

▶ **To check to see that the user can connect to the server that contains the roaming user profile file share**

1. Ping this computer by the server's IP address to test base-level IP connectivity.
2. Ping this computer by server name to test Domain Name System (DNS) name resolution.
3. Use the **net view** command to view this computer to see the available shares (this also confirms if the user has rights to access). You should be able to see the share name that stores the roaming user profile.

If the server that contains the roaming user profile is unavailable at logon, the following message might be generated in the event log:

```
Event ID:    1000
Category:    None
Source:        userenv

Text:    Windows cannot load your profile. Contact your network
administrator.
```

Check if network connectivity is lost when the roaming user profile is copying. You can find this information by using Eventvwr.msc to look for the two following Event Viewer messages:

```
Event ID:    1000
Category:    None
Source:        userenv

Text:    Windows cannot copy file <> to the location <>. Contact your
network administrator.

Event ID:    1000
Category:    None
Source:        userenv

Text:    Windows cannot load the profile and is logging you on with a
temporary profile. Changes you make to this profile will be lost when
you log off.
```

Check to see if the user has enough security file access rights to his or her roaming user profile. If the user has insufficient file security, the following message will be generated to Event Viewer:

```
Event ID:    1000
Category:    None
Source:        userenv

Text:    You do not have permission to access your central profile
located at <>. Contact your network Administrator.
```

Check to see if the user has any encrypted files that use the Windows 2000 Encrypting File System (EFS) in his or her roaming user profile. Roaming user profiles do not support roaming with encrypted files. If the user profile contains encrypted files, the following message is generated in the event log:

```
Event ID:    1000
Category:    None
Source:        userenv

Text:    Windows cannot copy your profile because it contains encrypted
files or directories. The keys to decrypt the files or directories are
also stored in the profile and are not available now. Please decrypt the
files and try again.
```

Check for a registry leak on the user's computer. You can find this by looking at Event Viewer:

```
Event ID:    1000
Category:    None
Source:        userenv

Text:    Windows cannot unload your registry file. If you have a roaming
profile, your settings are not replicated. Contact your administrator.
```

▶ To see if a disk quota has been exceeded

1. Make note of the size of the roaming user profile.
2. Check to see if there is a disk quota on the server that stores the roaming user profile.
3. Check to see if you have applied a Group Policy setting to limit roaming user profile size. The Group Policy setting that can limit the roaming user profile size is AdministrativeTemplates\UserSettings\System\LogonLogoff\Limit Profile Size.

▶ **To see if this Group Policy setting has been applied**

1. Run Gpresult.exe in verbose mode.
2. Check the output to see if either of these registry keys exists:

   HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows NT\Disk Quotas\Limit

   HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows NT\Disk Quotas\Threshold

If a disk quota is exceeded, as shown earlier, the following event is generated in the event log:

```
Event ID:    1000
Category:    None
Source:        userenv

Text:    Windows cannot update your roaming profile. Contact your
network administrator.
```

Remote OS Installation Issues

You might encounter problems using Remote OS Installation when using Windows 2000. If installation of Windows 2000 to a remote computer over the network does not function, troubleshooting largely depends on understanding boot information negotiation layer (BINL) error messages.

OSChooser Skips User

The user attempts to use Remote OS Installation to install Windows 2000 on his or her computer, and the OSChooser screen does not allow the user to make any entries. Instead, it quickly skips by before the user can make any selections.

Possible Causes:

- There is only one operating system image, which is automatically selected, and the OSChooser screen only displays momentarily.
- The user accidentally presses ENTER twice or several times, and the OSChooser accepts the keystrokes in the buffer to mean that the user wants the defaults from the OSChooser screen.

Diagnostic Tests:

▶ **To check that there is only a single image for installation**

1. Log on to the server as an administrator.
2. Click **Start**, click **Programs,** and then click **Administrative Tools**.

3. Click **Active Directory Users and Computers**, and navigate to the container with the RIS server. Typically, this is Computers, but it could be Domain Controllers.
4. Right-click the RIS server, and then click **Properties**.
5. click the **Remote Install** tab, and then click **Advanced Settings**.
6. Click the **Images** tab. You see a list of all the images. If there is only a single image, no OSChooser screen appears.

File Not Found Before Welcome.osc

A user attempts to use Remote OS Installation to install Windows 2000 on his or her computer, but before the user receives the Welcome.osc screen, he or she receives the following error message:

File Not Found...

Possible Causes:

- The file *ServerName*\Reminst\Oschooser\Welcome.osc does not exist.
- The server is running with a locale other than the locale of the installed client operating system.

Diagnostic Tests:

If the Welcome.osc file does not exist, copy the Welcome.osc file from *ServerName*\Reminst\Oschooser*Lang*\Welcome.osc, or run the Risetup program from the command line with the **/check** parameter.

Follow the Remote OS Installation multilanguage instructions.

File Not Found After Welcome.osc

The user attempts to use Remote OS Installation to install Windows 2000 on his or her computer, but after the user receives the Welcome.osc screen, he or she receives the following error message:

File Not Found...

Possible Causes:

- There are customized OSChooser screens, and one of the screens is missing.
- There are no customized OSChooser screens, but one of the supplied screens is missing.
- The server is running with a locale other than the locale of the installed client operating system.

Diagnostic Tests:

Make sure all of the customized screens exist and are in the appropriate directory.

Reinstall the client operating system image, and then select **Overwrite the existing client OSChooser screens**.

Follow the Remote OS Installation multilanguage instructions.

Duplicate Computer Account Warning Message

The user attempts to use Remote OS Installation to install Windows 2000 on his or her computer, and there is already a Computer object for the computer in Active Directory. The user selects **Custom install**, and enters the same computer name, but receives a warning message about the duplicate Computer object.

Possible Causes:

- The existing Computer object has a globally unique identifier (GUID) that does not match the GUID of the client that the user attempts to install.
- The existing account is in an Active Directory location that is not the same as where the boot information negotiation layer (BINL) is creating a new Computer object.

Diagnostic Tests:

Delete the Computer object from Active Directory, and the Client Installation Wizard creates a new Computer object. Allow Active Directory replication to complete after deleting the Computer object and before trying the Client Installation Wizard again.

Enter the directory service path for the existing Computer object in Custom.osc, or use the **Automatic** setting and the existing account is reused automatically.

Risetup Fails Due to Insufficient Rights

You attempt to run Risetup.exe, but it fails because you have insufficient rights.

Possible Causes:

- You are not a domain user.
- You are not an administrator of the local computer.

Diagnostic Tests:

Log on as a domain user who has full control of the specific computer object to create and modify the Remote Installation Services (RIS) settings stored in Active Directory.

Log on as a local administrator who has Full Control to start and stop services on the local computer.

RPC Server Unavailable Error Message

You run Risetup.exe, or perform an administrative task on the Remote Installation server (RIS) using the Active Directory Users and Computers snap-in, or any administrative task that requires communication with Active Directory, and an error message appears that indicates that the RPC server is unavailable.

Possible Causes:

There is a problem with the Domain Name System (DNS) server.

Diagnostic Tests:

Run Netdiag to determine the DNS problem, and take appropriate steps to fix the DNS service.

BINL Server Does Not Respond and No Scope Problem Error Message Appears

The user attempts to use Remote OS Installation to install Windows 2000 on his or her computer, but no boot information negotiation layer (BINL) server responds to his or her request. An error message appears that indicates that a scope issue is preventing the BINL server from answering.

Possible Causes:

If the DHCP and BINL server are running on the same server, the DHCP server always sees the client request first; therefore, the DHCP server must have a scope defined for the client's subnet, or the packet will be dropped and the BINL server will never have an opportunity to respond.

Diagnostic Tests:

Either create a scope within DHCP or, if the DHCP service is not needed on this server, remove the DHCP service.

BINL Server Does Not Respond and Debugging Error Message Appears

The user attempts to use Remote OS Installation to install Windows 2000 on his or her computer, but no boot information negotiation layer (BINL) server responds to his or her request. An error message appears that indicates that the BINL server needs debugging.

Possible Causes:

If either the DHCP server or the BINL server is running under a debugger, both services must run under the debugger. In this case, the BINL server responds only to clients that are on the same subnet as the BINL server.

Diagnostic Tests:

Run the debugger on both the DHCP and BINL servers, or run the debugger on neither of them.

BINL Server Does Not Respond and Server Not Authorized Error Message Appears

The user attempts to use Remote OS Installation to install Windows 2000 on his or her computer, but no boot information negotiation layer (BINL) server responds to his or her request. An error message appears that indicates that the server is not authorized.

Possible Causes:

The DHCP server is installed but not authorized.

Diagnostic Tests:

Authorize the DHCP server.

Note The Event Viewer system log messages originate from the DHCP server, and reports that the service is not authorized. This message can appear on the server with Remote Installation Services even when the DHCP server is not on that server. It still lists the DHCP server as the source of the message.

BINL Server Does Not Respond and Unable to Read Active Directory Settings Error Message Appears

The user attempts to use Remote OS Installation to install Windows 2000 on his or her computer, but no boot information negotiation layer (BINL) server responds to his or her request. An error message appears indicating that the computer is unable to read Active Directory settings

Possible Causes:

- The server cannot access Active Directory.
- There is a problem with the rights assigned to the server's Computer object in Active Directory.

Diagnostic Tests:

Run Netdiag and correct any communication errors.

Start the BINL server in the security context of the domain administrator. If the problem resolves, there is a server rights problem.

BINL Server Does Not Respond and Server Not Set to Answer Error Message Appears

The user attempts to use Remote OS Installation to install Windows 2000 on his or her computer, but no boot information negotiation layer (BINL) server responds to his or her request. An error message appears that indicates that the server is not set to answer.

Possible Causes:

The RIS server is not set to answer client requests for service.

Diagnostic Tests:

▶ **To check that the RIS server is set to respond to clients**

1. Click **Start**, point to **Programs**, and then click **Administration Tools**.
2. Click **Active Directory Users and Computers** and navigate to the container that has the Computer object for the RIS server. This is typically either Domain Controllers or Computers.
3. Right-click the RIS server object, and then click **Properties**.
4. Click the **Remote Install** tab.
5. Make sure that either the **Respond to all clients requesting service** check box or the **Do not respond to unknown clients** check box is selected, as appropriate.

BINL Server Does Not Respond and Client Unknown Error Message Appears

The user attempts to use Remote OS Installation to install Windows 2000 on his or her computer, but no boot information negotiation layer (BINL) server responds to the request. An error message appears that indicates that the client is unknown.

Possible Causes:

- The server is set to answer only known clients—that is, clients that were prestaged in Active Directory.
- The client's globally unique identifier (GUID) does not match the GUID in Active Directory.

Diagnostic Tests:

The clients must have accounts in Active Directory with the correct GUID—that is, the associated GUID. You can add a Computer object by using the media access control address of the client network adapter, padded with 0s to form a 32-character GUID if the client does not ship with a built-in GUID. Be sure that every number and letter in the GUID match.

BINL Server Does Not Respond and Prestaged Clients Server Down Error Message Appears

The user attempts to use Remote OS Installation to install Windows 2000 on his or her computer, but no boot information negotiation layer (BINL) server responds to his or her request. An error message appears that indicates that the prestaged server is down.

Possible Causes:

The client is set for a specific RIS server to answer it, and the server is down.

Diagnostic Tests:

Start the RIS server, and edit the client's Computer object in Active Directory, removing the reference to the existing server that is down. Enter the path to a working RIS server, and set the client for any RIS server to answer by leaving the **Server** text box empty.

BINL Server Does Not Respond and DHCP Packets Not Forwarded Error Message Appears

The user attempts to use Remote OS Installation to install Windows 2000 on his or her computer, but no boot information negotiation layer (BINL) server responds to his or her request. An error message appears that indicates that the DHCP packets are not forwarded.

Possible Causes:

There is a router between the RIS server and the clients, and the router is not set to forward DHCP to clients.

Diagnostic Tests:

Follow the router's instructions for setting up a DHCP relay agent to send packets to the DHCP server. If the RIS server is also on the other side of the router, you might need to follow the same steps to forward the packets to the RIS server.

BINL Server Does Not Start and Unknown User Error Message Appears

You attempt to start the boot information negotiation layer (BINL) service, and an error message appears that indicates that the user is unknown.

Possible Causes:

There is a service rights issue.

Diagnostic Tests:

Start the BINL service (Binlsvc.exe) under the context of a domain administrator. If the error resolves, then you need to configure the rights of the server's computer object so the local system can read the Computer object for the computer and any necessary sub-objects.

Remote OS Installation Cannot Join Domain

The user attempts to use Remote OS Installation to install Windows 2000 on his or her computer, and during the graphical user interface (GUI) portion of the setup, the user cannot join the domain.

Possible Causes:

- The client does not register correctly with the DNS, or the DNS is not replicated.
- There are other Active Directory issues.

Diagnostic Tests:

In both cases, avoid joining the domain at this time and do so after installation. You might have to repair an Active Directory issue.

Gathering More Troubleshooting Information

There are some registry keys, listed here, whose values might be relevant when troubleshooting Change and Configuration Management. For more information about these registry keys, see the Technical Reference to the Registry on the *Resource Kit* companion CD.

You might also find that using the command-line tools with the specified switches is helpful in gathering more troubleshooting information.

User Data Management

To troubleshoot Folder Redirection and Offline Files issues, it might be helpful to examine the following Event Viewer errors and registry keys.

Event Viewer Folder Redirection Errors

Elogdmp %computername% application [| findstr /i userenv]
[| findstr "ERROR"]

Elogdmp %computername% application [| findstr /i /c:"Folder Redirection"]
[| findstr "ERROR INFO"]

Folder Redirection Registry Keys

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\
Explorer\User Shell Folders

Offline Files Registry Keys

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\NetCache

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\NetCache

Software Installation and Maintenance

To troubleshoot Software Installation and Maintenance issues, it might be helpful to view the following Event Viewer errors and registry keys.

Event Viewer Software Installation Errors

Elogdmp %computername% application [| findstr /i userenv] [| findstr "Application Management"]

Software Installation and Maintenance Registry Keys

Application Management Client-Side Extension:

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Group Policy\AppMgmt

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Group Policy\AppMgmt

Windows Installer:

HKEY_CURRENT_USER\Software\Microsoft\Installer

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Installer

---

Caution Do not use a registry editor to edit the registry directly unless you have no alternative. The registry editors bypass the standard safeguards provided by Administrative Tools. These safeguards prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your system. Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. To configure or customize Windows 2000, use the programs in Control Panel or Microsoft Management Console (MMC) whenever possible.

Software Installation and Maintenance Event Log Entries

The following are the events that are always logged to the event log, along with a brief description of when and why. These events are written to the application event log with the source name of **Application Management** and the corresponding user name for which the event applies.

```
MessageId=101 SymbolicName=EVENT_APPMGMT_ASSIGN_FAILED
The assignment of application %1 from policy %2 failed. The error was
%%%3.
```

This is logged when an application assignment fails.

```
MessageId=102 SymbolicName=EVENT_APPMGMT_INSTALL_FAILED
The install of application %1 from policy %2 failed. The error was %%%3.
```

This is logged when an application installation fails, either through computer assignment or **Add/Remove Programs** in Control Panel.

```
MessageId=103 SymbolicName=EVENT_APPMGMT_UNASSIGN_FAILED
The removal of the assignment of application %1 from policy %2 failed.
The error was %%%3.
```

This is logged when the removal of an assignment of an application fails.

```
MessageId=104 SymbolicName=EVENT_APPMGMT_UNINSTALL_FAILED
The removal of application %1 from policy %2 failed.  The error was
%%%3.
```

This is logged when an application removal initiated by Group Policy fails. This is not logged when a removal through **Add/Remove Programs** in Control Panel fails.

```
MessageId=105 SymbolicName=EVENT_APPMGMT_REINSTALL_FAILED
The reinstall of application %1 from policy %2 failed.  The error was
%%%3.
```

This is logged when a reinstallation initiated by Group Policy as the result of an application redeployment fails. This is not logged for any **Add/Remove Programs** action in Control Panel.

```
MessageId=106 SymbolicName=EVENT_APPMGMT_UPGRADE_ABORT
Application %1 from policy %2 was configured to upgrade application %3
from policy %4.  The removal of application %3 from policy %4 failed
with error %%%5.  The upgrade will be aborted.
```

This is logged when an upgrade is aborted because the previous application was not successfully removed. It is logged both for upgrades initiated by Group Policy and **Add/Remove Programs** in Control Panel.

```
MessageId=107 SymbolicName=EVENT_APPMGMT_ZAP_FAILED
The execution of the setup program for application %1 from policy %2
failed.  The setup path was %3 and the error was %%%4.
```

This is logged when the start of a Setup command specified in a .zap file deployment fails. Only the successful creation of the original Setup command is validated. There is no validation that the setup, after it is started successfully, will complete successfully.

```
MessageId=108 SymbolicName=EVENT_APPMGMT_POLICY_FAILED
Failed to apply changes to software installation settings.  %1  The
error was %%%2.
```

This is logged when Group Policy processing fails for the Appmgmt extension. This is only logged when processing changes to Group Policy. Any errors encountered while applying settings when Group Policy did not change are not logged. The **%1** string is one of four messages to better identify the source of the problem:

```
The Active Directory path could not be updated at the client.
```

This indicates that the Appmgmt.ini file stored under *SystemRoot*\System32\Appmgmt*Usersid* could not be updated. In practice, this should never occur.

```
The list of applications in the AD could not be retrieved.
```

This indicates that there is a problem querying Active Directory. This is the most common problem.

```
Local software installation state could not be retrieved.
```

The third instance indicates that local status information about currently applied applications cannot load. Typically, this only occurs if the computer is out of memory or there is corruption of the user's profile, particularly Ntuser.dat, in the HKEY_CURRENT_USER registry hive.

```
Software changes could not be applied.  A previous log entry with
details should exist.
```

This indicates that an earlier problem, normally encountered after the first three conditions are not hit, caused Group Policy to terminate—such as a failed assignment or removal of an application.

```
MessageId=150 SymbolicName=EVENT_CS_NETWORK_ERROR
A network error occurred accessing software installation data in the AD.
The error was %%%1.
```

This is logged when the Appmgmt layer that performs all Active Directory Services Interface (ADSI) operations encounters an error.

User Settings Management

To troubleshoot User Settings Management, it might be useful to examine the following Event Viewer errors and registry keys.

Event Viewer Roaming User Profile Errors

Elogdmp %computername% application [| findstr /i userenv]

Roaming User Profile Registry Keys

All User Profile Information:

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders

Current User Profile Information:

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders

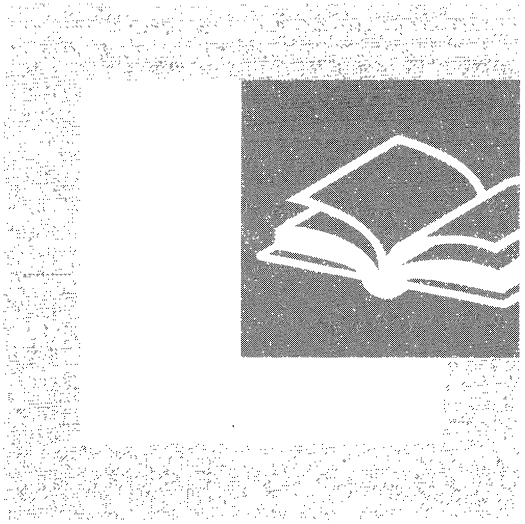
Exclusion List:

HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Winlogon ExcludeProfileDirs

HKEY_CURRENT_USER\Environment

PART 5

Appendixes

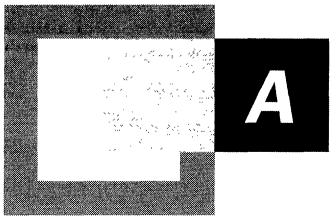


The appendixes provide detailed information about distributed systems in Microsoft® Windows® 2000. The additional information in this section is a useful source for network administrators who need technical reference material that is related to directory services, security, and desktop change and configuration management.

In This Part

APPENDIX A

Frequently Encountered LDAP API Functions



Some of the frequently encountered Lightweight Directory Access Protocol (LDAP) application programming interface (API) function calls are listed and described in Table A.1.

Table A.1 LDAP C-Binding APIs

API Name	Description
ldap_open	Opens a connection to an LDAP server.
ldap_bind	Authenticates a client to the LDAP server directory.
ldap_unbind	Frees all resources that are associated with an LDAP session.
ldap_search	Searches the LDAP directory and returns a requested set of attributes for each entry matched.
ldap_modify	Modifies an existing LDAP entry.
ldap_modrdn	Changes the relative distinguished name of an LDAP entry. (This function is obsolete and is provided solely for backward compatibility with LDAP 1 implementations.)
ldap_add	Adds an entry to the LDAP directory.
ldap_delete	Deletes an entry from the LDAP directory.
ldap_abandon	When a client calls it, cancels another in-progress asynchronous LDAP call.
ldap_result	Obtains the result of an asynchronous operation.
ldap_result2error	Interprets errors that are returned by other APIs.
ldap_err2string	Converts a numeric LDAP error code into a character string that describes the error.

(continued)

Table A.1 LDAP C-Binding APIs *(continued)*

API Name	Description
ldap_first_entry	Returns the first entry of a message.
ldap_next entry	Retrieves an entry from a search result chain.
ldap_count_entries	Counts the number of search entries that a server has returned.
ldap_first_attribute	For a specific directory entry, the function returns the first attribute.
ldap_next_attribute	For a specific entry, the function returns the next attribute.
ldap_get_values	Retrieves the values of a specific attribute for an object.
ldap_get_values_len	Retrieves the list of values for a specific attribute. The LDAP directory service uses this information when it parses a search response.
ldap_get_dn	Retrieves the distinguished name for a specific entry.
ldap_explode_dn	Breaks up an entry name into its component parts.
ldap_dn2ufn	Converts a distinguished name into a user-friendly format.

For information about all of the LDAP function calls, see the Microsoft Platform SDK link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

APPENDIX B

LDAP Requests for Comments

Table B.1 and Table B.2 are abridged lists of important Requests for Comments (RFCs) that apply to the Lightweight Directory Access Protocol version 2 (LDAPv2) and the Lightweight Directory Access Protocol version 3 (LDAPv3).

Table B.1 LDAPv2 RFCs

RFC	Description
RFC 1777	"Lightweight Directory Access Protocol" The protocol that is described in this document is designed to provide access to directories that support X.500 models without incurring the resource requirements that are incurred by the Directory Access Protocol (DAP). This protocol is specifically designed for management applications and browser applications that provide simple read/write, interactive access to an X.500 directory. It is intended to complement the DAP. For more information about RFC 1777, see the Request for Comments (RFC) link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.
RFC 1778	"The String Representation of Standard Attribute Syntaxes" LDAP requires that the contents of **AttributeValue** fields in protocol elements be octet strings. This document defines the requirements for the encoding rules that are used to render X.500 directory attribute syntaxes into a form that is suitable for LDAP. The document then defines the encoding rules for the standard set of LDAP attribute syntaxes.
RFC 1779	"A String Representation of Distinguished Names"

(continued)

Table B.1 LDAPv2 RFCs *(continued)*

RFC	Description
	The OSI directory uses distinguished names as the primary means of identifying entries in the directory. Distinguished names are encoded in ASN.1. When a distinguished name is communicated between two users who are not using a directory protocol (for example, in a mail message), a user-oriented string representation of the distinguished name is necessary. This RFC defines a string format for representing distinguished names. The specification is designed to provide unambiguous representations of all distinguished names.
RFC 1959	"An LDAP URL Format"
	This document describes a format for an LDAP Uniform Resource Locator (URL) that allows Internet clients to have direct access to the LDAP protocol. Although LDAP currently is used only as a front end to an X.500 directory, the URL format that is described here is general enough to handle stand-alone LDAP servers (that is, LDAP servers that do not have an X.500 back-end server). *This RFC has been superseded by RFC 2255 for LDAPv3.*
RFC 1960	"A String Representation of LDAP Search Filters"
	LDAP defines the network representation for a search filter that is transmitted to an LDAP server. For some applications, it might be useful to have a way of representing search filters in a readable form that can be recognized by everyone. This document defines a readable string format for representing LDAP search filters. *This RFC has been superseded by RFC 2254 for LDAPv3.*
RFC 1823	"The LDAP Application Program Interface"
	This document provides information about a C-language application program interface (API) to LDAP. This API is designed to be both powerful and simple to use. It defines compatible synchronous and asynchronous interfaces to LDAP that suit a wide variety of applications. The document contains a brief overview of the LDAP model and how the API is used by an application to obtain LDAP information. The document contains detailed descriptions of LDAP API calls, followed by an appendix in which code examples illustrate the use of the API.

Table B.2 LDAPv3 RFCs

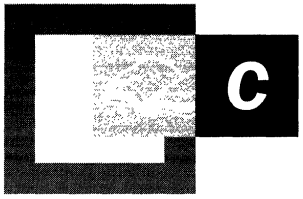
RFC	Description
RFC 2251	"Lightweight Directory Access Protocol (v3)" The protocol that is described in this document is designed to provide access to directories that support X.500 models without incurring the resource requirements that are incurred by the Directory Access Protocol (DAP). This protocol is specifically designed for management applications and browser applications that provide simple read/write, interactive access to X.500 directories. When LDAPv3 is used with a directory that supports X.500 protocols, it complements the X.500 DAP. For more information about RFC 2251, see the Request for Comments (RFC) link on the Web Resources page at http://windows.microsoft.com /windows2000/reskit/webresources.
RFC 2252	"Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions" LDAP requires that the contents of **AttributeValue** fields in protocol elements be octet strings. This document defines a set of syntaxes for LDAPv3 and the rules by which attribute values of these syntaxes are represented as octet strings for transmission in the protocol. The syntaxes that are defined in this document are referenced by this document and other documents that define attribute types. This document also defines the set of attribute types that LDAP servers must support.
RFC 2253	"Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names" The X.500 directory uses distinguished names as the primary means of identifying entries in the directory. Distinguished names are encoded in ASN.1 in the X.500 directory protocols. In LDAP, a string representation of distinguished names is transferred. This RFC defines the string format for representing distinguished names. The specification is designed to provide unambiguous representations of all distinguished names.
RFC 2254	"The String Representation of LDAP Search Filters" LDAP defines the network representation of a search filter that is transmitted to an LDAP server. For some applications, it might be useful to have a common way of representing search filters in a readable form. This document defines a readable string format for representing LDAP search filters. *This document supersedes RFC 1960.* *(continued)*

Table B.2 LDAPv3 RFCs *(continued)*

RFC	Description
RFC 2255	"The LDAP URL Format" This document describes a format for an LDAP URL. The format describes an LDAP search operation for retrieving information from an LDAP directory. It updates the LDAP URL format for LDAPv3 and clarifies how LDAP URLs are resolved. This document also defines an extension mechanism for LDAP URLs that makes it possible for future documents to extend their functionality (for example, to provide access to new LDAPv3 extensions as they are defined). *This document supersedes RFC 1959.*
RFC 2256	"A Summary of the X.500(96) User Schema for Use with LDAPv3" This document is an overview of the attribute types and object classes that are defined by the ISO and ITU-T committees in X.500 RFCs, in particular the attribute types and object classes that are intended for use by directory clients. This is the most widely used schema for LDAP directories and X.500 directories; many other schema definitions for white pages objects also use this schema as a base. This document does not cover attributes that are used for the administration of X.500 directory servers, nor does it include attributes that are defined by other ISO and ITU-T RFCs.

APPENDIX C

Active Directory Diagnostic Tool (Ntdsutil.exe)



Ntdsutil.exe is a command-line tool that provides management facilities for Active Directory™, the Microsoft® Windows® 2000 directory service. Use Ntdsutil to perform database maintenance of Active Directory, to manage and control single master operations, and to remove metadata left behind by domain controllers that were removed from the network without being properly uninstalled. This tool is intended to be used by experienced administrators. By default, Ntdsutil is installed in the Winnt\System32 folder.

In This Appendix

Invoking Ntdsutil Commands and Parameters

Ntdsutil provides menus that list the set of commands for the program. At any level, you can enter **?** or **Help** (or even **h** if there are no other options that start with **h** at that level) to read the list of commands for that part of the program. The command **quit** (or **q,** if sufficiently unique) is the universal command to return to the prior menu. The command **quit**, when used at the outermost level, exits the program.

For more information about Ntdsutil menu commands, see "List of Menu Commands" later in this appendix.

How to Use Ntdsutil Menu Commands

You can invoke Ntdsutil from the command prompt with no arguments. Rather than support and extend an ever-increasing set of cryptic command-line arguments, the tool parses keyboard input after you invoke it. The commands have been made as conversational as possible. For example, you can type the following:

list roles for connected server

connect to server *xxx*

For convenience, Ntdsutil allows you to be cryptic. You need only specify enough of each word to make it unique with respect to any other words that you can enter at that time. Thus, as you become more familiar with the tool, you might type the following:

li r f c s

rather than:

list roles for connected server

How Ntdsutil Processes Command Input

Ntdsutil processes as input all the arguments that you type when you start the program. For example, if you type the following:

ntdsutil help connections help quit quit

Ntdsutil does the following steps:

1. Invokes Ntdsutil.exe.
2. Displays its Help information.
3. Enters the Connections submenu.
4. Displays its Help information.
5. Closes the **Connections** submenu and returns to the top-level menu.
6. Quits the program.

How to Use Arguments with Ntdsutil Commands

Some commands take arguments that are shown as either **%s** or **%d** in Help. As you probably know, **%d** is the decimal number specifier and **%s** is the string specifier for the C-language program print and scan commands. When you enter a command whose Help indicates you use either a **%d** or **%s**, enter a number or string respectively. For example, one Help listing shows the following:

connect to server %s

You would type this command as the following:

connect to server *xxx*

where *xxx* is the character string you want to substitute for **%s**. If your string has spaces in it, enclose it in quotation marks as follows:

connect to server "*xxx yyy*"

How to Automate Ntdsutil Commands

You can automate Ntdsutil by creating batch files or scripts that contain a series of Ntdsutil commands. Many Ntdsutil commands that perform writes, open by default a message that asks users if they really want to perform a particular operation. When these messages appear, the program will pause and wait for keyboard input. Use the Popups %s command to disable these messages when running Ntdsutil from a batch file or script. For example, to disable these messages, type the following:

popups no

To reenable the display of these messages, type the following:

popups yes

It is good practice to disable these messages only when you are scripting Ntdsutil commands and to reenable them as soon as you finish scripting.

Managing Active Directory Files

The Microsoft® Windows® 2000 directory service is implemented on top of an indexed sequential access method (ISAM) table manager. This is the same table manager used by Microsoft® Exchange Server, the file replication service, the security configuration editor, the certificate server, Windows Internet Name Service (WINS), and other Windows 2000 components. The version of the database that Windows 2000 uses is called extensible storage engine (ESENT).

ESENT is a transacted database system that uses log files to support rollback semantics to ensure that transactions are committed to the database. Ideally, data and log files should be located on separate drives to improve performance and support recovery of the data if a disk fails.

The data file is called Ntds.dit. The Files menu of Ntdsutil provides commands for managing the directory service data and log files.

ESENT provides its own tool for certain database file management functions called Esentutl.exe, which is also installed in the Winnt\System32 folder. Several of the Ntdsutil file management commands invoke Esentutl, reducing the need to learn that tool's command-line arguments. In the cases where Ntdsutil invokes Esentutl, it brings up a separate window configured with a large history so that you can scroll back to see all of the Esentutl progress indicators.

The Windows 2000 directory service opens its files in exclusive mode. This means the files cannot be managed while the system is operating as a domain controller.

▶ To manage directory service files

1. Start the computer.
2. When the **Starting Windows** progress bar appears, press F8.
3. From the **Windows 2000 Advanced Options Menu**, select **Directory Services Restore Mode**.

Note Starting the computer in Directory Services Restore Mode causes your domain controller to temporarily operate as a stand-alone server. This causes some services to fail, especially those that are integrated with the directory service. When operating in this mode, the security accounts manager (SAM) uses a minimal set of user and group definitions stored in the registry. If your domain controller is not physically secure, you should set the administrative password for the Directory Services Restore Mode.

Table C.1 lists and describes the file management commands.

Table C.1 File Management Commands

Command	Description
Compact to %s (where %s identifies an empty target directory)	Invokes Esentutl.exe to compact the existing data file and writes the compacted file to the specified directory. The directory can be remote, that is, mapped by means of the **net use** command or similar means. After compaction is complete, archive the old data file, and move the newly compacted file back to the original location of the data file. ESENT supports online compaction, but this compaction only rearranges pages within the data file and does not release space back to the file system. (The directory service invokes online compaction regularly.)
Header	Writes the header of the Ntds.dit data file to the screen. This command can help support personnel analyze database problems.

(continued)

Table C.1 File Management Commands *(continued)*

Command	Description
Info	Analyzes and reports the free space for the disks that are installed in the system, reads the registry, and then reports the sizes of the data and log files. (The directory service maintains the registry, which identifies the location of the data files, log files, and directory service working directory.)
Integrity	Invokes Esentutl.exe to perform an integrity check on the data file, which can detect any kind of low-level database corruption. It reads every byte of your data file; thus it can take a long time to process large databases. Note that you should always run Recover before performing an integrity check.
Info	Analyzes and reports the free space for the disks that are installed in the system, reads the registry, and then reports the sizes of the data and log files. (The directory service maintains the registry, which identifies the location of the data files, log files, and directory service working directory.)
Integrity	Invokes Esentutl.exe to perform an integrity check on the data file, which can detect any kind of low-level database corruption. It reads every byte of your data file; thus it can take a long time to process large databases. Note that you should always run Recover before performing an integrity check.
Move DB to %s (where %s identifies a target directory)	Moves the Ntds.dit data file to the new directory specified by **%s** and updates the registry so that, upon system restart, the directory service uses the new location.

(continued)

Table C.1 File Management Commands *(continued)*

Command	Description
Move logs to %s (where %s identifies a target directory)	Moves the directory service log files to the new directory specified by **%s** and updates the registry so that, upon system restart, the directory service uses the new location.
Recover	Invokes Esentutl.exe to perform a soft recovery of the database. Soft recovery scans the log files and ensures all committed transactions therein are also reflected in the data file. The Windows 2000 Backup program truncates the log files appropriately. Logs are used to ensure committed transactions are not lost if your system fails or if you have unexpected power loss. In essence, transaction data is written first to a log file and then to the data file. When you restart after failure, you can rerun the log to reproduce the transactions that were committed but hadn't made it to the data file.
Repair	Invokes Esentutl.exe to perform a low-level repair of the data file. Use the repair command only on the advice of qualified service personnel, as it can cause data loss. Furthermore, this can only repair what ESENT knows about. This means that its notion of repair might eliminate some data that is key to the safe operation of the directory service.
Set path backup %s (where %s identifies a target directory)	Sets the disk-to-disk backup target to the directory specified by **%s**. The directory service can be configured to perform an online disk-to-disk backup at scheduled intervals.
Set path DB %s (where %s identifies a target directory)	Updates the part of the registry that identifies the location and file name of the data file. Use this command only to rebuild a domain controller that has lost its data file and that is not being restored by means of normal restoration procedures.

(continued)

Table C.1 File Management Commands *(continued)*

Command	Description
Set path logs %s (where %s identifies a target directory)	Updates the part of the registry that identifies the location of the log files. Use this command only if you are rebuilding a domain controller that has lost its log files and is not being restored by means of normal restoration procedures.
Set path working dir %s (where %s identifies a target directory)	Sets the part of the registry that identifies the directory service's working directory to the directory specified by **%s**.

Using the Connections Menu

Several Ntdsutil operations send LDAP or RPC commands to a particular server. The **Connections** menu appears as a selection within several other menus in Ntdsutil and provides a way to connect to a specific server or domain. By default, your connection is authenticated using the credentials of the logged-on user. You can also specify the credentials to use when authenticating to a server. Table C.2 lists and describes the connection commands.

Table C.2 Connection Commands

Command	Description
Clear creds	Clears any previously defined credentials and disconnects any previous connections to prevent ambiguity about which credentials are in use during subsequent commands.
Connect to domain %s (where %s identifies a target domain)	Finds any domain controller for the domain specified by **%s** and connects to it using the default credentials or any credentials specified earlier by **Set creds**.
Connect to server %s (where %s identifies a target domain controller)	Connects to the domain controller specified by **%s** using the default credentials or any credentials specified earlier by **Set creds**.
Info	Displays the credentials now in use and the current connection state.
Set creds %s %s %s (where the first %s represents the domain, the second %s represents the user name, and the third %s represents the password)	Sets the credentials for use in subsequent **Connect To** commands. Use the literal string "Null" to specify a null password.

Selecting an Operation Target

Several Ntdsutil operations require you to identify a particular site, server, or domain by its object in the Configuration container. Instead of requiring that you enter the full distinguished names of objects, many of these operations provide a numbered list of valid selections.

The **Select operation target** submenu appears as a selection within several other menus in Ntdsutil and provides a way to query an existing and operating domain controller about significant objects in the Configuration container. The general model is to connect to a server and list the objects that it knows about. The objects are displayed to the console and numbered from zero onward. You select a particular object by entering its number instead of its distinguished name. Table C.3 lists and describes the select operation target commands.

Table C.3 Select Operation Target Commands

Command	Description
Connections	Invokes the **Connections** submenu.
List current selections	Lists the currently selected site, domain, and server.
List domains	Lists all domains that have a corresponding cross-reference object in the partitions container. Note that some of these domains might not exist if the last domain controller for a domain was removed without performing the demotion properly.
List domains in site	Lists the domains that have domain controllers in the currently selected site.
List roles for connected server	Lists all the operations master roles that the server to which you are connected knows about and displays the domain controllers that are the current operations master role owners. Due to replication latency, the server to which you are connected might not have an up-to-date view of the current role owners.
List servers for domain in site	Lists all the servers known for the currently selected site and domain.
List servers in site	Lists all the domain controllers known to be in the currently selected site.
List sites	Lists all the sites in the forest.
Select domain %d	Selects the domain specified by **%d**.
Select server %d	Selects the server specified by **%d**.
Select site %d	Selects the site specified by **%d**.

Managing Operations Master Roles

Although Active Directory is based on a multimaster administration model, some operations support only a single master. For multimaster operations, conflict resolution ensures that after the system finishes replicating, all replicas agree on the value for a given property on a given object. However, some data, for which adequate conflict resolution is not possible, is key to the operation of the system as a whole. This data is controlled by individual domain controllers called operations masters. These domain controllers are referred to as holding a particular operations master role.

Note Operations masters are sometimes referred to as Flexible Single-Master Operations (FSMOs).

There are five operations master roles: some are enterprisewide, and some are per domain. The following paragraphs describe these five roles:

Schema Operations Master There is a single schema operations master role for the entire enterprise. This role allows the operations master server to accept schema updates. There are other restrictions on schema updates. For more information about operations masters, see "Managing Flexible Single-Master Operations" in this book.

Relative ID Master There is one relative ID master per domain. Each domain controller in a domain has the ability to create security principals. Each security principal is assigned a relative ID. Each domain controller is allocated a small set of relative IDs out of a domainwide relative ID pool. The relative ID master role allows the domain controller to allocate new subpools out of the domainwide relative ID pool.

Domain-Naming Master There is a single domain-naming master role for the entire enterprise. The domain-naming master role allows the owner to define new cross-reference objects representing domains in the Partitions container.

PDC Operations Master There is one primary domain controller (PDC) operations master role per domain. The owner of the PDC operations master role identifies which domain controller in a domain performs Microsoft® Windows NT® version 4.0 PDC activities in support of Windows NT 4.0 backup domain controllers and clients using earlier versions of Windows.

Infrastructure Master There is one infrastructure master role per domain. The owner of this role ensures the referential integrity of objects with attributes that contain distinguished names of other objects that might exist in other domains. Because Active Directory allows objects to be moved or renamed, the infrastructure master periodically checks for object modifications and maintains the referential integrity of these objects.

For more information about operations masters and operations master roles, see "Managing Flexible Single-Master Operations" in this book.

An operations master role can only be moved by administrative involvement; it is not moved automatically. Additionally, moving a role is controlled by standard Windows 2000 access controls. Thus a corporation should tightly control the location and movement of operations master roles. For example, an organization with a strong IT presence might place the schema role on a server in the IT group and configure its access control list (ACL) so that it cannot be moved at all.

Operations master roles require two forms of management: controlled transfer and seizure.

Use controlled transfer when you want to move a role from one server to another, perhaps to track a policy change with respect to role location or in anticipation of a server being shut down, moved, or decommissioned.

Seizure is required when a server that is holding a role fails and you do not intend to restore it. Even in the case of a server recovered from a backup, the server does not assume that it owns a role (even if the backup tape says so), because the server cannot determine if the role was legitimately transferred to another server in the time period between when the backup was made and the server failed and was recovered. The restored server assumes role ownership only if a quorum of existing servers is available during recovery and they all agree that the restored server is still the owner.

The **Roles** submenu in Ntdsutil is used to perform controlled transfer and recovery of operations master roles. Controlled transfer is simple and safe. Because the source and destination servers are running, the system software guarantees that the operations master role token and its associated data is transferred atomically. Operations master role seizure is equally simple but not as safe. You simply tell a particular domain controller that it is now the owner of a particular role.

Caution Do not make a server a role owner by means of seizure commands if the real role holder exists on the network. Doing this could create irreconcilable conflicts for key system data. If an operations master role owner is temporarily unavailable, do not make another domain controller the role owner. This could result in a situation where two computers function as the role owner, which might cause irreconcilable conflicts for key system data.

The commands listed in Table C.4 are found in the **Roles** submenu and perform controlled transfer and recovery of operations master roles.

Table C.4 Roles Commands

Command	Description
Abandon all roles	Instructs the domain controller to which you are connected to give away all operations master roles it owns. This command is not guaranteed to succeed because eligible role recipients might be currently unreachable or because the domain controller to which you are connected is the last domain controller for the domain.
Connections	Invokes the **Connections** submenu.
Seize domain naming master	Forces the domain controller to which you are connected to claim ownership of the domain-naming operations master role without regard to the data associated with the role. Use only for recovery purposes.
Seize infrastructure master	Forces the domain controller to which you are connected to claim ownership of the infrastructure operations master role without regard to the data associated with the role. Use only for recovery purposes.

(continued)

Table C.4 Roles Commands ***(continued)***

Command	Description
Seize PDC	Forces the domain controller to which you are connected to claim ownership of the PDC operations master role without regard to the data associated with the role. Use only for recovery purposes.
Seize RID master	Forces the domain controller to which you are connected to claim ownership of the relative ID master role without regard to the data associated with the role. Use only for recovery purposes.
Seize schema master	Forces the domain controller to which you are connected to claim ownership of the schema operations master role without regard to the data associated with the role. Use only for recovery purposes.
Select operation target	Invokes the **Select operation target** submenu.
Transfer domain naming master	Instructs the domain controller to which you are connected to obtain the domain-naming role by means of controlled transfer.
Transfer infrastructure master	Instructs the domain controller to which you are connected to obtain the infrastructure operations master role by means of controlled transfer.
Transfer PDC	Instructs the domain controller to which you are connected to obtain the PDC operations master by means of controlled transfer.
Transfer RID master	Instructs the domain controller to which you are connected to obtain the relative ID master role by means of controlled transfer.
Transfer schema master	Instructs the domain controller to which you are connected to obtain the schema operations master role by means of controlled transfer.

Managing Orphaned Metadata

The directory service maintains various metadata for each domain and server known to the forest. Normally, domains and domain controllers are created by means of promotion using the Active Directory Installation wizard provided with the Windows 2000 operating system and are removed by means of demotion using the same tool. You can invoke the Active Directory Installation wizard by typing **dcpromo** at the command prompt or by selecting the **Active Directory** option that is displayed on the welcome page of Configure Your Server.

Promotion and demotion are designed to correctly clean up the appropriate metadata. In the directory, however, you might have domain controllers that were decommissioned incorrectly. In this case, their metadata is not cleaned up. For example, a domain controller has failed, and rather than attempting to restore it, you decide to retire the server. This leaves some information about the retired domain controller in the directory. The general model of operation is to connect to a server known to have a copy of the offending metadata, select an operation target, and then delete it.

Caution Do not delete the metadata of existing domains and domain controllers.

Table C.5 lists and describes the metadata cleanup commands.

Table C.5 Metadata Cleanup Commands

Command	Description
Connections	Invokes the **Connections** submenu.
Remove selected domain	Removes the metadata associated with the domain selected in the **Select operation target** submenu.
Remove selected server	Removes the metadata associated with the domain controller selected in the **Select operation target** submenu.
Select operation target	Invokes the **Select operation target** submenu.

Performing an Authoritative Restore

When a domain contains more than one domain controller, Active Directory replicates directory objects, such as users, groups, organizational units, and computers, to all the domain controllers in that domain.

When you are restoring a domain controller by using backup and restore programs, such as Ntbackup or those from third-party providers, the default mode for the restore is nonauthoritative. This means that the restored server is brought up-to-date with its replicas through the normal replication mechanism. For example, if a domain controller is restored from a backup tape that is two weeks old, when you restart it, the normal replication mechanism brings it up-to-date with respect to its replication partners.

Authoritative restore allows the administrator to recover a domain controller, restore it to a specific point in time, and mark objects in Active Directory as being authoritative with respect to their replication partners. For example, you might need to perform an authoritative restore if an administrator inadvertently deletes an organizational unit containing a large number of users. If you restore the server from tape, the normal replication process would not restore the inadvertently deleted organizational unit. Authoritative restore allows you to mark the organizational unit as authoritative and force the replication process to restore it to all of the other domain controllers in the domain.

Table C.6 lists and describes the authoritative restore commands.

Table C.6 Authoritative Restore Commands

Command	Description
Restore database	Marks the entire Ntds.dit (both the domain and configuration naming contexts held by the domain controller) as authoritative. The schema cannot be authoritatively restored.
Restore database verinc %d	Marks the entire Ntds.dit (both the domain and configuration naming contexts held by the domain controller) as authoritative and increments the version number by **%d**. Use this option only to authoritatively restore over a previous, incorrect, authoritative restore, such as an authoritative restore from a backup that contains the problem you want to restore over.

(continued)

Table C.6 Authoritative Restore Commands *(continued)*

Command	Description
Restore subtree %s	Marks subtree (and all children of subtree) as being authoritative. The subtree is defined by using the fully distinguished name of the object.
Restore subtree %s verinc %d	Marks subtree (and all children of subtree) as being authoritative and increments the version number by **%d**. The subtree is defined by using the fully distinguished name of the object. Use this option only to authoritatively restore over a previous, incorrect, authoritative restore, such as an authoritative restore from a backup that contains the problem you want to restore over.

Managing Domains

Ordinarily a user must belong to the Enterprise Administrators group to create child domains and to promote servers as domain controllers. Often the staff member who installs the hardware and software on domain controllers is not the same person who requires high levels of administrative privilege. The domain management commands allow administrators who are members of the Enterprise Administrators group to precreate cross-reference and server objects in the directory.

Table C.7 lists and describes the domain management commands used by Ntdsutil.

Table C.7 Domain Management Commands

Command	Description
List	Lists all the naming contexts that exist in the enterprise, the schema and configuration naming contexts, as well as all domain naming contexts.
Precreate %s1 %s2	Creates a cross reference object for the domain **%s1** allowing a server named **%s2** to be promoted as the domain controller for that domain. The domain name must be specified by using a fully distinguished name, and the server must be named by using the fully qualified DNS name.

Managing Lightweight Directory Access Protocol Policies

To ensure that domain controllers can support service level guarantees, you need to specify operational limits for a number of Lightweight Directory Access Protocol (LDAP) operations. These limits prevent specific operations from adversely impacting the performance of the server and also make the server resilient to denial of service attacks.

LDAP policies are implemented by using objects of the class queryPolicy. Query Policy objects can be created in the container Query Policies, which is a child of the Directory Service container in the configuration naming context. For example: cn=Query-Policies, cn=Directory Service, cn=Windows NT, cn=Services *<configuration naming context>*.

A domain controller uses the following three mechanisms to apply LDAP policies:

- A domain controller might refer to a specific LDAP policy. The nTDSASettings object includes an optional attribute queryPolicyObject, which contains the distinguished name of a Query Policy.
- In the absence of a specific query policy being applied to a domain controller, the domain controller applies the Query Policy that has been assigned to the domain controller's site. The ntDSSiteSettings object includes an optional attribute queryPolicyObject, which contains the distinguished name of a Query Policy.
- In the absence of a specific domain controller or site Query Policy, a domain controller uses the default query policy named Default-Query Policy.

A Query Policy object includes the multivalued attributes LDAPIPDenyList and LDAPAdminLimits. Ntdsutil allows the administrator to set the LDAP administration limits and IP Deny list for the Default-Query Policy object.

The LDAP administration limits (with defaults in parentheses) are the following:

InitRecvTimeout Initial receive time-out (120 seconds).

MaxConnections Maximum number of open connections (5000).

MaxConnIdleTime Maximum amount of time a connection can be idle (900 seconds).

MaxActiveQueries Maximum number of queries that can be active at one time (20).

MaxNotificationPerConnection Maximum number of notifications that a client can request for a given connection (5).

MaxPageSize Maximum page size supported for LDAP responses (1000 records).

MaxQueryDuration Maximum length of time the domain controller can execute a query (120 seconds).

MaxTempTableSize Maximum size of temporary storage allocated to execute queries (10,000 records).

MaxResultSetSize Maximum size of the LDAP Result Set (262144 bytes).

MaxPoolThreads Maximum number of threads created by the domain controller for query execution (4 per processor).

MaxDatagramRecv Maximum number of datagrams that can be processed by the domain controller simultaneously (1024).

Table C.8 lists and describes the LDAP policies commands.

Table C.8 LDAP Policies Commands

Command	Description
Cancel	Cancels any uncommitted modifications of the LDAP administration limits to the default query policy.
Commit	Commits all modifications of the LDAP administration limits to the default query policy.
List	Lists all supported LDAP administration limits for the domain controller.
Set %s1 to %s2	Sets the value of the LDAP administration limit **%s1** to the value **%s2**.
Show values	Shows the current and proposed values for the LDAP administration limits.

Managing the IP Deny List

To provide higher levels of security for the domain controller, you can apply an IP Deny List that prevents the domain controller from accepting LDAP queries from clients with specified IP addresses. Similar to the LDAP administration limits, the IP Deny List only alters the Default LDAP Policy object. The default LDAP Policy is applied to any domain controller that has not had a specific LDAP policy applied to it or to the site in which it belongs.

Table C.9 lists and describes the Ntdsutil menu commands on the IP Deny List.

Table C.9 Ntdsutil IP Deny List Menu Commands

Command	Description
Add %s1 %s2	Adds an entry to the IP Deny List. The first parameter **%s1** is either the host component or network component of an IP address. If a host component is specified, the second parameter **%s2** is specified as NODE; whereas if the network component is specified, the second parameter is the subnet mask. For example, to deny access from a host with an address of 192.168.100.10, the command is: **Add 192.168.100.10 NODE** To deny access from all hosts with a network address of 192.168.100.0, the command is: **Add 192.168.100.0 255.255.255.0** The entries that you specify by using the add command are not applied until you commit them by using the **Commit** command.
Cancel	Cancels any uncommitted additions or deletions.
Commit	Commits all additions or deletions to the LDAP policy object.
Delete %d	Deletes the specified entry with the index number **%d**. Use the show command to display entries with the respective index number.
Show	Shows all IP addresses that are included in the IP Deny List.
Test %s	Determines whether the IP address specified by **%s** is allowed or denied access to the domain controller. For example, given an IP Deny List entry of 192.168.100.0 255.255.255.0, when tested with an address of 192.168.100.10, access is denied.

Managing Security Accounts

Each security account (users, groups, and computers) is identified by a unique security identifier (SID). Use a SID to uniquely identify a security account and to perform access checks against resources, such as files, file directories, printers, Exchange mailboxes, Microsoft® SQL Server™ databases, objects stored in Active Directory, or any data that is protected by the Windows 2000 security model.

A SID is made up of header information and a set of relative identifiers that identify the domain and the security account. Within a domain, each domain controller is capable of creating accounts and issuing each account a unique security identifier. Each domain controller maintains a pool of relative IDs that is used in the creation of security identifiers. When 80 percent of the relative ID pool is consumed, the domain controller requests a new pool of relative identifiers from the relative ID operations master. This ensures that the same pool of relative IDs is never allocated to different domain controllers and prevents the allocation of duplicate security identifiers. However, because it is possible (but rare) for a duplicate relative ID pool to be allocated, you need to identify those accounts that have been issued duplicate security identifiers so that you prevent undesirable application of security.

One cause of duplicate relative ID pools is when the administrator seizes the relative ID master role while the original relative ID master is operational but temporarily disconnected from the network. In normal practice, after one replication cycle, the relative ID master role is assumed by just one domain controller, but it is possible that before the role ownership is resolved, two different domain controllers might each request a new relative ID pool and be allocated the same relative ID pool.

Table C.10 lists and describes the menu commands for security account management.

Table C.10 Security Account Management Menu Commands

Command	Description
Check Duplicate SID	Checks the domain for any objects that have duplicate security identifiers.
Cleanup Duplicate SID	Deletes all objects that have duplicate security identifiers and logs these entries into the log file.
Log File %s	Sets the log file to **%s**. If a log file is not explicitly set, the log file defaults to Dupsid.log.

Using Semantics Database Analysis

Unlike the file management commands described earlier, which test the integrity of the database with respect to the ESENT database semantics, the semantic analysis analyzes the data with respect to Active Directory semantics. It generates reports on the number of records present, including deleted and phantom records.

Note End users should not use this command except when Microsoft requests them to use it as an aid to fault diagnosis.

Table C.11 lists and describes the menu commands for semantic database analysis.

Table C.11 Semantic Database Analysis Menu Commands

Command	Description
Get %d	Retrieves record number %d from the Ntds.dit.
Go	Starts the semantic analysis of the Ntds.dit. A report is generated and written to a file named Dsdit.dmp.*n*, in the current directory, where *n* is an integer incremented each time that you carry out the command.
Verbose %s	Toggles verbose mode on or off.

List of Menu Commands

Table C.12 lists the main menu and submenu commands. The **help**, **quit** and **?** commands are not included.

Note Menu commands preceded by an asterisk are functional only when the domain controller is operating in Directory Services Restore Mode.

Table C.12 Menu Commands

Main Menu Command	Submenu Options
* Authoritative restore	Restore database Restore database verinc %d Restore subtree %s Restore subtree %s verinc %d
Domain management	Connections List Precreate %s %s Select operation target
* Files	Compact to %s Header Info Integrity Move DB to %s Move logs to %s Recover Repair Set path backup %s Set path DB %s Set path logs %s Set path working dir %s
IPDeny List	Add %s %s Cancel Commit Connections Delete %d Show Test %s

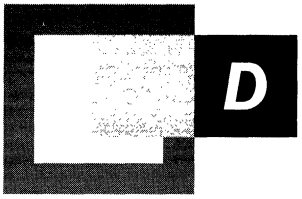
(continued)

Table C.12 Menu Commands ***(continued)***

Main Menu Command	Submenu Options
LDAP policies	Cancel Changes Commit Changes Connections List Set %s to %s Show Values
Metadata cleanup	Connections Remove selected domain Remove selected Naming Context Remove selected server Select operation target
Popups %s	
Roles	Connections Seize domain naming master Seize infrastructure master Seize PDC Seize RID master Seize schema master Select operation target Transfer domain naming master Transfer infrastructure master Transfer PDC Transfer RID master Transfer schema master
Security account management	Check Duplicate SID Cleanup Duplicate SID Connect to server %s Log File %s
* Semantic database analysis	Get %d Go Go Fixup Verbose %s

APPENDIX D

User Rights



User rights fall into two general categories: logon rights and privileges. Logon rights control who is authorized to log on to a computer and how they can log on. Privileges control access to system resources, and they can override the permissions that are set on a particular object on the computer.

The special account LocalSystem has built-in capabilities that correspond to almost all privileges and logon rights. Processes that are running as part of the operating system are associated with this account, and they require a complete set of user rights. The system services that are supplied with Microsoft® Windows® 2000 are configured automatically to run as LocalSystem. Although you can configure other services to also run under this account, it is recommended that you do so with care.

In This Appendix

Logon Rights

Logon rights control how security principals are allowed access to the computer—whether from the keyboard or through a network connection, or whether as a service or as a batch job. For each logon method, there exists a pair of logon rights—one to allow logging on to the computer and another to deny logging on to the computer. Use a deny logon right as you would use a deny permission—to exclude a subset of a group that has been assigned an allow logon right. For example, suppose that Alice wants all users except the members of the domain's Marketing group to be able to log on locally at her computer's keyboard. With this in mind, Alice creates a local group, which she names "LocalLogonDenied." Then she configures her computer as follows:

1. She assigns the log on locally user right to the Users group.
2. She assigns the deny local logon user right to the LocalLogonDenied group.
3. She makes the Marketing group a member of the LocalLogonDenied group.

Deny rights take precedence over allow rights, so members of the Marketing group are denied the right to log on locally even though they are also members of the Users group, which is allowed to log on locally.

The rule to keep in mind is: "Allow a set, and then deny a subset." Reversing the order can be disastrous. For example, Alice might want to allow no one but herself to log on locally. If she allowed herself the right to log on locally and denied the Users group the right to log on locally, she would be unpleasantly surprised to find she had locked herself out of the computer. Alice, after all, is a member of the Users group, so the deny right she assigned to the Users group would take precedence over the allow right she assigned to herself.

Logon rights are described in Table D.1. For developers, who are more familiar with the constants in Winnt.h than with the logon right display names in the Security Settings user interface, the Winnt.h string that corresponds to each user logon right is shown in parentheses after the display name.

Table D.1 Logon Rights

Right	Description
Access this computer from network (SeNetworkLogonRight)	Allows a user to connect to the computer from the network. By default, this right is assigned to Administrators, Everyone, and Power Users.
Log on as a batch job (SeBatchLogonRight)	Allows a user to log on by using a batch-queue facility. By default, this right is assigned to Administrators.
Log on locally (SeInteractiveLogonRight)	Allows a user to log on locally at the computer's keyboard. By default, this right is assigned to Administrators, Account Operators, Backup Operators, Print Operators, and Server Operators.
Log on as a service (SeServiceLogonRight)	Allows a security principal to log on as a service. Services can be configured to run under the LocalSystem account, which has a built-in right to log on as a service. Any service that runs under a separate account must be assigned the right. By default, this right is not assigned to anyone.
Deny access to this computer from network (SeDenyNetworkLogonRight)	Prohibits a user or group from connecting to the computer from the network. By default, no one is denied this right.
Deny local logon (SeDenyInteractiveLogonRight)	Prohibits a user or group from logging on locally at the keyboard. By default, no one is denied this right.
Deny logon as a batch job (SeDenyBatchLogonRight)	Prohibits a user or group from logging on through a batch-queue facility. By default, no one is denied the right to log on as a batch job.
Deny logon as a service (SeDenyServiceLogonRight)	Prohibits a user or group from logging on as a service. By default, no one is denied the right to log on as a service.

Privileges

To ease the task of account administration, it is recommended that you assign privileges primarily to groups rather than to individual user accounts. When you assign privileges to a group, the privileges are assigned automatically to each user who is added to the group. This is easier than assigning privileges to individual user accounts as each account is created.

The privileges that can be assigned are listed and described in Table D.2. The strings that correspond to the constants in Winnt.h are shown in parentheses.

Table D.2 Privileges

Privilege	Description
Act as part of the operating system (SeTcbPrivilege)	Allows a process to authenticate like a user and thus gain access to the same resources as a user. Only low-level authentication services should require this privilege. Note that potential access is not limited to what is associated with the user by default; the calling process might request that arbitrary additional privileges be added to the access token. Note that the calling process can also build an anonymous token that does not provide a primary identity for tracking events in the audit log. When a service requires this privilege, configure the service to use the LocalSystem account (which already includes the privilege), rather than create a separate account and assign the privilege to it.
Add workstations to a domain (SeMachineAccountPrivilege)	Allows the user to add a computer to a specific domain. For the privilege to be effective, it must be assigned to the user as part of local security policy for domain controllers in the domain. A user who has this privilege can add up to 10 workstations to the domain. In Windows 2000, the behavior of this privilege is duplicated by the Create Computer Objects permission for organizational units and the default Computers container in Active Directory™. Users who have the Create Computer Objects permission can add an unlimited number of computers to the domain. *(continued)*

Table D.2 Privileges *(continued)*

Privilege	Description
Back up files and directories (SeBackupPrivilege)	Allows the user to circumvent file and directory permissions to back up the system. The privilege is selected only when an application attempts access through the NTFS backup application programming interface (API). Otherwise, normal file and directory permissions apply. By default, this privilege is assigned to Administrators and Backup Operators. (See also "Restore files and directories" in this table.)
Bypass traverse checking (SeChangeNotifyPrivilege)	Allows the user to pass through folders to which the user otherwise has no access while navigating an object path in any Microsoft® Windows® file system or in the registry. This privilege does not allow the user to list the contents of a folder; it allows the user only to traverse its directories. By default, this privilege is assigned to Administrators, Backup Operators, Power Users, Users, and Everyone.
Change the system time (SeSystemTimePrivilege)	Allows the user to set the time for the internal clock of the computer. By default, this privilege is assigned to Administrators and Power Users.
Create a token object (SeCreateTokenPrivilege)	Allows a process to create an access token by calling NtCreateToken( ) or other token-creating APIs. When a process requires this privilege, use the LocalSystem account (which already includes the privilege), rather than create a separate user account and assign this privilege to it.
Create permanent shared objects (SeCreatePermanentPrivilege)	Allows a process to create a directory object in the Windows 2000 object manager. This privilege is useful to kernel-mode components that extend the Windows 2000 object namespace. Components that are running in kernel mode already have this privilege assigned to them; it is not necessary to assign them the privilege.
Create a pagefile (SeCreatePagefilePrivilege)	Allows the user to create and change the size of a pagefile. This is done by specifying a paging file size for a particular drive under **Performance Options** on the **Advanced** tab of **System Properties**. By default, this privilege is assigned to Administrators.

(continued)

Table D.2 Privileges *(continued)*

Privilege	Description
Debug programs (SeDebugPrivilege)	Allows the user to attach a debugger to any process. This privilege provides access to sensitive and critical operating system components. By default, this privilege is assigned to Administrators.
Enable computer and user accounts to be trusted for delegation (SeEnableDelegationPrivilege)	Allows the user to change the **Trusted for Delegation** setting on a user or computer object in Active Directory. The user or computer that is granted this privilege must also have write access to the account control flags on the object. Delegation of authentication is a capability that is used by multi-tier client/server applications. It allows a front-end service to use the credentials of a client in authenticating to a back-end service. For this to be possible, both client and server must be running under accounts that are trusted for delegation. Misuse of this privilege or the **Trusted for Delegation** settings can make the network vulnerable to sophisticated attacks on the system that use Trojan horse programs, which impersonate incoming clients and use their credentials to gain access to network resources.
Force shutdown from a remote system (SeRemoteShutdownPrivilege)	Allows a user to shut down a computer from a remote location on the network. (See also "Shut down the system" in this table.) By default, this privilege is assigned to Administrators.
Generate security audits (SeAuditPrivilege)	Allows a process to generate entries in the security log. The security log is used to trace unauthorized system access. (See also "Manage auditing and security log" in this table.)
Increase quotas (SeIncreaseQuotaPrivilege)	Allows a process that has Write Property access to another process to increase the processor quota that is assigned to the other process. This privilege is useful for system tuning, but it can be abused, as in a denial-of-service attack. By default, this privilege is assigned to Administrators.

(continued)

Table D.2 Privileges *(continued)*

Privilege	Description
Increase scheduling priority (SeIncreaseBasePriorityPrivilege)	Allows a process that has Write Property access to another process to increase the execution priority of the other process. A user with this privilege can change the scheduling priority of a process in the **Task Manager** dialog box. By default, this privilege is assigned to Administrators.
Load and unload device drivers (SeLoadDriverPrivilege)	Allows a user to install and uninstall Plug and Play device drivers. This privilege does not apply to device drivers that are not Plug and Play; these device drivers can be installed only by Administrators. Note that device drivers run as trusted (highly privileged) programs; a user can abuse this privilege by installing hostile programs and giving them destructive access to resources. By default, this privilege is assigned to Administrators.
Lock pages in memory (SeLockMemoryPrivilege)	Allows a process to keep data in physical memory, which prevents the system from paging the data to virtual memory on disk. Assigning this privilege can result in significant degradation of system performance. This privilege is obsolete and is therefore never selected.
Manage auditing and security log (SeSecurityPrivilege)	Allows a user to specify object access auditing options for individual resources such as files, Active Directory objects, and registry keys. Object access auditing is not actually performed unless you have enabled it in **Audit Policy** (under **Security Settings**, **Local Policies**). A user who has this privilege also can view and clear the security log from Event Viewer. By default, this privilege is assigned to Administrators.
Modify firmware environment values (SeSystemEnvironmentPrivilege)	Allows modification of system environment variables either by a process through an API or by a user through **System Properties**. By default, this privilege is assigned to Administrators.

(continued)

Table D.2 Privileges *(continued)*

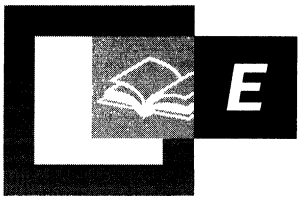
Privilege	Description
Profile a single process (SeProfileSingleProcessPrivilege)	Allows a user to run Microsoft® Windows NT® and Windows 2000 performance-monitoring tools to monitor the performance of nonsystem processes. By default, this privilege is assigned to Administrators and Power Users.
Profile system performance (SeSystemProfilePrivilege)	Allows a user to run Windows NT and Windows 2000 performance-monitoring tools to monitor the performance of system processes. By default, this privilege is assigned to Administrators.
Remove computer from docking station (SeUndockPrivilege)	Allows the user of a portable computer to undock the computer by clicking **Eject PC** on the **Start** menu. By default, this privilege is assigned to Administrators, Power Users, and Users.
Replace a process-level token (SeAssignPrimaryTokenPrivilege)	Allows a parent process to replace the access token that is associated with a child process.
Restore files and directories (SeRestorePrivilege)	Allows a user to circumvent file and directory permissions when restoring backed-up files and directories and to set any valid security principal as the owner of an object. (See also "Back up files and directories" in this table.) By default, this privilege is assigned to Administrators and Backup Operators.
Shut down the system (SeShutdownPrivilege)	Allows a user to shut down the local computer. (See also "Force shutdown from a remote system" in this table.) In Microsoft® Windows® 2000 Professional, this privilege is assigned by default to Administrators, Backup Operators, Power Users, and Users. In Microsoft® Windows® 2000 Server, this privilege is by default not assigned to Users; it is assigned only to Administrators, Backup Operators, and Power Users. *(continued)*

Table D.2 Privileges ***(continued)***

Privilege	Description
Synchronize directory service data (SeSynchAgentPrivilege)	Allows a process to provide directory synchronization services. This privilege is relevant only on domain controllers. By default, this privilege is assigned to the Administrator and LocalSystem accounts on domain controllers.
Take ownership of files or other objects (SeTakeOwnershipPrivilege)	Allows a user to take ownership of any securable object in the system, including Active Directory objects, files and folders, printers, registry keys, processes, and threads. By default, this privilege is assigned to Administrators.

APPENDIX E

Well-Known Security Identifiers



A security identifier (SID) is a unique value of variable length that is used to identify a security principal or security group in Microsoft® Windows® 2000 and Microsoft® Windows NT®. *Well-known SIDs* are a group of SIDs that identify generic users or generic groups. Their values remain constant. Table E.1 is a list of all SIDs in this category.

For information about how SIDs are assigned and used, see "Access Control" in this book.

Table E.1 Well-Known Security Identifiers

SID	Name	Description
S-1-0	Null Authority	An identifier authority.
S-1-0-0	Nobody	No security principal.
S-1-1	World Authority	An identifier authority.
S-1-1-0	Everyone	A group that includes all users, even anonymous users and guests. Membership is controlled by the operating system.
S-1-2	Local Authority	An identifier authority.
S-1-3	Creator Authority	An identifier authority.
S-1-3-0	Creator Owner	A placeholder in an inheritable access control entry (ACE). When the ACE is inherited, the system replaces this SID with the SID for the object's current owner.
S-1-3-1	Creator Group	A placeholder in an inheritable ACE. When the ACE is inherited, the system replaces this SID with the SID for the primary group of the object's current owner. The primary group is used only by the POSIX subsystem.

(continued)

Table E.1 Well-Known Security Identifiers *(continued)*

SID	Name	Description
S-1-3-2	Creator Owner Server	[SID not used in Windows 2000.]
S-1-3-3	Creator Group Server	[SID not used in Windows 2000.]
S-1-4	Nonunique Authority	An identifier authority.
S-1-5	NT Authority	An identifier authority.
S-1-5-1	Dialup	A group that implicitly includes all users who are logged on to the system through a dial-up connection. Membership is controlled by the operating system.
S-1-5-2	Network	A group that implicitly includes all users who are logged on through a network connection. Membership is controlled by the operating system.
S-1-5-3	Batch	A group that implicitly includes all users who have logged on through a batch queue facility such as task scheduler jobs. Membership is controlled by the operating system.
S-1-5-4	Interactive	A group that includes all users who have logged on interactively. Membership is controlled by the operating system.
S-1-5-5-*X*-*Y*	Logon Session	A logon session. The *X* and *Y* values for these SIDs uniquely identify a particular logon session.
S-1-5-6	Service	A group that includes all security principals that have logged on as a service. Membership is controlled by the operating system.
S-1-5-7	Anonymous	A user who has logged on anonymously.
S-1-5-8	Proxy	[SID not used in Windows 2000.]

(continued)

Table E.1 Well-Known Security Identifiers *(continued)*

SID	Name	Description
S-1-5-9	Enterprise Controllers	A group that includes all domain controllers an Active Directory™ directory service forest of domains. Membership is controlled by the operating system.
S-1-5-10	Principal Self (or Self)	A placeholder in an ACE on a user, group, or computer object in Active Directory. When you grant permissions to Principal Self, you grant them to the security principal represented by the object. During an access check, the operating system replaces the SID for Principal Self with the SID for the security principal represented by the object.
S-1-5-11	Authenticated Users	A group that includes all users whose identities were authenticated when they logged on. Membership is controlled by the operating system.
S-1-5-12	Restricted Code	[SID reserved for future use.]
S-1-5-13	Terminal Server Users	A group that includes all users who have logged on to a Terminal Services server. Membership is controlled by the operating system.
S-1-5-18	Local System	A service account that is used by the operating system.
S-1-5-<*domain*>-500	Administrator	A user account for the system administrator. This account is the first account created during operating system installation. The account cannot be deleted or locked out. It is a member of the Administrators group and cannot be removed from that group.
S-1-5-<*domain*>-501	Guest	A user account for people who do not have individual accounts. This user account does not require a password. By default, the Guest account is disabled.

(continued)

Table E.1 Well-Known Security Identifiers *(continued)*

SID	Name	Description
S-1-5-<*domain*>-502	KRBTGT	A service account that is used by the Key Distribution Center (KDC) service.
S-1-5-<*domain*>-512	Domain Admins	A global group whose members are authorized to administer the domain. By default, the Domain Admins group is a member of the Administrators group on all computers that have joined a domain, including the domain contollers. Domain Admins is the default owner of any object that is created in the domain's Active Directory by any member of the group. If members of the group create other objects, such as files, the default owner is the Administrators group.
S-1-5-<*domain*>-513	Domain Users	A global group that, by default, includes all user accounts in a domain. When you create a user account in a domain, it is added to this group automatically.
S-1-5-<*domain*>-514	Domain Guests	A global group that, by default, has only one member, the domain's built-in Guest account.
S-1-5-<*domain*>-515	Domain Computers	A global group that includes all computers that have joined the domain, excluding domain controllers.
S-1-5-<*domain*>-516	Domain Controllers	A global group that includes all domain controllers in the domain. New domain controllers are added to this group automatically.

(continued)

Table E.1 Well-Known Security Identifiers *(continued)*

SID	Name	Description
S-1-5-<*domain*>-517	Cert Publishers	A global group that includes all computers that are running an enterprise certificate authority. Cert Publishers are authorized to publish certificates for User objects in Active Directory.
S-1-5-<*root domain*>-518	Schema Admins	A group that exists only in the root domain of an Active Directory forest of domains. It is a universal group if the domain is in native-mode, a global group if the domain is in mixed-mode. The group is authorized to make schema changes in Active Directory. By default, the only member of the group is the Administrator account for the forest root domain.
S-1-5-<*root domain*>-519	Enterprise Admins	A group that exists only in the root domain of an Active Directory forest of domains. It is a universal group if the domain is in native-mode, a global group if the domain is in mixed-mode. The group is authorized to make forest-wide changes in Active Directory, such as adding child domains. By default, the only member of the group is the Administrator account for the forest root domain.

(continued)

Table E.1 Well-Known Security Identifiers *(continued)*

SID	Name	Description
S-1-5-<*domain*>-520	Group Policy Creators Owners	A global group that is authorized to create new Group Policy objects in Active Directory. By default, the only member of the group is Administrator. The default owner of a new Group Policy object is usually the user who created it. If the user is a member of Administrators or Domain Admins, all objects that are created by the user are owned by the group. Owners have full control of the objects they own.
S-1-5-<*domain*>-553	RAS and IAS Servers	A domain local group. By default, this group has no members. Computers that are running the Routing and Remote Access service are added to the group automatically. Members of this group have access to certain properties of User objects, such as Read Account Restrictions, Read Logon Information, and Read Remote Access Information.
S-1-5-32-544	Administrators	A built-in group. After the initial installation of the operating system, the only member of the group is the Administrator account. When a computer joins a domain, the Domain Admins group is added to the Administrators group. When a server becomes a domain controller, the Enterprise Admins group also is added to the Administrators group. The Administrators group has built-in capabilties that give its members full control over the system. The group is the default owner of any object that is created by a member of the group.

(continued)

Table E.1 Well-Known Security Identifiers *(continued)*

SID	Name	Description
S-1-5-32-545	Users	A built-in group. After the initial installation of the operating system, the only member is the Authenticated Users group. When a computer joins a domain, the Domain Users group is added to the Users group on the computer. Users can perform tasks such as running applications, using local and network printers, shutting down the computer, and locking the computer. Users can install applications that only they are allowed to use if the installation program of the application supports per-user installation.
S-1-5-32-546	Guests	A built-in group. By default, the only member is the Guest account. The Guests group allows occasional or one-time users to log on with limited privileges to a computer's built-in Guest account.
S-1-5-32-547	Power Users	A built-in group. By default, the group has no members. This group does not exist on domain controllers. Power Users can create local users and groups; modify and delete accounts that they have created; and remove users from the Power Users, Users, and Guests groups. Power Users also can install most applications; create, manage, and delete local printers; and create and delete file shares.

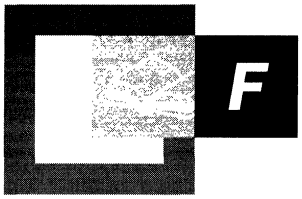
(continued)

Table E.1 Well-Known Security Identifiers *(continued)*

SID	Name	Description
S-1-5-32-548	Account Operators	A built-in group that exists only on domain controllers. By default, the group has no members. By default, Account Operators have permission to create, modify, and delete accounts for users, groups, and computers in all containers and organizational units (OUs) of Active Directory except the Builtin container and the Domain Controllers OU. Account Operators do not have permission to modify the Administrators and Domain Admins groups, nor do they have permission to modify the accounts for members of those groups.
S-1-5-32-549	Server Operators	A built-in group that exists only on domain controllers. By default, the group has no members. Server Operators can log on to a server interactively; create and delete network shares; start and stop services; back up and restore files; format the hard disk of the computer; and shut down the computer.
S-1-5-32-550	Print Operators	A built-in group that exists only on domain controllers. By default, the only member is the Domain Users group. Print Operators can manage printers and document queues.
S-1-5-32-551	Backup Operators	A built-in group. By default, the group has no members. Backup Operators can back up and restore all files on a computer, regardless of the permissions that protect those files. Backup Operators also can log on to the computer and shut it down.
S-1-5-32-552	Replicators	Not used in Windows 2000. In Windows NT domains, it is a built-in group used by the File Replication service on domain controllers.

APPENDIX F

"Certified for Microsoft Windows" Applications



Although Microsoft® Windows® 2000 can work with your existing applications, there are several distinct advantages when you use "Certified for Microsoft Windows" applications. Applications that display the "Certified for Microsoft Windows" logo are designed and independently tested to take advantage of Windows 2000 technologies and features, such as Windows Installer, IntelliMirror™ management technologies, and Active Directory™, which is the directory service included with Windows 2000. Figure F.1 shows the "Certified for Microsoft Windows" logo.

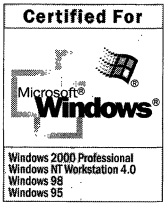


Figure F.1 "Certified for Microsoft Windows" logo

The "Certified for Microsoft Windows" logo identifies software products that have been designed and tested to meet the application specification for Windows 2000. The Application Specification for Windows 2000 was developed by Microsoft in cooperation with customers and third-party programmers to provide clear, concise guidelines to help programmers create reliable and manageable applications. The application specification for Windows 2000 has two versions: a core specification for desktop applications, and a specification for distributed applications. When you are choosing new applications or designing in-house applications, keep these features in mind.

Windows 2000 Desktop Applications

To be certified for Windows operating systems, desktop applications must adhere to the following criteria:

- Provide a robust, self-repairing installation that helps minimize conflicts among shared components such as dynamic-link libraries (DLLs) so that applications can better co-exist.
- Offer improved software deployment and management.
- Correctly maintain user preferences and computer settings, enable roaming users, enable multiple users per computer, and enable regeneration of application settings in computer replacement situations.
- Run in a tightly controlled network environment so that network administrators can secure and control corporate desktops.
- Support OnNow power management, which allows users to start their computers at the touch of a key for improved mobile computing experience.
- Provide a consistent user experience for all users in your organization by supporting accessibility standards to reduce support and training costs.
- Have the ability to upgrade from Microsoft® Windows® 95 or Microsoft® Windows® 98 to Microsoft® Windows® 2000 Professional.

Windows 2000–based Distributed Applications

Windows 2000–based distributed applications provide similar benefits of a desktop application, plus the following:

- Support for a globally available infrastructure by using Active Directory for easier access for the user and easier management for the administrator.
- Secure access to systemwide resources.
- Security for client-server interactions.
- Reduced service downtime caused by system failures or planned server maintenance and upgrades. This is done by exploiting the Cluster service for applications that comply with Windows 2000 Advanced Server requirements.

Certified for Windows 2000 applications provide the following features:

Better Application Reliability Applications that are certified for Windows 2000 are designed to be more dependable, more available, and more manageable because of their reliance on Windows 2000 technologies.

Roaming and Multi-User Support Certified applications separate user and computer settings, which enable roaming and multi-user experiences. This allows user settings to follow the individual user, regardless of the computer that is used on the network. This is important in an environment where users roam, many users share computers, or a single user logs on to several different computers.

Adherence to Group Policy Certified applications check and adhere to important Group Policy settings that you can set to specify the desktop environment for your users. Certified applications do not override these Group Policy settings.

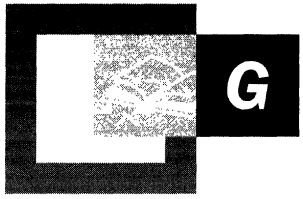
Easier Management Certified applications use Active Directory as the information source for security, policy, and addresses for computers and users. Because certified applications use Active Directory, certified applications provide simplified user and computer settings, and automation of system administrator tasks; so the time that you spend configuring and managing systems is reduced.

Simple Installation and Removal of Applications Certified applications are installed and removed in a simple and predictable way by using Windows Installer. This helps applications interoperate better.

Many independent software vendors have developed products that work on the Windows 2000 operating system. To see a directory of applications that that run on Windows 2000, including applications that are Certified for Windows 2000, see the Directory of Windows 2000 Applications link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

APPENDIX G

OSCML and Client Installation Wizard Variables



The Client Installation Wizard screens can be customized to meet the needs of your organization. You can use these screens to install clients, and you can customize the screens to fit your organization. Entire screens or individual options within screens can be added or removed, and you can modify the text that is displayed to users to include custom information such as technical support phone numbers or corporate policies and procedures. You can build screens that prompt the user for information. This information can be used to control the installation of the image.

Note The Client Installation Wizard uses the Boot Information Negotiation Layer (BINL) service. This service listens for Dynamic Host Configuration Protocol (DHCP) requests and Client Installation Wizard requests. The BINL service directs the client to the files that are required to start the installation process. The BINL service also checks Active Directory to verify credentials; it also determines whether a client needs the remote operating system installation service and whether to create a new computer account object on behalf of the client or reset an existing computer account object.

The default screens that are used during the client logon process and the installation of the operating system are the following:

Welcome.osc. Displays Welcome screen to the user.

Login.osc. Requires the user to log on.

Choice.osc. Displays setup options to the user: **Automatic**, **Custom**, **Restart**, **Maintenance**, and **Tools**. The degree to which this screen and its options are displayed is controlled through Remote Installation Services (RIS) Group Policy settings.

OSAuto.osc. Determines whether a computer account object already exists in Active Directory that has the same GUID as the computer that is running the Client Installation Wizard. If a duplicate GUID is found, DupAuto.osc is displayed. If no duplicate GUID is found, OSChoice.osc is displayed. (This screen is not displayed to users.)

DupAuto.osc. Indicates that a duplicate GUID was found and instructs the user to contact the network administrator.

OSChoice.osc. Displays the list of operating system images that are available to the user who is logged on to the RIS server.

Warning.osc. Warns the user that the drive is going to be formatted.

Install.osc. Displays a summary page to the user.

These screens (sometimes referred to as OSC screens or OSCML) are modeled after Hypertext Markup Language (HTML) version 2.0 specifications. The files for these screens are simple text files that have the file name extension .osc in the format of the OSChooser Markup Language (OSCML). *OSChooser* stands for Operating System Chooser. You can use Notepad to view and modify the tags in these files.

When Remote Installation Setup Wizard (RISetup.exe) is run, a default set of the Client Installation Wizard screens is installed at RemoteInstall\Oschooser\<*language*>.

The \Oschooser portion of the RIS tree contains the .osc files. If multiple languages are installed, there is a subdirectory for each language that the server supports. All of the files that the Client Installation Wizard uses are in this directory, with the exception of Welcome.osc and Multiling.osc.

The Welcome.osc file is the first screen that every user sees, no matter what language the server uses; the screen displays the languages that are supported. When the user connects to the server, he or she can select the appropriate language. Then the Welcome.osc file in the specified *%language%* subdirectory appears in the user interface. If only one language is installed on the server, the screen does not display any options.

Note To support more than one language on a single server, you must modify the Welcome.osc file. The Multiling.osc file that is provided with Windows 2000 Server is an example of how to customize the Welcome.osc file. If more than one language is installed on the server, you can modify the Multiling.osc screen file to list the languages that you want to support. After you have modified Multiling.osc, rename it as Welcome.osc so that it replaces the existing Welcome.osc file.

You can customize the screen format to suit the needs of your organization. For example, you can modify the Login.osc file to display additional help (for example, "If problems occur during the logon process, contact the Help desk at extension 12345") when a logon error message appears.

The customized screen information is stored in a text file that is written in OSCML. Creating an OSCML file is similar to creating an HTML 2.0 file, but instead of using the <HTML> and </HTML> start and end tags, you use the OSCML start and end tags. Only the tags that are specified in this appendix are supported in OSCML.

There are other differences between OSCML and HTML 2.0. The Client Installation Wizard (.osc) files must contain text that is in the original equipment manufacturer (OEM) character set. The OEM character set supports a subset of the tags that HTML 2.0 supports, plus a few additional tags. Some HTML tags are missing in OSCML, most notably the anchor start and end tags (<A> and </A>) and the <IMG> tags. Because only text is allowed in the .osc file, the anchor and <IMG> tags are not supported and are not parsed. You can preview OSCML files in an HTML browser.

The following is a list of the tags that are supported in OSCML; each tag is described in detail after this list.

Note When (!HTML2.0) appears in a tag, the tag differs from the HTML 2.0 specification and is OSCML-specific.

- <OSCML> </OSCML> (!HTML2.0)
- <META SERVER ACTION="*server side action*">
- <META KEY= F1 | F3 | ENTER | ESC HREF="*screen name*">
- <META ACTION="*LOGIN*">
- <TITLE> *Title* </TITLE>
- <FOOTER> (!HTML2.0) *Footer* </FOOTER> (!HTML2.0)
- <BODY [LEFT="*left margin*" (!HTML2.0)] [RIGHT="*right margin*" (!HTML2.0)]>
- <PRE [LEFT="*left margin*" (!HTML2.0)] [RIGHT="*right margin*" (!HTML2.0)]> </PRE>
- <FORM ACTION="*screen name*"> </FORM>
- <INPUT [NAME="*osc var name*"] [TYPE="*TEXT* | *PASSWORD*"] [VALUE="*starting value*"] [SIZE="*display size*"] [MAXLENGTH="*max length*"]>

- <SELECT [MULTIPLE] [NOAUTO (!HTML2.0) [NAME="*osc var name*"] [SIZE="*height*"]> [<OPTION.....> <OPTION.....>] </SELECT>
- <OPTION [SELECTED] [TIP="*help text*" (!HTML2.0)] [VALUE="*return value*"]> *Item description*
- <SELECT>
- </SELECT>
-
- <P [LEFT="*left margin*" (!HTML2.0)] [RIGHT="*right margin*" (!HTML2.0)] > </P>
- <BOLD> *text* </BOLD>
- <FLASH> *text* </FLASH>
- <TIPAREA [LEFT="*left margin*"] [RIGHT="*right margin*"] [SIZE="*height*"]> (!HTML2.0)
- <OPTION VALUE="OSAUTO"TIP= ".....">

<OSCML> (!HTML2.0) </OSCML> (!HTML2.0)

Indicates that the file is an OSCML file. The <OSCML> tag must be the first tag that is found; otherwise, OSChooser ignores all other header tags (<META>, <TITLE>, and <FOOTER>). The </OSCML> tag must be at the end of the file. An item or text that is entered after the </OSCML> end tag is not displayed. (The space after </OSCML> is where you can place administrative comments and notes.)

<META SERVER ACTION="*server side action*">

Instructs the BINL service (BINLSVC) to perform a particular action. <META> tags are supported by HTML 2.0, but the specifics of the metadata vary from server to server. The following tags conform to HTML 2.0 and are specific to the OSChooser-BINLSVC conversation. Currently defined actions include the following:

- ENUM directory

  Enumerates the Setup Information File (SIF) under the X:\RemoteInstall\Setup *%LANGUAGE%*\images\ "*directory*" name.

  The path under which a SIF is searched for is the following:

  \images\"*directory name*"\

  OSC stores the results of its search in the %OPTIONS% variable. For an example of this tag, see the Tools.osc file that is created when you run the Client Installation Wizard screens; the file is located in the RemoteInstall\Oschooser\<*language*> subdirectory.

- WARNING

  Indicates that using WARNING represents the repartition warning screen. If "*Repartition=No*" is in the selected SIF, this screen is skipped; the server searches for the <META KEY=ENTER .....> line and goes to that screen.

- DNRESET

  Resets all variables to their initial state (as if you had logged on again).

- FILTER name

  Instructs BINL to check the Group Policy object of the domain to determine what choices can be displayed to the user.

- CHECKGUID

  Checks Active Directory to determine whether this globally unique identifier (GUID) duplicates an existing GUID.

<META KEY= F1 | F3 | ENTER | ESC HREF="*screen name*">

Goes to the screen name that is specified when the user presses the function key that is specified within the tag. (The .osc file name extension is appended by the server.) F1, F3, ENTER, and ESC are the only keys that are supported for use within this tag.

<META ACTION="LOGIN">

Instructs OSChooser to authenticate the user with the credentials that OSChooser has stored after the user has pressed ENTER. If authentication is not successful, OSChooser goes to Loginerr.osc. If authentication is successful, OSChooser goes to the screen in <META KEY=ENTER.....>. The variables that are sent from the *.osc screen are encrypted; all screens sent after this screen are encrypted as well.

Note All conversations prior to the start of the logon process are sent as plaintext. No screen variables except %LANGUAGE% are allowed to be sent to the server prior to the user logging on.

<TITLE> *Title* </TITLE>

Indicates a title line. The title is written at the top of the screen in blue text on a white background.

<FOOTER> (!HTML2.0) *Footer* </FOOTER> (!HTML2.0)

Indicates a footer line. The footer is written at the bottom of the screen in blue text on a white background.

<BODY [LEFT="*left margin*" (!HTML2.0)] [RIGHT="*right margin*" (!HTML2.0)] >

Indicates the end of the header tags and the beginning of the text that are to be shown to the user. You can set both the left margin and the right margin (in characters).

<PRE [LEFT="l*eft margin*" (!HTML2.0)] [RIGHT="*right margin*" (!HTML2.0)] > </PRE>

Indicates that the text that follows the tag is preformatted with no text wrapping. OSChooser truncates any lines that are longer than the current margin settings. You do have the option, however, of specifying the left and right margins (in characters).

<FORM ACTION="*screen name*"> </FORM>

Indicates that there are input controls on this page. It also specifies what screen is to be retrieved next. If no name is provided and the controls on the page within the file don't have "NAME=" tags in them, the resulting value for the control is the next screen that is retrieved. For an example of this tag, see Choice.osc. There can be only one <FORM.....> tag per page.

<INPUT [NAME="*osc var name*"] [TYPE="*TEXT | PASSWORD*"] [VALUE="*starting value*"] [SIZE="*display size*"] [MAXLENGTH="*max length*"]>

Creates an edit control on the page. The edit control in OSCML works the same as an edit control in HTML 2.0. For an example, see Login.osc. The following are descriptions of the parameters for this tag:

- NAME="*osc var name*"

  Indicates what OSC variable is sent back when a value is entered in the edit box by the user.

- TYPE="*TEXT* | *PASSWORD*"

  Defaults to TEXT. Specify the password type PASSWORD to make all text that has been entered appear as asterisks. The password type HIDDEN is not supported by OSChooser.

- VALUE="*starting value*"

  Allows a default value to be added by the user to the edit box.

- SIZE="*display size*"

  Specifies the number of characters that the user can enter in the edit box.

Note The default length of the edit box equals the value that you have entered for SIZE= plus four characters for the pair of brackets ([ ]) and for the two extra end spaces.

- MAXLENGTH="*max length*"

  Indicates the maximum length of the string that can be entered in the edit box. For an example, see custom.osc. If the value of the SIZE parameter is greater than the number of characters that remain on the line, the INPUT tag defaults to the automatic size. The *automatic size* is calculated by subtracting the number of characters that you have entered from the number of characters for the default length of the edit box. REMAINING is the number of character spaces from the end of the string to the right margin minus four character spaces for the pair of brackets and two extra end spaces.

<SELECT [MULTIPLE] [NOAUTO (!HTML2.0)] [NAME="*osc var name*"] [SIZE="*height*"]> [<OPTION.....> .. <OPTION.....>] </SELECT>

Creates a list from which users can use arrow keys to select an option. An example from Choice.osc is <SELECT SIZE=10>. The following <OPTION> tags specify what the option list contains:

- <OPTION VALUE="OSAUTO" TIP="....."> is the **Automatic Setup** option. You can use this option to install an operating system on your computer. It's the easiest way to install an operating system on your computer because most installation options are already configured by your network administrator.
- <OPTION VALUE="CUSTOM"> is the **Custom Setup** option. You can use this option to define a unique name for the computer that is being set up and specify where the computer account is to be located in Active Directory. Use this option if you are setting up the computer for someone else in your company.
- <OPTION VALUE="RESTART"> is the **Restart a Previous Setup Attempt** option. You can use this option to restart a computer on which a previous remote installation attempt has been detected.
- <OPTION VALUE="TOOLS"> is the **Maintenance and Troubleshooting** option. You can use this option to gain access to tools that you can use to keep your computer up-to-date and to troubleshoot problems.

<SELECT>

Indicates the beginning of a list. The visible size of the list defaults to 1 and can be overridden by the SIZE field. There is no user interface to indicate extra items in the list (such as scroll bars), so make sure that the list is large enough to show all your items. For an example, see Choice.osc.

This tag is often used with the %OPTIONS% variable after a <META SERVER ACTION="ENUM .....> tag has been used to list the SIFs available under a particular directory. For an example, see Tools.osc.

If the <SELECT> tag has only one <OPTION> tag, OSChooser automatically selects that option and continues. The <NOAUTO> tag prevents OSChooser from automatically selecting an option. For an example, see Tools.osc.

</SELECT>

Indicates the end of a list.

<OPTION [SELECTED] [TIP="*help text*" (!HTML2.0)] [VALUE="r*eturn value*"]>*Item description*

Indicates what items are going to be shown in a list and is used in conjunction with the <SELECT> tag. The text that follows the <OPTION> tag is displayed in the list. The <VALUE> tag indicates what string is sent back to the server. The <TIP> tag is a way to display Help text in the <TIPAREA>. The <SELECTED> tag indicates what items must be selected first. If there is more than one <OPTION> tag designated, the last option is used. For an example, see Choice.osc.

**
**

Indicates a new line break.

<P [LEFT="l*eft margin*" (!HTML2.0)] [RIGHT="*right margin*" (!HTML2.0)] > </P>

Indicates a new paragraph. In OSCML, this tag acts like the <BR> tag to conserve space, except that it has options that allow you to change the margins.

<BOLD> *text* </BOLD>

Displays the word "text" in white type. For an example, see Warning.osc.

<FLASH> *text* </FLASH>

Flashes the word "text" on the screen. For an example, see Warning.osc.

Note You can apply either the <BOLD> or <FLASH> tag to text, but not both. One tag overrides the other.

<TIPAREA [LEFT="*left margin*"] [RIGHT="*right margin*"] [SIZE="*height*"]> (!HTML2.0)

Indicates where any tips can be displayed. All tips are wrapped to the margins. The default height of the tip area is the amount of screen area that remains, unless the amount of screen area is overridden by the SIZE parameter. For an example, see Choice.osc.

OPTION VALUE="OSAUTO" TIP="....."

Indicates automatic installation of the operating system. This is the easiest way to install an operating system on a computer. When this tag is used, the user sees the option AUTOMATIC SETUP in the list; when that option is selected, the text that is noted by the TIP="....." is displayed in the area noted by <TIPAREA> in the *.osc file.

Glossary

.

.adm The file name extension for Administrative Templates files.

.msi The file name extension for Windows Installer package files.

8

88 class A class defined before 1993 not required to fall into one of the structural, abstract, or auxiliary categories. This type of class is specified by a value of 0 in the objectClass category.

A

abstract classes Templates used only to derive new Structural classes. Abstract classes cannot be instantiated in the directory.

access control The security mechanism in Windows NT and Windows 2000 that determines which objects a security principal can use and how the security principal can use them. See also authorization; security principal.

access control entry (ACE) An entry in an access control list (ACL) containing the security ID (SID) for a user or group and an access mask that specifies which operations by the user or group are allowed, denied, or audited. See also access control list; access mask; security descriptor.

access control list (ACL) A list of security protections that apply to an entire object, a set of the object's properties, or an individual property of an object. There are two types of access control lists: discretionary and system. See also access control entry; discretionary access control list; security descriptor; system access control list.

access mask A 32-bit value that specifies the rights that are allowed or denied in an access control entry (ACE) of an access control list (ACL). An access mask is also used to request access rights when an object is opened. See also access control entry.

ACE See access control entry.

active cluster member A node that is running and participating in cluster operations.

Active Directory The directory service included with Windows 2000 Server. It stores information about objects on a network and makes this information available to users and network administrators. Active Directory gives network users access to permitted resources anywhere on the network using a single logon process. It provides network administrators with an intuitive hierarchical view of the network and a single point of administration for all network objects. See also directory; directory service.

Active Directory replication Synchronization of directory partition replicas between Windows 2000 domain controllers. Directory partition replicas are writable on each domain controller, except for Global Catalog replicas. Replication automatically copies the changes from a specified directory partition replica to all other domain controllers that hold the same directory partition replica. More specifically, a server called the "destination" pulls changes from another server called the "source." See also directory partition; File Replication Service; multimaster replication; replication.

Active Directory Service Interfaces (ADSI) A set of high-level programming interfaces that provide a single, consistent, open set of interfaces that enables Windows 2000, Windows NT, Windows 98 and Windows 95 client applications to access several network directory services, including Active Directory. ADSI provides the means for client applications of directory services to use one set of interfaces to communicate with any namespace that provides an ADSI implementation (provider)

active/active The cluster configuration of an application in which the application runs on all nodes at the same time. See also active/passive.

active/passive The cluster configuration of an application in which the application runs on only one node at a time. See also active/active.

additional domain controller When installing Active Directory, a domain controller that is being added to an existing Windows 2000 domain.

administrative template (.adm file) A text file used by the Group Policy console as a source to generate the user interface for Group Policy settings an administrator can set. Windows NT 4.0 used an earlier version of .adm files to generate user interface for registry-based System Policy settings in the System Policy Editor.

ADSI See Active Directory Service Interfaces.

ADSI provider COM objects that implement ADSI for a particular namespace (for example, an LDAP namespace such as Active Directory).

ambiguous name resolution In an LDAP search, the process of searching for a string value in a set of attributes by using one filter of the form (ANR=string). A defined set of attributes is available for ANR searches, and when the (ANR=string) filter is encountered, the filter is expanded to include a search of every attribute in the ANR set.

answer file A text file that you can use to provide automated input for unattended installation of Windows 2000. This input includes parameters to answer the questions required by Setup for specific installations. In some cases, you can use this text file to provide input to wizards, such as the Active Directory Installation wizard, which is used to add Active Directory to Windows 2000 Server through Setup. The default answer file for Setup is known as Unattend.txt.

anti-replay A feature for preventing replay attacks. See also replay attack.

assigned applications Applications that are assigned to users or computers by an administrator using the Software Installation snap-in an extension to Group Policy. Assigned applications are always available to users or computers managed by a Group Policy object. User- assigned applications appear to be installed on a user's computer and can be installed by selecting the software from the Start menu, or selecting a shortcut on the desktop. Applications assigned to a computer are installed when the computer is turned on.

asymmetric key algorithm See public-key algorithm.

atomic transaction In Active Directory, database transactions that are either completed in full or are not applied at all. If for any reason an error occurs and a transaction is unable to complete all of its steps, the system is returned to the state it was in before the transaction was started.

atomic update In a server cluster, the means by which the cluster registry key is replicated to all nodes. If any part of an atomic update on a node fails, all of it fails. In Active Directory, the method of updating an Active Directory attribute. An LDAP directory server processes each update request as an atomic action: The request either is committed and all its effects are durable, or it is terminated and has no effect. In Active Directory replication, the scope of an atomic update is the object. All of the attribute changes made to an object that are replicated at the same time are applied together atomically.

attribute (object) In Active Directory, an attribute describes characteristics of an object and the type of information an object can hold. For each object class, the schema defines what attributes an instance of the class must have and what additional attributes it might have.

attributeID The object identifier that is the unique name of an attribute.

attributeSyntax The syntax object identifier for this attribute.

authentication A basic security function of cryptography. Authentication verifies the identity of the entities that communicate over the network. For example, the process that verifies the identity of a user who logs on to a computer either locally, at a computer's keyboard, or remotely, through a network connection. See also cryptography; confidentiality; integrity; Kerberos authentication protocol; nonrepudiation; NTLM authentication protocol.

authenticator A data structure used by one party to prove that another party knows a secret key. In the Kerberos authentication protocol, authenticators include timestamps, to prevent replay attacks, and are encrypted with the session key issued by the Key Distribution Center (KDC). See also Kerberos authentication protocol; Key Distribution Center; replay attack; secret key.

authoritative restore In Backup, a type of restore operation on a Windows 2000 domain controller in which the objects in the restored directory are treated as authoritative, replacing (through replication) all existing copies of those objects. Authoritative restore is applicable only to replicated System State data such as Active Directory data and File Replication service data. The Ntdsutil.exe utility is used to perform an authoritative restore. See also nonauthoritative restore; System State.

authorization The process that determines what a user is permitted to do on a computer system or network. For remote access or demand-dial routing connections, the verification that the connection attempt is allowed. Authorization occurs after successful authentication. See also access control; authentication.

auxiliaryClass A multivalued property that specifies the auxiliary classes from which this class inherits. For an existing classSchema object, values can be added to this property but not removed.

Each value is the lDAPDisplayName of a class. You must ensure that the class exists or will exist when the new class is written to the directory. If one of the classes does not exist, the classSchema object fails to be added to the directory.

The full set of auxiliary classes is the union of the systemAuxiliaryClass and auxiliaryClass on this class as well as the systemAuxiliaryClass and auxiliaryClass properties of all inherited classes.

availability A measure of the fault tolerance of a computer and its programs. A highly available computer runs 24 hours a day, 7 days a week. See also fault tolerance.

B

base DIT The directory that is installed during a fresh install of a Windows 2000 domain controller.

base search See search scope.

BINL service See Boot Information Negotiation Layer service.

BINLSVC See Boot Information Negotiation Layer Service.

Boot Information Negotiation Layer (BINL) service A service that runs on Windows 2000 Server that acts on client boot requests. For example, by using Remote Installation Service the BINL service listens for and answers DHCP (PXE) requests. It also services Client Installation Wizard requests. BINL directs the client to the files needed to start the installation process. This service also checks Active Directory to verify credentials, determine whether a client needs service, and whether to create a new or reset an existing computer account on behalf of the client.

bridgehead server In Active Directory replication, a single server in each site that is designated to perform site-to-site replication. Bridgehead servers are designated automatically by the KCC, or they can be assigned manually by an administrator. Bridgehead servers ensure that most replication occurs within sites rather than between sites.

brute force attack See key search attack.

buffer overflow attack An attack in which an attacker exploits a weakness in a program or service to force a buffer overflow condition and then cause malicious code (provided by the attacker) to run in the computer's memory. Through a successful buffer overflow attack, an attacker can take control of the computer with the rights and permissions of the system and the logged–on user.

bulk encryption A process in which large amounts of data, such as files, e-mail messages, or online communications sessions, are encrypted for confidentiality. It is usually done with a symmetric key algorithm. See also encryption; symmetric key encryption.

C

certificate A digital document that is commonly used for authentication and secure exchange of information on open networks, such as the Internet, extranets, and intranets. A certificate securely binds a public key to the entity that holds the corresponding private key. Certificates are digitally signed by the issuing certification authority and can be issued for a user, a computer, or a service. The most widely accepted format for certificates is defined by the ITU-T X.509 version 3 international standard. See also certification authority; private key; public key.

certificate revocation list (CRL) A document maintained and published by a certification authority that lists certificates that have been revoked. A CRL is signed with the private key of the CA to ensure its integrity. See also certificate; certification authority.

Certificate Services The Windows 2000 service that issues certificates for a particular CA. It provides customizable services for issuing and managing certificates for the enterprise. See also certificate; certification authority.

certificate stores Windows 2000 stores public key objects, such as certificates and certificate revocation lists, in logical stores and physical stores. Logical stores group public key objects for users, computers, and services. Physical stores are where the public key objects are actually stored in the registry of local computers (or in Active Directory for some user certificates). Logical stores contain pointers to the public key objects in the physical stores. Users, computers, and services share many public key objects, so logical stores enable public key objects to be shared without requiring the storage of duplicates of the objects for each user, computer, or service.

certificate template A Windows 2000 construct that profiles certificates (that is, it pre-specifies format and content) based on their intended usage. When requesting a certificate from a Windows 2000 enterprise certification authority (CA), certificate requesters are, depending on their access rights, able to select from a variety of certificate types that are based on certificate templates, such as "User" and "Code Signing." See also certificate; enterprise certification authority.

certificate trust list (CTL) A signed list of root certification authority certificates that an administrator considers reputable for designated purposes, such as client authentication or secure e-mail. See also certificate; certification authority; root certificate; root certification authority.

Certificates console A snap-in to the MMC. This console is used to manage certificate stores for users, computers, and services. See also certificate; certificate stores.

certification authority (CA) An entity responsible for establishing and vouching for the authenticity of public keys belonging to users (end entities) or other certification authorities. Activities of a certification authority can include binding public keys to distinguished names through signed certificates, managing certificate serial numbers, and certificate revocation. See also certificate; public key.

Certification Authority console A Snap-in to the MMC. This console is used to configure and manage Windows 2000 certification authorities. See also certification authority.

certification hierarchy A model of trust for certificates in which certification paths are created through the establishment of parent-child relationships between certification authorities. See also certification authority; certification path.

certification path An unbroken chain of trust from a certificate to the root certification authority in a certification hierarchy. See also certification hierarchy; certificate.

Certification Practices Statement (CPS) A formal statement that describes the certification policies and practices of a certification authority. See also certification authority.

change log See quorum log.

checkpoint In a server cluster node's registry, a snapshot of the registry cluster key or of an application key. The checkpoint is written to the quorum disk when certain events take place, such as a node failure. See also cluster database.

child domain For DNS and Active Directory, a domain located in the namespace tree directly beneath another domain name (its parent domain). For example, "example.reskit.com" is a child domain of the parent domain, "reskit.com." Child domain is also called subdomain. See also directory partition; domain; parent domain.

child object An object that is the immediate subordinate of another object in a hierarchy. A child object can have only one immediate superior, or parent, object. In Active Directory, the schema determines what classes of objects can be child objects of what other classes of objects. Depending on its class, a child object can also be the parent of other objects. See also object; parent object.

cipher The method of forming a hidden message. The cipher is used to transform a readable message called plaintext (also sometimes called cleartext) into an unreadable, scrambled, or hidden message called ciphertext. Only someone with a secret decoding key can convert the ciphertext back into its original plaintext. See also ciphertext; plaintext; cryptography.

ciphertext Text that has been encrypted using an encryption key. Ciphertext is meaningless to anyone who does not have the decryption key. See also decryption; encryption; encryption key; plaintext.

class A category of objects that share a common set of characteristics. Each object in the directory is an instance of one or more classes in the schema.

cleartext See plaintext.

client Any computer or program connecting to, or requesting services of, another computer or program. See also server.

client request A service request from a client to a server or, for Network Load Balancing, a cluster of computers. Network Load Balancing forwards each client request to a specific host within the cluster according to the system administrator's load-balancing policy. See also client; cluster; host; server.

CLUSDB In a server cluster, the snapshot of the startup cluster registry key stored in the local disk.

cluster A group of independent computer systems known as nodes or hosts, that work together as a single system to ensure that mission-critical applications and resources remain available to clients. A server cluster is the type of cluster that the Cluster service implements. Network Load Balancing provides a software solution for clustering multiple computers running Windows 2000 Server that provides networked services over the Internet and private intranets.

Cluster Administrator An application (Cluadmin.exe) used to configure a cluster and its nodes, groups, and resources. Cluster Administrator can run on any member of the trusted domain regardless of whether the computer is a cluster node. See also cluster; Cluster Administrator extension; Cluster.exe; node; resource.

Cluster Administrator extension A dynamic-link library (DLL) that enables Cluster Administrator to manage a custom resource type. A Cluster Administrator extension uses the Cluster Administrator Extension API. See also cluster; Cluster Administrator; resource.

cluster API A collection of functions implemented by the cluster software and used by a cluster-aware client or server application, a cluster management application, or a resource DLL. The cluster API is used to manage the cluster, cluster objects, and the cluster database. See also cluster; dynamic-link library; node; resource; resource DLL.

cluster database The database of configuration data (cluster objects and their settings) pertinent to the cluster. This database is the product of the cluster registry key checkpoint and the changes recorded in the quorum log. A local copy of this database is maintained by all the nodes of the cluster hive in the registry. See also checkpoint.

cluster disk A disk on a shared bus connected to the cluster nodes, which all the cluster nodes can access (though not at the same time).

cluster log An optionally enabled trace record of Cluster service events on a node. Not synonymous with quorum log.

cluster object A physical or logical unit managed by the Cluster service. Cluster objects include nodes, networks, network interfaces (see network adapter), groups, resources, and resource types.

cluster registry key The portion of the system registry on each node that contains the property and configuration data for the cluster, nodes, and specified resources. The cluster key is synchronized on all nodes in the cluster and on the quorum disk.

Cluster service Clussvc.exe, the primary executable of the Windows Clustering component that creates a server cluster, controls all aspects of its operation, and manages the cluster database. Each node in a server cluster runs one instance of the Cluster service.

cluster-aware The classification of an application or service that runs on a server cluster node, is managed as a cluster resource, and is designed to be aware of and interact with the server cluster environment. Cluster-aware applications use the Cluster API to receive status and notification information from the server cluster. See also Cluster API; cluster-unaware application; node; resource DLL.

cluster-capable disk A disk that can be accessed by all server cluster nodes.

cluster-unaware application In a server cluster, the classification of an application or service that can run on a node and be managed as a cluster resource but does not support the Cluster API and therefore has no inherent knowledge of its environment. See also cluster-aware; node.

Cluster.exe An alternative to using Cluster Administrator to administer clusters from the Windows 2000 command prompt. Cluster.exe can be called from command scripts to automate many cluster administration tasks. See also Cluster Administrator.

cn (Common-Name) The descriptive relative distinguished name for the schema object.

code signing The process of digitally signing software code to ensure its integrity and provide assurance of its origin.

Comma Separated Value (CSV) scripts

Windows 2000 includes a command-line utility, CSVDE, to import directory objects using .csv files and export directory objects as .csv files. CSV scripts are targeted for ease-of-use. The first line in the script identifies the attributes in the lines that follow. Columns are separated by commas. The file format is compatible with the Microsoft Excel CSV format, so that files are easily created. Use Excel or any other tool that can read and write .csv files. A benefit of using CSVDE is that it supports Unicode.

Comma Separated Value Directory Exchange (CSVDE)

A command–line utility that allows you to import and export objects to and from Active Directory. You can not create, modify, and delete directory objects using this utility. By using this utility, objects are stored in the Microsoft Comma-Separated Value (CSV) file format. The CSV file format is supported by many other applications, such as Microsoft Excel, that can read and save data in the CSV file format. Also, Microsoft Exchange Server administration tools can import and export data using the CSV format. CSVDE can be run on a Windows 2000 server or copied to a Windows 2000 workstation.

confidentiality A basic security function of cryptography. Confidentiality provides assurance that only authorized users can read or use confidential or secret information. Without confidentiality, anyone with network access can use readily available tools to eavesdrop on network traffic and intercept valuable proprietary information. For example, an Internet Protocol security service that ensures a message is disclosed only to intended recipients by encrypting the data. See also cryptography; authentication; integrity; nonrepudiation.

connection object An Active Directory object that 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 identifies the replication source server, contains a replication schedule, and specifies a replication transport. Connection objects are created automatically by the Knowledge Consistency Checker, but they can also be created manually. Automatically generated connections must not be modified by the user unless they are first converted into manual connections.

console tree The tree view pane in a Microsoft Management Console (MMC) that displays the hierarchical namespace. By default it is the left pane of the console window, but it can be hidden. The items in the console tree (for example, Web pages, folders, and controls) and their hierarchical organization determines the management capabilities of a console. See also Microsoft Management Console (MMC); namespace.

container object An object that can logically contain other objects. For example, a folder is a container object. See also object.

cross-reference object In Active Directory, an object that contains knowledge of one directory partition. Cross reference objects are used to generate referrals to other directory partitions and to foreign directories. On a specified domain controller, subject to replication latency, the combination of all cross references provides knowledge of all directory partitions in the forest, irrespective of location in the directory tree.

cryptanalysis The art and science of breaking ciphertext. In contrast, the art and science of keeping messages secure is cryptography. See also ciphertext; cryptography; plaintext.

crypto-accelerator board A hardware device that speeds up cryptographic operations by offloading operations to a special processor on the board.

CryptoAPI (CAPI) An application programming interface (API) that is provided as part of Windows 2000. CryptoAPI provides a set of functions that allow applications to encrypt or digitally sign data in a flexible manner while providing protection for private keys. Actual cryptographic operations are performed by independent modules known as cryptographic service providers (CSPs). See also cryptographic service provider; private key.

cryptographic key See encryption key.

cryptographic service provider (CSP) An independent software module that performs cryptography operations such as secret key exchange, digital signing of data, and public key authentication. Any Windows 2000 service or application can request cryptography operations from a CSP. See also CryptoAPI.

cryptography The art and science of information security. It provides four basic information security functions: confidentiality, integrity, authentication, and nonrepudiation. See also confidentiality; integrity; authentication; nonrepudiation.

cryptology The science that encompasses both cryptography and cryptanalysis. See also cryptanalysis; cryptography.

CSVDE See Comma-Separated Value Directory Exchange.

custom resource type A resource type defined by a third-party developer using the Cluster service API.

D

data decryption field (DDF) A header field, in a file encrypted by using the Encrypting File System, that contains the file encryption key encrypted with the file encryptor's public key.

data recovery field (DRF) A header field, in a file encrypted by using the Encrypting File System, that contains the file encryption key encrypted with a recovery agent's public key.

Database Manager The Cluster service component that controls access to the cluster database.

datagram An unacknowledged packet of data sent to another network destination. The destination can be another device directly reachable on the local area network (LAN) or a remote destination reachable using routed delivery through a packet-switched network.

decryption The process of making encrypted data readable again by converting ciphertext to plaintext. See also ciphertext; encryption; plaintext.

deep search See search scope.

defaultObjectCategory The distinguished name of the classSchema object for the class that should be used by default as the objectCategory for new instances of this class. This is an indexed property used to make object class searches fast and efficient.

By default, defaultObjectCategory is set to the distinguished name of the classSchema object for this class. If this object will be frequently queried by the value of a super class rather than the object's own class, the defaultObjectCategory can be defined in the schema to the desired value.

If you are sub-classing a structural class, best practice is to set this to the same value as the superclass. This allows the standard UI to "find" your sub-class.

defragmentation The process of rewriting parts of a file to contiguous sectors on a hard disk to increase the speed of access and retrieval. When files are updated, the computer tends to save these updates on the largest continuous space on the hard disk, which is often on a different sector than the other parts of the file. When files are thus fragmented, the computer must search the hard disk each time the file is opened to find all of the parts of the file, which slows down response time. In Active Directory, defragmentation rearranges how the data is written in the directory database file to compact it. See also fragmentation.

delegation The ability to assign responsibility for management and administration of a portion of the namespace to another user, group, or organization. For DNS, a name service record in the parent zone that lists the name server authoritative for the delegated zone. See also inheritance; parenting.

denial-of-service attack An attack in which an attacker exploits a weakness or a design limitation of a network service to overload or halt the service, so that the service is not available for use. This type of attack is typically launched to prevent other users from using a network service such as a Web server or a file server.

dependency tree A discrete set of resources that are connected to each other by dependency relationships. All resources in a specified dependency tree must be members of a single group. See also resource.

Dfs See Distributed file system.

Dfs link Part of the Distributed file system (Dfs) topology that lies below the Dfs root and forms a connection to one or more shared folders or another Dfs root. It does this by mapping a DNS name to the standard UNC of the target shared folder.

Dfs root A Server Message Block share at the top of the Dfs topology that is the starting point for the links and shared files that make up the Dfs namespace. A Dfs root can be defined at the domain level, for domain-based operation, or at the server level, for stand-alone operation. Domain-based Dfs can have multiple roots in the domain but only one root on each server. See also namespace.

Dfs shared folder Files or folders in the Dfs namespace that are shared by users with proper permissions. Shared folders can exist at the root level (domain-based Dfs only) or be referred to by Dfs links.

Dfs topology The overall logical hierarchy of a Distributed file system, including elements such as roots, links, shared folders, and replica sets, as depicted in the Dfs administrative console. This is not to be confused with the Dfs namespace, which is the logical view of shared resources seen by users.

dictionary attack An attack in which an attacker tries known words in the dictionary and numerous common password names in an attempt to "guess" the password. Because most users prefer easily remembered passwords, dictionary attacks are often a shortcut to finding a password in significantly less time than key search (brute force) attacks would take to find the same password. See also key search attack.

Diffie-Hellman (DH) algorithm An algorithm that predates Rivest-Shamir-Adleman (RSA) encryption and offers better performance. It is one of the oldest and most secure algorithms used for key exchange. The two parties publicly exchange keying information, which Windows 2000 additionally protects with hash function encryption. Neither party ever exchanges the actual key; however, after their exchange of keying material, each is able to generate the identical shared key. At no time is the actual key ever exchanged.

Diffie-Hellman Key Agreement See Diffie-Hellman (DH) algorithm.

digital certificate See certificate.

digital signature A means for originators of a message, file, or other digitally–encoded information to bind their identity to the information. The process of digitally signing information entails transforming the information, as well as some secret information held by the sender, into a tag called a signature. Digital signatures are used in public key environments and they provide nonrepudiation and integrity services. See also public key cryptography.

Digital Signature Algorithm (DSA) See Digital Signature Security Standard.

Digital Signature Security Standard (DSS) A standard that uses the Digital Signature Algorithm (DSA) for its signature algorithm and SHA-1 as its message hash algorithm. DSA is a public key algorithm that is used only to generate digital signatures and cannot be used for data encryption. Digital Signature Standard is also called DSS.

direct memory access (DMA) Memory access that does not involve the microprocessor. DMA is frequently used for data transfer directly between memory and a peripheral device, such as a disk drive.

directory An information source that contains information about computer files or other objects. In a file system, a directory stores information about files. In a distributed computing environment (such as a Windows 2000 domain), the directory stores information about objects such as printers, applications, databases, and users.

directory partition A contiguous subtree of Active Directory that is replicated as a unit to other domain controllers in the forest that contain a replica of the same subtree. In Active Directory, a single server always holds at least three directory partitions: schema, (class and attribute definitions for the directory); configuration (replication topology and related metadata); domain (subtree that contains the per-domain objects for one domain). The schema and configuration directory partitions are replicated to every domain controller in a specified forest. A domain directory partition is replicated only to domain controllers for that domain. In addition to a full, writable replica of its own domain directory partition, a Global Catalog server also holds partial, read-only replicas of all other domain directory partitions in the forest. See also full replica; Global Catalog; partial replica.

directory service Both the directory information source and the service that make the information available and usable. A directory service enables the user to find an object given any one of its attributes. See also Active Directory; directory.

directory store The physical storage for Active Directory directory partition replicas on a given domain controller. The store is implemented using the Extensible Storage Engine.

directory system agent (DSA) The process that manages and provides access to stored directory information.

directory tree A hierarchy of objects and containers in a directory that can be viewed graphically as an upside-down tree, with the root object at the top. Endpoints in the tree are usually single (leaf) objects, and nodes in the tree, or branches, are container objects. A tree shows how objects are connected in terms of the path from one object to another. A simple tree is a single container and its objects. A contiguous subtree is any unbroken path in the tree, including all the members of any container in that path.

discontiguous namespace Namespace that is based on different DNS root domain names, such as that of multiple trees in the same forest. See also flat namespace; hierarchical namespace; namespace.

discretionary access control list (DACL) The part of an object's security descriptor that grants or denies specific users and groups permission to access the object. Only the owner of an object can change permissions granted or denied in a DACL; thus access to the object is at the owner's discretion. See also access control entry; object; security descriptor; system access control list.

display specifiers Objects in Active Directory that store localized graphical user interface information. Display specifiers enable the graphical user interface to be extended for each class of object in Active Directory.

distinguished name (DN) A name that uniquely identifies an object by using the relative distinguished name for the object, plus the names of container objects and domains that contain the object. The distinguished name identifies the object as well as its location in a tree. Every object in Active Directory has a distinguished name. An example of a distinguished name is CN=MyName,CN=Users,DC=Reskit,DC=Com.

This distinguished name identifies the "MyName" user object in the reskit.com domain.

Distributed file system (Dfs) A Windows 2000 service consisting of software residing on network servers and clients that transparently links shared folders located on different file servers into a single namespace for improved load sharing and data availability.

DNS See Domain Name System.

DNS server A computer that runs DNS server programs containing name-to-IP address mappings, IP address-to-name mappings, information about the domain tree structure, and other information. DNS servers also attempt to resolve client queries.

domain In Windows 2000 and Active Directory, a collection of computers defined by the administrator of a Windows 2000 Server network that share a common directory database. A domain has a unique name and provides access to the centralized user accounts and group accounts maintained by the domain administrator. Each domain has its own security policies and security relationships with other domains and represents a single security boundary of a Windows 2000 computer network. Active Directory is made up of one or more domains, each of which can span more than one physical location. For DNS, a domain is any tree or subtree within the DNS namespace. Although the names for DNS domains often correspond to Active Directory domains, DNS domains should not be confused with Windows 2000 and Active Directory networking domain.

domain controller For a Windows NT Server or Windows 2000 Server domain, the server that authenticates domain logons and maintains the security policy and the security accounts master database for a domain. Domain controllers manage user access to a network, which includes logging on, authentication, and access to the directory and shared resources.

domain controller discovery A process by which the Net Logon service attempts to locate a domain controller that is running Windows NT Server in the trusted domain. After a domain controller has been discovered, it is used for subsequent user account authentication.

domain controller locator (Locator) An algorithm that runs in the context of the Net Logon service and that finds domain controllers on a Windows 2000 network. Locator can find domain controllers by using DNS names (for IP/DNS-compatible computers) or by using NetBIOS names (for computers that are running Windows 3.*x*, Windows for Workgroups, Windows NT 3.5 or later, Windows 95, or Windows 98, or it can be used on a network where IP transport is not available).

domain hierarchy The parent-child tree structure of domains.

domain local group A Windows 2000 group only available in native mode domains and can contain members from anywhere in the forest, in trusted forests, or in a trusted pre-Windows 2000 domain. Domain local groups can only grant permissions to resources within the domain in which they exist. Typically, domain local groups are used to gather security principals from across the forest to control access to resources within the domain.

domain name In Windows 2000 and Active Directory, the name given by an administrator to a collection of networked computers that share a common directory. For DNS, domain names are specific node names in the DNS namespace tree. DNS domain names use singular node names, known as "labels," joined together by periods (.) that indicate each node level in the namespace. See also Domain Name System (DNS); namespace.

domain name label Each part of a full DNS domain name that represents a node in the domain namespace tree. Domain names are made up of a sequence of labels, such as the three labels ("noam," "reskit," and "com") that make up the DNS domain name "noam.reskit.com." Each label used in a DNS name must have 63 or fewer characters.

Domain Name System (DNS) A hierarchical naming system used for locating domain names on the Internet and on private TCP/IP networks. DNS provides a service for mapping DNS domain names to IP addresses, and vice versa. This allows users, computers, and applications to query the DNS to specify remote systems by fully qualified domain names rather than by IP addresses. See also domain; Ping.

domain naming master The domain controller that has the domain naming master role is the only domain controller that can do the following: Add new domains to the forest; Remove existing domains from the forest; Add or remove cross-reference objects to external directories. See also Active Directory; domain controller; multimaster replication; operations master; replication.

domain tree In DNS, the inverted hierarchical tree structure that is used to index domain names. Domain trees are similar in purpose and concept to the directory trees used by computer filing systems for disk storage. See also domain name; namespace.

domain-based Dfs An implementation of Dfs that stores its configuration information in Active Directory. Because this information is made available on every domain controller in the domain, domain-based Dfs provides high availability for any distributed file system in the domain. A domain-based Dfs root has the following characteristics: it must be hosted on a domain member server, it has its topology published automatically to Active Directory, it can have root-level shared folders and it supports root and file replication through File Replication Service.

dynamic-link library (DLL) A feature of the Microsoft Windows family of operating systems and the OS/2 operating system. DLLs allow executable routines, generally serving a specific function or set of functions, to be stored separately as files with .dll extensions, and to be loaded only when needed by the program that calls them.

E

encrypted data recovery agent account An account that can be used to decrypt a file encrypted by using the Encrypting File System (EFS) if the file owner's decryption key becomes unavailable.

Encrypting File System (EFS) A new feature in Windows 2000 that protects sensitive data in files that are stored on disk using the NTFS file system. It uses symmetric key encryption in conjunction with public key technology to provide confidentiality for files. It runs as an integrated system service, which makes EFS easy to manage, difficult to attack, and transparent to the file owner and to applications.

encryption The process of disguising a message or data in such a way as to hide its substance.

encryption key A bit string that is used in conjunction with an encryption algorithm to encrypt and decrypt data. See also public key; private key; symmetric key.

enterprise certification authority A Windows 2000 certification authority that is fully integrated with Active Directory. See also certification authority; stand-alone certification authority.

Extensible Storage Engine The Active Directory database engine. ESE (Esent.dll) implements a transacted database system, which means that it uses log files to ensure that committed transactions are safe.

external reference In Active Directory, knowledge about a referral location that is external to the forest. Virtual containers and foreign containers are external references.

external trust relationship A manually–created trust relationship between Windows 2000 domains that are in different forests or between a Windows 2000 domain and a domain whose domain controller is running Windows NT 4.0 or earlier.

F

factoring attack An attack on a public key encryption algorithm in which the attacker tries all possible factors to discover the private key of a public/private key pair. This attack is similar to the key search attack that can be conducted on symmetric key encryption algorithms, but the number of possible factors varies depending on the public key algorithm.

failback (v., fail back) In a server cluster, the moving of a failed-over group to the next node on the group's Preferred Owners list. See also failover; node; resource.

failover (v., fail over) In a server cluster, the means of providing high availability. Upon failure, either of a resource in a group or of the node where the group is online, the Cluster service takes the group offline on that node, and then brings it online on another node. See also node; resource.

fault tolerance The assurance of data integrity when hardware failures occur. On the Windows NT and Windows 2000 platforms, fault tolerance is provided by the Ftdisk.sys driver.

File Replication service A multithreaded replication engine that allows simultaneous replication of files between different computers. File Replication service replaces the LMRepl service that is used in Microsoft Windows NT.

FIPS 140-1 A standard entitled "Security Requirements for Cryptographic Modules." FIPS 140-1 describes government requirements that hardware and software cryptomodules must meet for Sensitive, but Unclassified (SBU) use. FIPS 140-1 is also called Federal Information Processing Standard 140-1.

flat namespace A namespace that is unstructured and cannot be partitioned, such as the network basic input/output system (NetBIOS) namespace. In a flat namespace, every object must have a unique name. See also namespace; hierarchical namespace; noncontiguous namespace.

Flexible Single Master Operations (FSMO) Active Directory operations that are not permitted to occur at different places in the network at the same time. Each role controls another specific set of directory changes. For each role, only the domain controller holding that role can make the associated directory changes. For example, Active Directory performs schema updates to prevent conflicts in a single-master fashion. Only one domain controller in the entire forest, the domain controller holding the schema master role, accepts updates to schema objects. An administrator can shift the schema master role from one domain controller to another as the need arises, but at any moment only one domain controller holds the schema master role.

forest A collection of one or more Windows 2000 Active Directory trees, organized as peers and connected by two-way transitive trust relationships between the root domains of each tree. All trees in a forest share a common schema, configuration, and Global Catalog. When a forest contains multiple trees, the trees do not form a contiguous namespace.

FORTEZZA A family of security products, including PCMCIA-based cards, compatible serial port devices, combination cards (such as FORTEZZA/Modem and FORTEZZA/Ethernet), server boards, and others. FORTEZZA is a registered trademark held by the National Security Agency.

fragmentation The scattering of parts of the same disk file over different areas of the disk. Fragmentation occurs as files on a disk are deleted and new files are added. It slows disk access and degrades the overall performance of disk operations, although usually not severely. See also defragmentation.

FSMO See Flexible Single Master Operations, (pronounced "fizmo")

FSMO role owner The computer where an operation is allowed to occur is called the "FSMO role owner" for that operation. When a new FSMO role owner is selected the replication system handles synchronous transfer of FSMO role ownership and the data the FSMO protects.

full replica A read and write replica of a directory partition that contains all attributes of all objects in the partition. Every domain controller has three full replicas: domain, schema, and configuration directory partitions. A full replica is also called a master replica. See also partial replica.

G

garbage collection interval A measurement of time indicating how often a domain controller examines its database for expired tombstones that can be collected.

Global Catalog A domain controller that contains a partial replica of every domain directory partition in the forest as well as a full replica of its own domain directory partition and the schema and configuration directory partitions. The Global Catalog holds a replica of every object in Active Directory, but each object includes a limited number of its attributes. The attributes in the Global Catalog are those most frequently used in search operations (such as a user's first and last names) and those attributes that are required to locate a full replica of the object. The Global Catalog enables users and applications to find objects in Active Directory given one or more attributes of the target object, without knowing what domain holds the object. The Active Directory replication system builds the Global Catalog automatically. The attributes replicated into the Global Catalog include a base set defined by Microsoft. Administrators can specify additional properties to meet the needs of their installation.

global group For Windows 2000 Server, a group that can be used in its own domain, in member servers and in workstations of the domain, and in trusting domains. In all those places a global group can be granted rights and permissions and can become a member of local groups. However, a global group can contain user accounts only from its own domain. See also group; local group.

governsID The object identifier that uniquely identifies the classSchema objects.

group A collection of users, computers, contacts, and other groups. Groups can be used as security or as e-mail distribution collections. Distribution groups are used only for e-mail. Security groups are used both to grant access to resources and as e-mail distribution lists. In a server cluster, a group is a collection of resources, and the basic unit of failover. See also domain local group; global group; native mode; universal group.

Group Policy An administrator's tool for defining and controlling how programs, network resources, and the operating system operate for users and computers in an organization. In an Active Directory environment, Group Policy is applied to users or computers on the basis of their membership in sites, domains, or organizational units.

Group Policy object A collection of Group Policy settings. Group Policy objects are the documents created by the Group Policy snap-in. Group Policy objects are stored at the domain level, and they affect users and computers contained in sites, domains, and organizational units. Each Windows 2000-based computer has exactly one group of settings stored locally, called the local Group Policy object.

Group Policy Security Settings The subtrees of the Group Policy console that allow a security administrator to manually configure security levels assigned to a Group Policy object or local computer policy.

H

hash See message digest; message digest function.

hash function See message digest; message digest function.

hashing algorithm See message digest; message digest function.

heartbeat In a server cluster or Network Load Balancing cluster, a periodic message sent between nodes to detect system failure of any node.

hierarchical namespace A namespace, such as the DNS namespace or Active Directory namespace, that is hierarchically structured and provides rules that allow the namespace to be partitioned. See also namespace; flat namespace; noncontiguous namespace.

high availability The ability to keep an application or service operational and usable by clients most of the time.

host A Windows 2000 computer that runs a server program or service used by network or remote clients. For Network Load Balancing, a cluster consists of multiple hosts connected over a local area network.

I

impersonation A circumstance that occurs when Windows NT or Windows 2000 allows one process to take on the security attributes of another.

inactive cluster member In a server cluster, a node that is not running.

index key A sequence of attributes from a database table, whose value uniquely identifies each row in the table. Also called a key segment.

infrastructure master The domain controller holding the infrastructure master role for the group's domain that is responsible for updating the cross-domain group-to-user reference to reflect the user's new name. The infrastructure master updates these references locally and uses replication to bring all other replicas of the domain up–to–date. If the infrastructure master is unavailable, these updates are delayed. See also Active Directory; domain controller; multimaster replication; operations master; replication.

inheritance The ability to build new object classes from existing object classes. The new object is defined as a subclass of the original object. The original object becomes a superclass of the new object. A subclass inherits the attributes of the superclass, including structure rules and content rules.

input/output (I/O) port A channel through which data is transferred between a device and the microprocessor. The port appears to the microprocessor as one or more memory addresses that it can use to send or receive data.

integrity A basic security function of cryptography. Integrity provides verification that the original contents of information have not been altered or corrupted. Without integrity, someone might alter information or the information might become corrupted, but the alteration can go undetected. For example, an Internet Protocol security property that protects data from unauthorized modification in transit, ensuring that the data received is exactly the same as the data sent. Hash functions sign each packet with a cryptographic checksum, which the receiving computer checks before opening the packet. If the packet-and therefore signature-has changed, the packet is discarded. See also cryptography; authentication; confidentiality; nonrepudiation.

IntelliMirror A set of Windows 2000 features used for desktop change and configuration management. When IntelliMirror is used in both the server and client, the users' data, applications, and settings follow them when they move to another computer.

Internet Control Message Protocol (ICMP)
A required maintenance protocol in the TCP/IP suite that reports errors and allows simple connectivity. ICMP is used by the Ping tool to perform TCP/IP troubleshooting.

Internet Protocol security (IPSec)
A set of industry-standard, cryptography-based protection services and protocols. IPSec protects all protocols in the TCP/IP protocol suite and Internet communications using L2TP. See also Layer Two Tunneling Protocol (L2TP).

interrupt request (IRQ) lines Hardware lines over which devices can send signals to get the attention of the processor when the device is ready to accept or send information. Interrupt request (IRQ) lines are numbered from 0 to 15. Each device must have a unique IRQ line.

IP/DNS-compatible Locator See domain controller locator.

IPSec See Internet Protocol security.

isMemberOfPartialAttributeSet A Boolean value that defines whether the attribute is replicated to the Global Catalog (in the Global Catalog has a value of TRUE, not in the Global Catalog is FALSE).

isSingleValued A Boolean value that specifies whether the attribute is single-valued (TRUE) or multivalued (FALSE). Default is FALSE if this value is not set.

K

Kerberos authentication protocol
An authentication mechanism used to verify user or host identity. The Kerberos v5 authentication protocol is the default authentication service for Windows 2000. Internet Protocol security and the QoS Admission Control Service use the Kerberos protocol for authentication. See also Internet Protocol security (IPSec); NTLM authentication protocol; QoS Admission Control Service.

kernel mode A highly privileged mode of operation where program code has direct access to all memory, including the address spaces of all user-mode processes and applications, and to hardware. Kernel mode is also known as supervisor mode, protected mode, or Ring 0.

key attack See key search attack.

Key Distribution Center (KDC) A network service that supplies session tickets and temporary session keys used in the Kerberos authentication protocol. In Windows 2000, the KDC runs as a privileged process on all domain controllers. The KDC uses Active Directory to manage sensitive account information such as passwords for user accounts. See also Kerberos authentication protocol; session ticket.

key exchange Confidential exchange of secret keys online, which is commonly done with public key cryptography. See also public key cryptography.

key management Secure management of private keys for public key cryptography. Windows 2000 manages private keys and keeps them confidential with CryptoAPI and CSPs. See also private key; CryptoAPI; cryptographic service provider.

key management server (KM server)
A secure mail management service for Microsoft Exchange Service.

key pair A private key and its related public key. See also public/private key pair.

key search attack An attack to find a secret password or a symmetric encryption key by trying all possible passwords or keys until the correct password or key is discovered. Also called a brute force attack.

Knowledge Consistency Checker (KCC)
A built-in process that runs on all domain controllers and generates the replication topology for the Active Directory forest. At specified intervals, the KCC reviews and makes modifications to the replication topology to ensure propagation of data either directly or transitively.

knowledge reference In Active Directory, knowledge about the existence and location of directory partitions in the forest, including the names of the directory partitions and what server is holding read-only copies (partial directory partitions stored on Global Catalogs) and/or writable copies (full directory partitions). See also external reference.

L

LAN See local area network.

latency See replication latency.

Layer two Tunneling Protocol (L2TP)
A tunneling protocol that encapsulates PPP frames to be sent over IP, X.25, Frame Relay, or ATM networks. L2TP is a combination of the Point-to-Point Tunneling Protocol (PPTP) and Layer 2 Forwarding (L2F), a technology proposed by Cisco Systems, Inc.

LDAP See Lightweight Directory Access Protocol.

LDAP API See Lightweight Directory Access Protocol Application Programming Interface.

LDAP Data Interchange Format (LDIF)
A draft Internet standard for a file format that can be used to perform batch operations on directories that conform to LDAP standards.

LDAP referral A reference to another domain controller, returned by an LDAP search when the requested object is not found on the domain controller being searched.

lDAPDisplayName (LDAP-Display-Name)
The name by which LDAP clients identify an attribute. The lDAPDisplayName property must be unique across all lDAPDisplayName attributes for all schemaClass and schemaAttribute objects in the Schema container.

It is recommended that the lDAPDisplayName be the cn with the hyphens removed (except do not remove the hyphen separating your name prefix from the rest of the name) and the first character uses the lower–case. The lDAPDisplayName property must be unique across all lDAPDisplayName attributes for all schemaClass and schemaAttribute objects in the schema container.

It is also recommended that you specify the lDAPDisplayName rather than letting this attribute default.

LDIF Directory Exchange tool A command–line utility that allows you to import and export objects to and from Active Directory. You can create, modify, and delete directory objects by using this utility. Objects are stored in the LDIF file format. The utility can be run on a Windows 2000 server or copied to a Windows 2000 workstation. For example, LDIFDE 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.

LDIFDE See LDIF Directory Exchange tool.

Lightweight Directory Access Protocol (LDAP) A directory service protocol that runs directly over TCP/IP and the primary access protocol for Active Directory. LDAP version 3 is defined by a set of Proposed Standard documents in Internet Engineering Task Force (IETF) RFC 2251. See also Lightweight Directory Access Protocol application programming interface (LDAP API).

Lightweight Directory Access Protocol application programming interface (LDAP API) An API for experienced C programmers who want to enable new or existing applications to connect to, search, and update LDAP servers. You can use the LDAP API to write directory-enabled applications that allow LDAP client applications to search for and retrieve information from an LDAP server. LDAP API enables the modification of directory objects, where such modifications are permitted. There are also functions that provide access control for servers, by allowing clients to authenticate themselves.

The LDAP API is delivered with Windows 2000 and is found in the Wldap32.dll file. The Microsoft LDAP API is compatible with both version 2 and version 3 of the LDAP standard.

linkID An integer that indicates that the attribute is a linked attribute. An even integer is a forward link and an odd integer is a back link.

This value must be unique for linkIDs of all attributeSchema objects. A back link must have corresponding forward link.

local area network (LAN) A communications network connecting a group of computers, printers, and other devices located within a relatively limited area (for example, a building). A LAN allows any connected device to interact with any other on the network. See also wide area network (WAN).

local group For computers running Windows 2000 Professional and member servers, a group that is granted permissions and rights from its own computer to only those resources on its own computer on which the group resides. See also global group.

local security authority (LSA) A protected subsystem that authenticates and logs users onto the local system. In addition, the LSA maintains information about all aspects of local security on a system (collectively known as the local security policy), and provides various services for translation between names and identifiers.

local security policy Security information about all aspects of local security on a system. The local security policy identifies who is assigned privileges and what security auditing is to be performed.

logical printer The software interface between the operating system and the printer in Windows 2000. While a printer is the device that does the actual printing, a logical printer is its software interface on the print server. This software interface determines how a print job is processed and how it is routed to its destination (to a local or network port, to a file, or to a remote print share). When a document is printed, it is spooled (or stored) on the logical printer before it is sent to the printer itself.

logical store See certificate stores.

loose consistency In multimaster directory replication, the tolerance for replication latency. In Active Directory replication, 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. Factors that affect replication latency include same or different site, number of hops between domain controllers, whether changes are found via notification or periodic synchronization, bandwidth of links, whether systems are down, and replication load. See also replication convergence.

M

mandatory attributes Object attributes for which values must be specified.

master replica See full replica.

mayContain A multivalued property that specifies the attributes that can be present on instances of this class. These are optional attributes that are not mandatory and, therefore, may or may not be present on an instance of this classSchema object. For an existing classSchema object, values can be added to this property but not removed. Each value is the lDAPDisplayName of an attribute. You must ensure that the attributes exist or will exist when the new class is written to the directory. If one of the attributes does not exist, the classSchema object will fail to be added to the directory.

The full set of optional attributes for this class is the union of the systemMayContain and mayContain on this class as well as the systemMayContain and mayContain properties of all inherited classes.

member server A computer that runs Windows 2000 Server but is not a domain controller of a Windows 2000 domain. Member servers participate in a domain, but do not store a copy of the directory database.

memory address A portion of computer memory that can be allocated to a device or used by a program or the operating system. Devices are usually allocated a range of memory addresses.

message digest A fixed-size result obtained by applying a one-way mathematical function called a message digest function (sometimes called a "hash function" or "hash algorithm") to an arbitrary amount of data. Given a change in the input data, the resulting value of the message digest will change. Message digest is also called a hash. See message digest function.

message digest function One-way mathematical algorithm used to produce a message digest (also called a hash). See also message digest.

Messaging Application Programming Interface (MAPI)
A Microsoft API used to support messaging applications.

Microsoft Management Console (MMC) A framework for hosting administrative consoles. A console is defined by the items on its console tree, which might include folders or other containers, World Wide Web pages, and other administrative items. A console has one or more windows that can provide views of the console tree and the administrative properties, services, and events that are acted on by the items in the console tree. The main MMC window provides commands and tools for authoring consoles. The authoring features of MMC and the console tree might be hidden when a console is in User Mode. See also console tree.

mixed mode 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 co-exist 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 after all Windows NT domain controllers are either removed from the domain or upgraded to Windows 2000. See also native mode.

mixed version Used to describe a server cluster in which different nodes are running different versions of the server cluster software.

MMC snap-in A type of management tool that you can add to the console tree of a console supported by Microsoft Management Console (MMC), for example, Device Manager. A snap-in can be either a stand-alone or an extension snap-in. A stand-alone snap-in can be added by itself; an extension snap-in can only be added to extend another snap-in. See also Microsoft Management Console (MMC).

multimaster replication A system of replication in which all replicas of a given directory partition are writable, allowing updates to be applied to any replica. Active Directory uses a multimaster replication system, and replicates the changes from a given replica to all other replicas automatically and transparently. All replicas are not necessarily fully consistent at all times. This model differs from other replication models in which one computer stores the single modifiable copy of the directory and other computers store backup copies. See also domain controller; replication; loose consistency.

Multipurpose Internet Mail Extensions (MIME) A common method for transmitting non-text data through Internet e-mail. MIME encodes non-text data as ASCII text and then decodes it back to its original format at the receiving end. A MIME header is added to the file which includes the type of data contained and the encoding method used. See also Secure/Multipurpose Internet Mail Extensions (S/MIME).

multithreading The ability of an operating system to run several processes at the same time to handle multiple tasks.

mustContain A multi-valued property that specifies the attributes that must be present on instances of this class. These are mandatory attributes that must be present during creation and cannot be cleared after creation. After creation of the class, this attribute cannot be changed.

Each value is the lDAPDisplayName of an attribute. You must ensure that the attributes exist or will exist when the new class is written to the directory. If one of the attributes does not exist, the classSchema object will fail to be added to the directory.

The full set of mandatory attributes for this class is the union of the systemMustContain and mustContain on this class as well as the systemMustContain and mustContain properties of all inherited classes.

N

name resolution The process of having software translate between names that are easy for users to work with, and numerical IP addresses, which are difficult for users but necessary for TCP/IP communications. Name resolution can be provided by software components such as the Domain Name System (DNS) or the Windows Internet Name Service (WINS). In directory service, the phase of LDAP directory operation processing that involves finding a domain controller that holds the target entry for the operation. See also Domain Name System (DNS); Transmission Control Protocol/Internet Protocol (TCP/IP); Windows Internet Name Service (WINS).

name server In the DNS client/server model, a server authoritative for a portion of the DNS database. The server makes computer names and other information available to client resolvers that are querying for name resolution across the Internet or an intranet. See also Domain Name System (DNS).

namespace A set of unique names for resources or items used in a shared computing environment. The names in a namespace can be resolved to the objects they represent. For Microsoft Management Console (MMC), the namespace is represented by the console tree, which displays all of the snap-ins and resources that are accessible to a console. For Domain Name System (DNS), namespace is the vertical or hierarchical structure of the domain name tree. For example, each domain label, such as "host1" or "example," used in a fully qualified domain name, such as "host1.example.microsoft.com," indicates a branch in the domain namespace tree. For Active Directory, namespace corresponds to the DNS namespace in structure, but resolves Active Directory object names.

naming context See directory partition.

National Registration Authority (NRA) An identified body in each nation responsible for issuing object identifiers to enterprises.

native mode The condition in which all domain controllers within a domain are Windows 2000 domain controllers and an administrator has enabled native mode operation (through Active Directory Users and Computers). See also mixed mode.

Net Logon service A service that runs in the Windows 2000 security subsystem in user mode, and performs the following functions; Replication of Windows NT 3.*x* and Windows NT 4.0 backup domain controllers with the Windows 2000 PDC emulator; NTLM pass-through authentication; Periodic password updates for computer accounts and interdomain trust relationships; Domain controller discovery using NetBIOS naming for non-directory-aware domain controllers (domain controllers that run Windows NT 3.5 and Windows NT 4.0); Domain controller discovery in closest site using NetBIOS naming or DNS naming for directory-aware domain controllers (Windows 2000 domain controllers).

network adapter Software or a hardware plug-in board that connects a node or host to a local area network.

Network Load Balancing Cluster Up to 32 IIS servers from which Network Load Balancing presents a single IP address to Web clients and among which Network Load Balancing distributes incoming Web requests.

network name In server clusters, the name through which clients access server cluster resources. A network name is similar to a computer name, and when combined in a resource group with an IP address and the applications clients access, presents a virtual server to clients.

node In tree structures, a location on the tree that can have links to one or more items below it. In local area networks (LANs), a device that is connected to the network and is capable of communicating with other network devices. In a server cluster, a server that has Cluster service software installed and is a member of the cluster. See also local area network (LAN).

non-authoritative restore The default restore mode when using the Windows 2000 Ntbackup utility. When a domain controller is restored from a backup tape, the domain controller is brought up-to-date with its replica partners using normal Active Directory replication protocols. It is a non-authoritative restore because the objects in the restored directory are not treated as authoritative. The restored objects are replaced with changes held in other replicas of the restored domain.

nonce A randomly generated value used to defeat replay attacks. See also replay attack.

noncontiguous namespace A namespace based on different DNS root domain names, such as that of multiple trees in the same forest. See also namespace; hierarchical namespace; flat namespace.

nonpaged pool An area of system memory reserved for objects that must remain in physical memory as long as they are active. The alternative is the paged pool.

nonrepudiation A basic security function of cryptography. Nonrepudiation provides assurance that a party in a communication cannot falsely deny that a part of the communication occurred. Without nonrepudiation, someone can communicate and then later deny the communication or claim that the communication occurred at a different time. See also cryptography; authentication; confidentiality; integrity.

nontransitive trust relationship A type of trust relationship that is bounded by the two domains in the relationship. For example, if domain A trusts domain B and domain B trusts domain C, there is no trust relationship between domain A and domain C. A nontransitive trust relationship can be a one-way or two-way relationship. It is the only type of trust relationship that can exist between a Windows 2000 domain and a Windows NT domain or between Windows 2000 domains in different forests. See also trust relationship; transitive trust relationship.

NTLM authentication protocol A challenge/response authentication protocol. The Windows NT LAN Manager (NTLM) protocol was the default for network authentication in Windows NT version 4.0 and earlier. The protocol continues to be supported in Windows 2000 but no longer is the default. See also authentication.

O

object An entity, such as a file, folder, shared folder, printer, or Active Directory object, described by a distinct, named set of attributes. For example, the attributes of a File object include its name, location, and size; the attributes of an Active Directory User object might include the user's first name, last name, and e-mail address. For OLE and ActiveX objects, an object can also be any piece of information that can be linked to, or embedded into, another object. See also attribute; child object; container object; parent object.

object class The object class is the formal definition of a specific kind of object that can be stored in the directory. An object class is a distinct, named set of attributes that represents something concrete, such as a user, a printer, or an application. The attributes hold data describing the thing that is identified by the directory object. Attributes of a user might include the user's first name, last name, and e-mail address. The terms object class and class are used interchangeably. The attributes that can be used to describe an object are determined by the content rules.

objectClassCategory An integer value that specifies the category of the class. The category can be Structural, Abstract, or Auxiliary.

offline In a server cluster, the state of a resource, group, or node when it is unavailable to the cluster. Resources and groups also have an offline state. See also group; node; online, paused; resource.

oMObjectClass For object-syntaxed attributes (OM-syntax = 127), a binary value that describes the type of object.

oMSyntax The OM syntax of an attribute. Syntax of this attribute as defined by the XAPIA X/Open Object Model (XOM) specification.

one-level search See search scope.

online In a server cluster, the state of a resource, group, or node when it is available to the cluster. See also heartbeat; node; offline; paused; resource.

operational attribute An attribute that is used only for administering the directory database. It is an artifact attribute that is never defined in the schema and does not require any storage. Generally, when you set the operational attribute, you trigger some immediate action on the server.

operations master A domain controller that has been assigned one or more special roles in an Active Directory domain. The domain controllers assigned these roles perform operations that are single-master (not permitted to occur at different places in the network at the same time). Examples of these operations include resource identifier allocation, schema modification, primary domain controller election and certain infrastructure changes. The domain controller that controls the particular operation owns the operations master role for that operation. The ownership of these operations master roles can be transferred to other domain controllers. See also Active Directory; domain naming master; infrastructure master; multimaster replication; relative ID master; replication.

organizational unit (OU) An Active Directory container object used within domains. An organizational unit is a logical container into which users, groups, computers, and other organizational units are placed. It can contain objects only from its parent domain. An organizational unit is the smallest scope to which a Group Policy object can be linked, or over which administrative authority can be delegated.

originating update In Active Directory replication, a write to a property at the system initiating the change, as opposed to a write to a property that is caused by replication. For example, a change to attribute A on object O on server S1 is the originating write. When S2 receives the change from S1, the local database update on S2 for object O, attribute A is the corresponding replicated write. See also replicated update.

OSChooser Markup Language (OSCML) The markup language used to modify the Client Installation Wizard (CIW) .osc files for the screens presented when using Remote Installation Service.

P

parent class All structural object classes are subclasses, directly or indirectly, of a single abstract object class, which is called top. Every object represented in the directory belongs to top and, as a result, every entry must have an objectClass attribute. When you create a new class, you must specify the superclass. If you are not creating a subclass of an existing class, the new class is a subclass of top. The parent object becomes a superclass of the new object. Superclass is a classSchema object from which one or more other classSchema objects inherit information. The inherited information includes mandatory and optional attributes (systemMust-Contain, mustContain, systemMayContain, and mayContain) and its parent classes in the directory hierarchy (systemPossSuperiors and possSuperiors).

parent domain For DNS and Active Directory, domains that are located in the namespace tree directly above other derivative domain names (child domains). For example, "reskit.com" would be the parent domain for "eu.reskit.com," a child domain. See also child domain; directory partition; domain.

parent object The object that is the immediate superior of another object in a hierarchy. A parent object can have multiple subordinate, or child, objects. In Active Directory, the schema determines what objects can be parent objects of what other objects. Depending on its class, a parent object can be the child of another object. See also child object; object.

parent-child trust relationship The two-way, transitive trust relationship that is established when a domain is added to an Active Directory tree. The Active Directory installation process automatically creates a trust relationship between the domain you are creating (the new child domain) and the parent domain.

parenting The concept of managing the growth and delegation of a parent domain into further child domains, which are derived and delegated from the parent name. See also child domain; parent domain.

partial replica A read-only replica of a directory partition that contains a subset of the attributes of all objects in the partition. Each Global Catalog contains partial replicas of all domains in the forest. The attributes contained in a partial replica are defined in the schema as the attributes whose attributeSchema objects have the isMemberOfPartialAttributeSet attribute set to TRUE. See also full replica; Global Catalog.

partition knowledge table (PKT) Repository of information about the Dfs topology and its mappings to the underlying physical shares. For a domain-based Dfs root, the PKT is stored in Active Directory and made available to each server that hosts a domain-based Dfs root. For a stand-alone Dfs root, the PKT is stored in the individual server's registry.

paused The state of a node that is a fully active member in the server cluster but cannot host groups. The paused state is provided for an administrator to perform maintenance. See also; failback; failover; node; offline.

PC/SC smart card specification An open standard for smart cards and smart card readers published by the PC/SC Workgroup, a consortium of industry-leading computer software and hardware manufacturers.

permission A rule associated with an object to regulate which users can gain access to the object and in what manner. Permissions are granted or denied by the object's owner. See also object; privilege; user rights.

personal identification number (PIN) A secret identification code that is used to protect smart cards from misuse. The PIN is similar to a password and is known only to the owner of the card. The smart card can be used only by someone who possesses the smart card and knows the PIN. See also smart card.

physical stores See certificate stores.

Ping A tool that verifies connections to one or more remote hosts. The ping command uses the ICMP Echo Request and Echo Reply packets to determine whether a particular IP system on a network is functional. Ping is useful for diagnosing IP network or router failures. See also Internet Control Message Protocol (ICMP).

plaintext Data that is not encrypted. Sometimes also called clear text. See also ciphertext; encryption; decryption.

possible owner A node on which a resource can operate, and which has been added to the resource's list of possible owners. Resources fail over only to possible owners.

possSuperiors A multivalued property that specifies the classes that can be legal parents of instances of this class. For an existing classSchema object, values can be added to this property but not removed.

Each value is the lDAPDisplayName of a class. You must ensure that the classes exist or will exist when the new class is written to the directory. If one of the classes does not exist, the classSchema object will fail to be added to the directory.

The full set of possible superiors is the union of the systemPossSuperiors and possSuperiors on this class as well as the systemPossSuperiors and possSuperiors properties of all inherited superclasses (structural or abstract classes). Note that possSuperiors are not inherited from auxiliary classes.

primary domain controller emulator master The domain controller assigned to act as a Windows NT primary domain controller (PDC) to service network clients that do not have Active Directory client software installed, and to replicate directory changes to any Windows NT backup domain controllers (BDCs) in the domain. For a Windows 2000 domain operating in native mode, the PDC emulator master receives preferential replication of password changes performed by other domain controllers in the domain and handles any password authentication requests that fail at the local domain controller. At any time, there can be only one PDC emulator in a particular domain. See also Active Directory; domain controller; multimaster replication; operations master; replication.

private key The secret half of a cryptographic key pair that is used with a public key algorithm. Private keys are typically used to digitally sign data and to decrypt data that has been encrypted with the corresponding public key. See also public key.

privilege A user's right to perform a specific task, usually one that affects an entire computer system rather than a particular object. Privileges are assigned by administrators to individual users or groups of users as part of the security settings for the computer. See also permission; user rights.

process An operating system object that consists of an executable program, a set of virtual memory addresses, and one or more threads. When a program runs, a Windows 2000 process is created. See also thread.

protected storage service A service that provides applications with a place to store per-user data that must be kept secret or free from modification. Protected storage uses the Hash-based Message Authentication Code (HMAC) and the SHA1 cryptographic hash function to encrypt the user's master key.

pruning A process that removes unavailable printers from Active Directory listing. An orphan pruner program running on the domain controller periodically checks for orphaned printers, that is, printers that are offline or powered down, and deletes the printer objects of the printers it cannot find.

public key The non-secret half of a cryptographic key pair that is used with a public key algorithm. Public keys are typically used to verify digital signatures or decrypt data that has been encrypted with the corresponding private key. See also private key.

public key certificate A digital passport that serves as proof of identity. Public key certificates are issued by a certification authority (CA). See also certification authority (CA); Kerberos authentication protocol.

public key cryptography A method of cryptography in which two different but complimentary keys are used: a public key and a private key for providing security functions. Public key cryptography is also called asymmetric key cryptography. See also cryptography; public key; private key.

public key cryptography standards (PKCS) A family of standards for public key cryptography that includes RSA encryption, Diffie-Hellman key agreement, password-based encryption, extended-syntax, cryptographic message syntax, private key information syntax, and certificate request syntax, as well as selected attributes. Developed, owned and maintained by RSA Data Security, Inc. See also certificate; public key.

public key infrastructure (PKI) The term generally used to describe the laws, policies, standards, and software that regulate or manipulate certificates and public and private keys. In practice, it is a system of digital certificates, certification authorities, and other registration authorities that verify and authenticate the validity of each party involved in an electronic transaction. Standards for PKI are still evolving, even though they are being widely implemented as a necessary element of electronic commerce. See also certificate; certification authority; public key cryptography.

public-key algorithm An asymmetric cipher that uses two keys, one for encryption, the public key, and the other for decryption, the private key. See also public key encryption; public key; private key; symmetric-key algorithm.

public-key encryption A method of encryption that uses two encryption keys that are mathematically related. One key is called the private key and is kept confidential. The other is called the public key and is freely given out to all potential correspondents. In a typical scenario, a sender uses the receiver's public key to encrypt a message. Only the receiver has the related private key to decrypt the message. The complexity of the relationship between the public key and the private key means that, provided the keys are long enough, it is computationally infeasible to determine one from the other. Public key encryption is also called asymmetric encryption. See also public key; symmetric key encryption.

public/private key pair A set of cryptographic keys used for public key cryptography. One key is used to encrypt, the other to decrypt. See also public key; private key.

published applications An application that is available to users managed by a Group Policy object. Each user decides whether or not to install the published application by using Add/Remove Programs in Control Panel.

Q

QoS Admission Control Service A software service that controls bandwidth and network resources on the subnet to which it is assigned. Important applications can be given more bandwidth, less important applications less bandwidth. The QoS Admission Control Service can be installed on any network-enabled computer running Windows 2000.

quorum disk The cluster disk on which configuration data is maintained in the quorum log, cluster database checkpoint, and resource checkpoints. The quorum disk is managed by the Quorum resource, which is usually a special kind of Physical Disk resource.

quorum log The record, stored on the quorum disk, of changes that have been made to the cluster database of the registry since the last cluster database checkpoint was taken. Also known as the recovery log or change log.

Quorum resource A quorum-capable resource (usually a Physical Disk resource) that has been configured to manage the quorum log and cluster database checkpoints, which comprise the configuration data necessary for recovery of the cluster.

quorum-capable resource In a server cluster, a resource that can act as the cluster's Quorum resource. To be quorum-capable, a resource must provide shared storage and a means of persistent arbitration. The Cluster service defines only Physical Disk resources as quorum-capable.

R

rangeLower Optional. An integer that specifies the lower value of the range of values for this attribute. All values set for this attribute must be greater than or equal to this value. If both rangeLower and rangeUpper are set, rangeLower must be less than rangeUpper. For integers, value means the value of the integer. For string syntaxes, value means the number of characters in the string. For octet strings, value means the number of bytes.

rangeUpper Optional. An integer that specifies the upper value of the range of values for this attribute. All values set for this attribute must be less than or equal to this value. If both rangeLower and rangeUpper are set, rangeLower must be less than rangeUpper. For integers, value means the value of the integer. For string syntaxes, value means the number of characters in the string. For octet strings, value means the number of bytes.

rDnAttId Specifies the ldapDisplayName of the attribute that will be the naming attribute for the new class – if different than the default ("cn").

Use of a naming attribute other than "cn" is discouraged. Naming attributes should be drawn from the well-known set (OU, CN, O, L and DC) that is understood by all LDAP version 3 clients.

recovery agent An account that can be used to decrypt a file encrypted by using the Encrypting File System (EFS) if the file owner's decryption key becomes unavailable.

recovery log See quorum log.

referral In Dfs, information that maps a DNS name in the logical namespace to the UNC equivalent name of a physical share. When a Dfs client gains access to a shared folder in the Dfs namespace, the Dfs root server returns a referral for the client to use in locating the shared folder. In DNS, a pointer to an authoritative DNS server that is authoritative for a lower level of the domain namespace. See also LDAP referral.

regroup event In a server cluster, when one node detects a communication failure with another cluster node, the first node's broadcast of a message to the entire cluster causing all members to verify their view of the current cluster membership.

relative distinguished name The part of an object name that is an attribute of the object itself. For example, given the distinguished name of cn=JamesSmith,ou=Managers,dc=Reskit,dc=com, the relative distinguished name of the JamesSmith user object is cn=James Smith. The relative distinguished name of the parent object is ou=Managers. The relative distinguished name of the domain object is dc=Reskit.

relative ID (RID) The part of a security ID (SID) that uniquely identifies an account or group within a domain. See also security ID.

relative ID master The domain controller assigned to allocate sequences of relative IDs to each domain controller in its domain. Whenever a domain controller creates a security principal (user, group, or computer object), the domain controller assigns the object a unique security ID. The security ID consists of a domain ID that is the same for all security IDs created in a particular domain, and a relative ID that is unique for each security ID created in the domain. At any time, there can be only one relative ID master in a particular domain. See also Active Directory; domain controller; multimaster replication; operations master; replication.

remote installation boot floppy (RBFG.exe) A tool that is used to generate a remote installation boot floppy disk. The remote installation boot floppy disk is used to start the process of remote operating system installation for computers which lack a supported PXE-based remote boot ROM.

Remote Installation Services (RIS) Software services that allow a user to install Windows 2000 Professional from a Remote Install server with minimal interaction.

remote operating system installation See Remote Installation Services (RIS).

replay attack An attempt to circumvent an authentication protocol by copying authentication messages from a legitimate client and then resending them during the impostor's own authentication to the server. See also nonce.

replica In Active Directory replication, a copy of a logical Active Directory partition that is synchronized through replication between domain controllers that hold copies of the same directory partition. "Replica" can also refer to the composite set of directory partitions held by any one domain controller. These are specifically called a directory partition replica and server replica, respectively. See also full replica; partial replica.

replicated update In Active Directory replication, a write to a property on one replica as the result of replication of an update that originated at another replica. See also originating update.

replication The process of copying data from a data store or file system to multiple computers that store the same data for the purpose of synchronizing the data. In Windows 2000, replication of the directory service occurs through Active Directory replication, and replication of the file system occurs through the File Replication service. See also Active Directory replication; ; Distributed file system; File Replication service.

replication convergence In multimaster replication, the guarantee that all replicas eventually converge on the same set of values if the system is allowed to reach a steady state, in which no new updates are occurring and all previous updates have been completely replicated. In a steady state, all replicas of a directory partition have the same objects, the same attributes, and the same values. See also Active Directory replication; loose consistency; multimaster replication.

replication cost A numeric setting on a site link object. The total cost of a replication path between two sites is the sum of the costs of the links on the least costly route. Higher cost numbers represent more expensive messages. When the Knowledge Consistency Checker selects a site to obtain a source for a given directory partition, it selects the site with the least cost.

replication latency In Active Directory replication, the delay between the time an update is applied to a given replica of a directory partition and the time it is applied to some other replica of the same directory partition. A server will receive changes no sooner than either

· It is notified of a change from its neighbor in the same site, or

· Its periodic replication timer expires.

Latency is sometimes referred to as propagation delay. See also multimaster replication.

replication partner A domain controller that acts as a replication source for a given domain controller. The Knowledge Consistency Checker determines which servers are best suited to replicate with each other, and generates the list of domain controllers that are candidates for replication partners from the list of domain controllers in the site on the basis of connectivity, history of successful replication, and matching of full and partial replicas. A domain controller has some number of direct replication partners with whom it replicates for a given directory partition. The other domain controllers in the site replicate transitively with this domain controller. See also store-and-forward replication.

replication topology In Active Directory replication, the set of connections that domain controllers use to replicate information among themselves, both within sites and between sites. The site topology is defined by site link objects. The connection topology is defined by connection objects. See also Active Directory replication; connection object; domain controller; site link.

replication transport The protocols that are used to transport replication data over the network. For intrasite replication, data is always transferred by using RPC over IP. For intersite replication, data is transferred using either RPC synchronous transport (RPC over IP) or intersite messaging asynchronous transport (SMTP over IP). The choice of transport is controlled by the transport object (IP or SMTP) within which the site links are created in Active Directory. Different transports have different characteristics that make them better suited to different environments.

resource Any part of a computer system or network, such as a disk drive, printer, or memory, that can be allotted to a program or a process while it is running. For Device Manager, any of four system components that control how the devices on a computer work. These four system resources are: interrupt request (IRQ) lines, direct memory access (DMA) channels, input/output (I/O) ports, and memory addresses. In a server cluster, an instance of a resource type; the Cluster service manages various physical or logical items as resources. See also direct memory access (DMA); input/output (IO) port; interrupt request (IRQ) lines; memory address.

Resource DLL A dynamic-link library that defines default properties and behavior for a specific type of resource. The resource DLL contains an implementation of the Server Cluster API for a specific type of resource and is loaded into the address space of its Resource Monitor. See also dynamic-link library; Resource Monitor.

Resource Monitor The server cluster component that manages communication between a node's Cluster service and one or more of its resources. See also node; resource.

resource type A server cluster object used to manage resources of similar characteristics. A resource type is associated with a resource DLL that manages all the resources of that type in the cluster.

revision level One of three levels that can be viewed in Network Monitor traces which refer to Dfs client compatibility. Windows NT 4.0, Windows 95 and Windows 98 clients support Dfs revision level 2; Windows 2000 clients support revision level 3. There are no known version 1 clients. Dfs clients and servers negotiate the highest common protocol revision supported.

Rivest-Shamir-Adleman (RSA) cryptographic algorithms
A widely used set of public key algorithms that are available from RSA Data Security, Inc. The RSA cryptographic algorithms are supported by the Microsoft Base Cryptographic Service Provider and the Microsoft Enhanced Cryptographic Service Provider.

rolling upgrade In a cluster, the process of upgrading cluster nodes by turns while the other nodes continue to provide service.

root The highest or uppermost level in a hierarchically organized set of information. The root is the point from which further subsets are branched in a logical sequence that moves from a broad or general focus to narrower perspectives.

root authority See root certification authority.

root certificate A self-signed certification authority certificate. It is called a root certificate because it is the certificate for the root authority. The root authority must sign its own certificate because there is no higher certifying authority in the certification hierarchy. See also certificate; certification authority; root certification authority.

root certification authority The most trusted certification authority (CA), which is at the top of a certification hierarchy. The root CA has a self-signed certificate. Also called the root authority. See also certification authority; certification path; root certificate.

RSA cryptographic algorithms See Rivest-Shamir-Adleman (RSA) cryptographic algorithms.

S

scalability A measure of how well a computer, service, or application can expand to meet increasing performance demands. For server clusters, the ability to incrementally add one or more systems to an existing cluster when the overall load of the cluster exceeds its capabilities.

schema The universe of objects that can be stored in the directory is defined in the schema. For each object class, the schema defines what attributes an instance of the class must have, what additional attributes it may have, and what object class can be a parent of the current object class.

The Active Directory schema is implemented as a set of object class instances stored in the directory. This is very different than many directories which have a schema but store it as a text file read at startup. Storing the schema in the directory has many advantages. For example, user applications can read it to discover what objects and properties are available.

The Active Directory schema can be updated dynamically. That is, an application can extend the schema with new attributes and classes and use the extensions immediately. Schema updates are accomplished by creating or modifying the schema objects stored in the directory. Like every object in the Active Directory, schema objects are protected by ACLs, so only authorized users may alter the schema.

schema cache All changes made to Active Directory are validated first against the schema. For performance reasons, this validation takes place against a version of the schema that is held in memory on the domain controllers. This "in-memory version," called the schema cache, is updated automatically after the on-disk version has been updated. The schema cache provides mapping between attribute identifiers, such as a database column identifier or a MAPI identifier, and the in-memory structures that describe those attributes. The schema cache also provides lookups for class identifiers to get in-memory structures describing those classes.

schema master role The domain controller that holds the schema master role is the only domain controller that can perform write operations to the directory schema. Those schema updates are replicated from the schema master to all other domain controllers in the forest.

schemaIDGUID A GUID that uniquely identifies the attribute. It is recommended that you generate your own GUID for each attribute so that all installations of your schema extension use the same schemaIDGUID to refer to the attribute. If no value is specified, Active Directory generates a GUID.

search base In an LDAP search, the distinguished name of the search base object, which defines the location in the directory from which to begin searching.

search filter An argument in an LDAP search that allows certain entries in the subtree and excludes others. Filters allow you to define search criteria and give you better control to achieve more effective and efficient searches.

search scope Defines how deep to search within the search base.–Base, or zero levels, searches the base object only (a read of that object).–One level searches objects immediately subordinate to the base object, but not including the base object itself.–Subtree searches the entire subtree of which the base DN is the topmost object, including that base object. Also called a deep search.

searchFlags An integer value that contains bit flags. The attribute is indexed if the least significant bit is set to 1, or non-indexed if the bit is zero.

The searchFlags property of each property's attributeSchema object defines whether a property is indexed (indexed has a value of 1; nonindexed is 0).

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 (should be used in conjunction with 1);

8 = Preserve this attribute on logical deletion (that is, make this attribute available on tombstones).

secret key An encryption key that two parties share with each other and with no one else. See also symmetric key encryption.

secure electronic transaction (SET)
A standard protocol that is used for securing online credit card payments that are made over the Internet.

Secure/Multipurpose Internet Mail Extensions (S/MIME)
An extension of MIME to support secure mail. It enables message originators to digitally sign e-mail messages to provide proof of message origin and data integrity. It also enables messages to be transmitted in encrypted format to provide confidential communications. See also Multipurpose Internet Mail Extensions (MIME).

Security Accounts Manager (SAM)
A protected subsystem that manages user and group account information. In Windows NT 4.0, both local and domain security principals are stored by SAM in the registry. In Windows 2000, workstation security accounts are stored by SAM in the local computer registry, and domain controller security accounts are stored in Active Directory.

security descriptor A data structure that contains security information associated with a protected object. Security descriptors include information about who owns the object, who may access it and in what way, and what types of access will be audited. See also access control list; object.

security ID (SID) A data structure of variable length that uniquely identifies user, group, service, and computer accounts within an enterprise. Every account is issued a SID when the account is first created. Access control mechanisms in Windows 2000 identify security principals by SID rather than by name. See also relative ID; security principal.

security principal An account-holder, such as a user, computer, or service. Each security principal within a Windows 2000 domain is identified by a unique security ID (SID). When a security principal logs on to a computer running Windows 2000, the Local Security Authority (LSA) authenticates the security principal's account name and password. If the logon is successful, the system creates an access token. Every process executed on behalf of this security principal will have a copy of its access token. See also security ID; security principal name.

security principal name A name that uniquely identifies a user, group, or computer within a single domain. This name is not guaranteed to be unique across domains. See also security principal.

Security Reference Monitor A subsystem that is the primary authority for enforcing access control on a computer running Windows 2000 or Windows NT.

security subsystem See Local Security Authority (LSA).

security template A physical file representation of a security configuration that can be applied to a local computer or imported to a Group Policy object in Active Directory. When you import a security template to a Group Policy object, Group Policy processes the template and makes the corresponding changes to the members of that Group Policy object, which can be users or computers.

server A computer that provides shared resources to network users.

server cluster A cluster created and administered by the Cluster service and associated software (.exe and .dll files), between whose nodes the Cluster service provides failover support for applications running on the servers. The server cluster includes the hardware and the cluster configuration as well as the Cluster service. See also cluster; node.

Server Cluster API The collection of functions that are implemented by the Cluster service and used by cluster-aware applications, cluster management applications, and resource DLLs. The Server Cluster API includes functions for managing server cluster objects and the cluster database.

service ticket See session ticket.

session In the context of load balancing TCP/IP traffic, a set of client requests directed to a server. These requests can be invoked with multiple, possibly concurrent, TCP connections. The server program sometimes maintains state information between requests. To preserve access to the server state, Network Load Balancing needs to direct all requests within a session to the same cluster host when load balancing. See also client request; server; TCP/IP.

session key A key used primarily for encryption and decryption. Session keys are typically used with symmetric encryption algorithms where the same key is used for both encryption and decryption. For this reason, session and symmetric keys usually refer to the same type of key. See also symmetric key encryption.

session ticket A credential presented by a client to a service in the Kerberos authentication protocol. Because session tickets are used to obtain authenticated connections to services, they are sometimes called service tickets. See also Kerberos authentication protocol; Key Distribution Center (KDC).

shortcut trust A two-way trust relationship that is explicitly created between two Windows 2000 domains in the same forest. The purpose of a shortcut trust is to optimize the inter-domain authentication process by shortening the trust path. All shortcut trusts are transitive and must be created manually in each direction. See also domain tree; forest; transitive trust relationship.

site A location in a network that holds Active Directory servers. A site is defined as one or more well-connected TCP/IP subnets. ("Well-connected" means that network connectivity is highly reliable and fast—for example, LAN speeds of 10 MM bits-per-second or greater) Because computers in the same site are close to each other in network terms, communication among them is reliable, fast, and efficient. Defining a site as a set of subnets allows administrators to configure Active Directory access and replication topology to take advantage of the physical network. When users log on to the network, Active Directory clients find Active Directory servers in the same site as the client. In Systems Management Server, site servers and client computers bounded by a group of subnets, such as an IP subnet or an IPX network number. See also domain controller locator; subnet; replication topology.

site link An Active Directory object that represents a set of sites that can communicate at uniform cost through some intersite transport. For IP transport, a typical site link connects just two sites and corresponds to an actual WAN link. An IP site link connecting more than two sites might correspond to an ATM backbone connecting more than two clusters of buildings on a large campus, or several offices in a large metropolitan area connected via leased lines and IP routers. See also connection object; site link bridge.

site link bridge An Active Directory object that represents a set of site links, all of whose sites can communicate via some transport. Typically a site link bridge corresponds to a router (or a set of routers) in an IP network. By default, the Knowledge Consistency Checker may form a route through any and all site links in a transitive manner. If this behavior is turned off, each site link represents its own distinct and isolated network. Sets of site links that can be treated as a single route are expressed through a site link bridge. Each bridge represents an isolated communication environment for network traffic.

smart card A credit card-sized device that is used with a PIN number to enable certificate-based authentication and single sign-on to the enterprise. Smart cards securely store certificates, public and private keys, passwords, and other types of personal information. A smart card reader attached to the computer reads the smart card. See also authentication; certificate; nonrepudiation.

smart-card reader A device that is installed in computers to enable the use of smart cards for enhanced security features. See also smart card.

stand-alone certification authority A Windows 2000 certification authority that is not integrated with Active Directory. See also certification authority; enterprise certification authority.

stand-alone Dfs Implementation of Dfs that stores its configuration in the local registry. It is intended for backward compatibility with previous versions of Dfs. A stand-alone Dfs root has the following characteristics: it does not use Active Directory (or FRS file replication) and it cannot have replicas at the root level.

stand-alone server A computer that runs Windows 2000 Server but does not participate in a domain. A stand-alone server has only its own database of users, and it processes logon requests by itself. It does not share account information with any other computer and cannot provide access to domain accounts. See also member server; domain controller; global group; local group.

storage-class resource A required dependency for many resource types: a resource that manages a disk in the cluster that can be accessed using a drive letter. Windows 2000 Advanced Server provides one storage-class resource: Physical Disk. However, vendors or resellers may supply other storage-class resource types. See also resource type.

store-and-forward replication A replication model, used by Active Directory, in which changes are not sent directly from one domain controller to all other domain controllers. Instead, a system of replication partners is created automatically by the system, taking advantage of the existing connections. Replication through neighboring systems is also called transitive replication. See also Active Directory replication; multimaster replication.

structural classes The only classes that can have instances in the directory. That is, you can create directory objects whose class is one of the structural classes.

subclass A classSchema object that inherits from some other classSchema object. For example, a subclass inherits structure and content rules from the parent object.

subClassOf The class from which this object inherits attributes. For structural classes, the subClassOf can be a structural or abstract class. For abstract classes, the subClassOf can only be an abstract class. For auxiliary classes, the subClassOf can be an abstract or auxiliary class.

The value is the lDAPDisplayName of a class. You must ensure that the class exists or will exist when the new class is written to the directory. If class does not exist, the classSchema object will fail to be added to the directory.

subnet A subdivision of an IP network. Each subnet has its own unique subnetted network ID.

subordinate reference In Active Directory, knowledge of a partition or partitions directly below a partition held by a domain controller.

subordinate referral In an LDAP search, information about a directory location that is returned by a subtree search. If a subtree search has a search base that includes child directory partitions, the domain controller uses subordinate references to return a subordinate referral to a domain controller that stores the requested partition.

subtree search See search scope.

superclass The class from which a subclass derives all mandatory and optional attributes in addition to those specific to the class itself.

superior reference In Active Directory, knowledge about a referral location that is used when the domain controller has no knowledge of the search base.

symmetric key A single key that is used with symmetric encryption algorithms for both encryption and decryption. See also bulk encryption; encryption; decryption; session key.

symmetric key encryption An encryption algorithm that requires the same secret key to be used for both encryption and decryption. This is often called secret key encryption. Because of its speed, symmetric encryption is typically used rather than public key encryption when a message sender needs to encrypt large amounts of data.

symmetric-key algorithm A symmetric cipher that uses the same key for encryption and decryption. See also symmetric key encryption; symmetric key; public key algorithm.

symmetric-key cryptography A type of cryptography that uses symmetric keys to provide confidentiality. See also cryptography; symmetric-key encryption; symmetric-key algorithm.

system access control list (SACL) The part of an object's security descriptor that specifies which events are to be audited per user or group. Examples of auditing events are file access, logon attempts, and system shutdowns. See also access control entry (ACE); discretionary access control list (DACL); object; security descriptor.

System Key (SysKey) A tool provided with Windows 2000 to protect all symmetric cryptographic keys in a domain or organizational unit by encrypting them with a 128-bit random key.

System State A collection of system-specific data that can be backed up and restored. For all Windows 2000 operating systems, the System State data includes the registry, the class registration database, and the system boot files. For Windows 2000 Server, the system state data also includes the Certificate Services database (if the server is operating as a certificate server). If the server is a domain controller, the system state data also includes Active Directory and the Sysvol directory. See also Active Directory; domain controller; Sysvol.

system-Only attributes Attributes on which Windows 2000 and Active Directory depend for normal operations.

systemAuxiliaryClass A multivalued property that specifies the auxiliary classes from which a class inherits. After creation of the class, this property cannot be changed.

Each value is the lDAPDisplayName of a class. You must ensure that the classes exist, or will exist, when the new class is written to the directory. If one of the classes does not exist, the classSchema object will fail to be added to the directory.

The full set of auxiliary classes that this class inherits from is the union of the systemAuxiliaryClass and auxiliaryClass on this class as well as the systemAuxiliaryClass and auxiliaryClass properties of all inherited classes.

systemMayContain A multi-valued property that specifies the attributes that may be present on instances of this class. These are optional attributes that are not mandatory and, therefore, may or may not be present on an instance of this class. After creation of the class, this property cannot be changed.

Each value is the lDAPDisplayName of an attribute. You must ensure that the attributes exist or will exist when the new class is written to the directory. If one of the attributes does not exist, the classSchema object will fail to be added to the directory.

The full set of optional attributes for this class is the union of the systemMayContain and mayContain on this class as well as the systemMayContain and mayContain properties of all inherited classes.

systemMustContain A multivalued property that specifies the attributes that must be present on instances of this class. These are mandatory attributes that must be present during creation and cannot be cleared after creation. After creation of the class, this property cannot be changed.

Each value is the lDAPDisplayName of an attribute. You must ensure that the attributes exist or will exist when the new class is written to the directory. If one of the attributes does not exist, the classSchema object will fail to be added to the directory.

The full set of mandatory attributes for this class is the union of the systemMustContain and mustContain on this class as well as the systemMustContain and mustContain properties of all inherited classes.

systemPossSuperiors A multivalued property that specifies the structural classes that can be legal parents of instances of this class. After creation of the class, this property cannot be changed.

Each value is the lDAPDisplayName of a class. You must ensure that the classes exist or will exist when the new class is written to the directory. If one of the classes does not exist, the classSchema object will fail to be added to the directory.

The full set of possible superiors is the union of the systemPossSuperiors and possSuperiors on this class as well as the systemPossSuperiors and possSuperiors properties of all inherited superclasses (structural or abstract classes). Note that possSuperiors are not inherited from auxiliary classes.

Sysvol A shared directory that stores the server's copy of the domain's public files, which are replicated among all domain controllers in the domain. See also domain controller.

T

TCP/IP See Transmission Control Protocol/Internet Protocol.

thread A type of object within a process that runs program instructions. Using multiple threads allows concurrent operations within a process and enables one process to run different parts of its program on different processors simultaneously. A thread has its own set of registers, its own kernel stack, a thread environment block, and a user stack in the address space of its process.

ticket-granting ticket A credential issued to a user by the Key Distribution Center (KDC) when the user logs on. The user must present the TGT to the KDC when requesting session tickets for services. Because a TGT is normally valid for the life of the user's logon session, it is sometimes called a user ticket. See also Kerberos authentication protocol; Key Distribution Center; session ticket.

tombstone In Active Directory, an object that is removed from the directory but not yet deleted.

tombstone lifetime The length of time that an object lives as a tombstone in the directory before being collected as garbage.

transitive trust relationship The trust relationship that inherently exists between Windows 2000 domains in a domain tree or forest, or between trees in a forest, or between forests. When a domain joins an existing forest or domain tree, a transitive trust is automatically established. In Windows 2000 transitive trusts are always two-way relationships. See also domain tree; forest; nontransitive trust relationship.

Transmission Control Protocol/Internet Protocol (TCP/IP)
A set of software networking protocols widely used on the Internet that provide communications across interconnected networks of computers with diverse hardware architectures and operating systems. TCP/IP includes standards for how computers communicate and conventions for connecting networks and routing traffic.

Transport Layer Security (TLS) A standard protocol that is used to provide secure Web communications on the Internet or intranets. It enables clients to authenticate servers or, optionally, servers to authenticate clients. It also provides a secure channel by encrypting communications for confidentiality.

tree-root trust relationship The trust relationship that is established when you add a new tree to an Active Directory forest. Active Directory installation process automatically creates a transitive trust relationship between the domain you are creating (the new tree root) and the forest root.

trust path A series of trust links from one domain to another domain for passing authentication requests.

trust relationship A logical relationship established between domains that allows pass-through authentication in which a trusting domain honors the logon authentications of a trusted domain. User accounts and global groups defined in a trusted domain can be granted rights and permissions in a trusting domain, even though the user accounts or groups do not exist in the trusting domain's directory. See also authentication; domain.

U

Unattended Setup An automated, hands-free method of installing Windows 2000. During installation, Unattended Setup uses an answer file to supply data to Setup instead of requiring that an administrator interactively provide the answers.

universal group A Windows 2000 group only available in native mode that is valid anywhere in the forest. A universal group appears in the Global Catalog but contains primarily global groups from domains in the forest. This is the simplest form of group and can contain other universal groups, global groups, and users from anywhere in the forest. See also domain local group; forest; Global Catalog.

up-to-dateness vector In Active Directory replication, a value that the source domain controller uses to reduce the set of objects and attributes that it sends to the destination domain controller. The up-to-dateness vector is provided to the source domain controller by the destination domain controller and indicates the highest update sequence number (USN) of originating write that has been received for the relevant directory partition from the source domain controller.

update sequence number (USN) In Active Directory replication, a 64-bit counter that is maintained by each Active Directory domain controller. At the start of each update transaction (originating or replicated) on a domain controller, the domain controller increments its current USN and associates this new value with the update request.

user mode The processing mode in which applications run.

user rights Tasks a user is permitted to perform on a computer system or domain. There are two types of user rights: privileges and logon rights. An example of a privilege is the right to shut down the system. An example of a logon right is the right to log on to a computer locally (at the keyboard). Both types are assigned by administrators to individual users or groups as part of the security settings for the computer. See also permission; privilege.

user ticket See ticket-granting ticket.

V

virtual server In a server cluster, a set of resources, including a Network Name resource and an IP address resource, that is contained by a resource group. To clients, a virtual server presents the appearance of a system that is running Windows NT Server or Windows 2000 Server.

W

WAN See wide area network.

Web farm A Network Load Balancing cluster of IIS servers that support client Web site requests.

wide area network (WAN) A communications network connecting geographically separated computers, printers, and other devices. A WAN allows any connected device to interact with any other on the network. See also local area network (LAN).

Windows Installer (.msi files) The Windows Installer is an operating system service that allows the operating system to manage the installation process. Windows Installer technologies are divided into two parts that work in combination: a client–side installer service (MSIEXEC.EXE) and a package file (.msi file). Windows Installer uses the information contained within a package file to install the application.

Windows Internet Name Service (WINS) A software service that dynamically maps IP addresses to computer names (NetBIOS names). This allows users to access resources by name instead of requiring them to use IP addresses that are difficult to recognize and remember. WINS servers support clients running Windows NT 4.0 and earlier versions of Windows operating systems. See also Domain Name System (DNS).

Windows NT 4.0–compatible Locator See domain controller locator.

WINS See Windows Internet Name Service.

wire protocol A protocol that defines the formats of client and server messages and interactions with various application programming interfaces (APIs), which provide programmatic access to these protocols.

X

X.509 version 3 certificate Version 3 of the ITU-T recommendation X.509 for syntax and format. This is the standard certificate format used by Windows 2000 certificate-based processes. An X.509 certificate includes the public key and information about the person or entity to whom the certificate is issued, information about the certificate, plus optional information about the certification authority (CA) issuing the certificate. See also certificate; public key.

Z

zone In a DNS database, a zone is a contiguous portion of the DNS tree that is administered as a single separate entity by a DNS server. The zone contains resource records for all the names within the zone. In the Macintosh environment, a logical grouping that simplifies browsing the network for resources, such as servers and printers. It is similar to a domain in Windows 2000 Server networking. See also domain; Domain Name System (DNS); DNS server.

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z

IT Professional

Microsoft® Windows® 2000 Server Distributed Systems Guide

Inside, you'll find the comprehensive technical information and tools you need to understand and maintain distributed systems in Microsoft Windows 2000 Server.

Coverage includes:

- Using Active Directory™ services to centrally manage users, groups, security services, and network resources
- Extending the schema so users can define new objects and add properties to existing objects
- Resolving names in Active Directory
- Performing multimaster replication in an environment where all domain controllers are peers
- Deploying distributed security—authentication, access control, and encryption services—in a Windows 2000 operating system environment
- Managing Windows Clustering for better load balancing—and for increased server availability
- Using Group Policy to manage users' desktop environments

Part No. 097-0001948

Microsoft