Windows 2000 Server

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.)

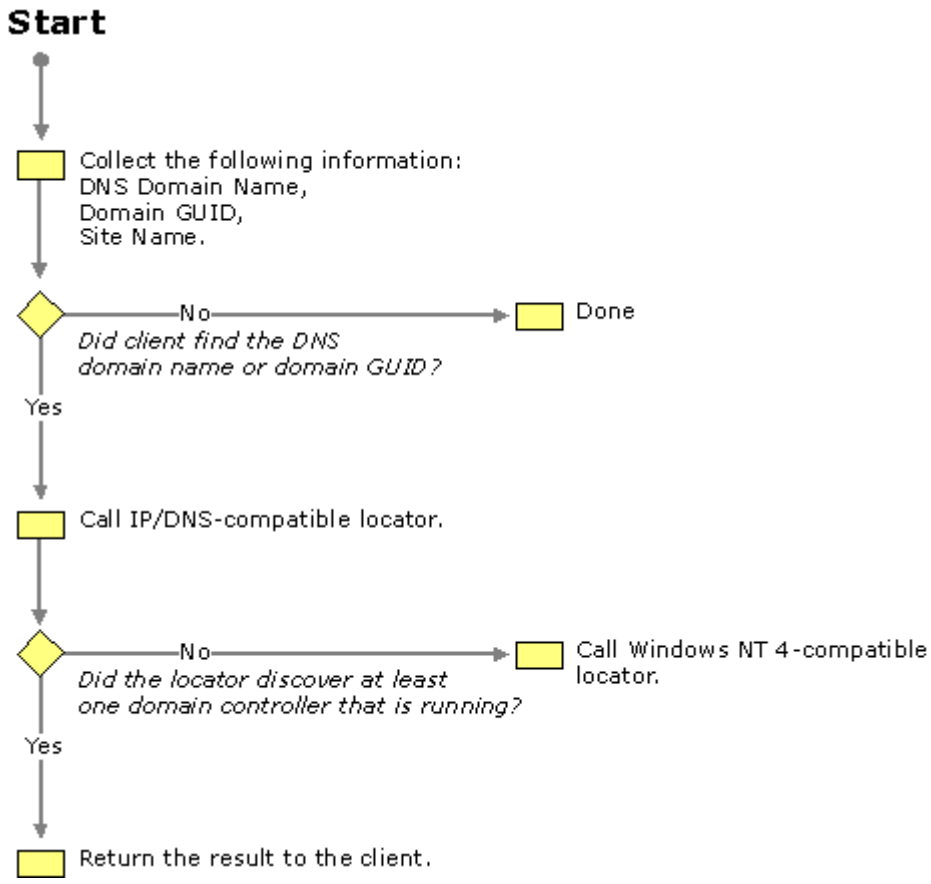
   Note UDP allows an application on one computer to send a datagram to an application on another computer. UDP includes a protocol port number, which allows the sender to distinguish among multiple destinations (applications) on the remote computer.
5. Each available domain controller responds to the datagram 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.



If your browser does not support inline frames, click here to view on a separate page.

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.

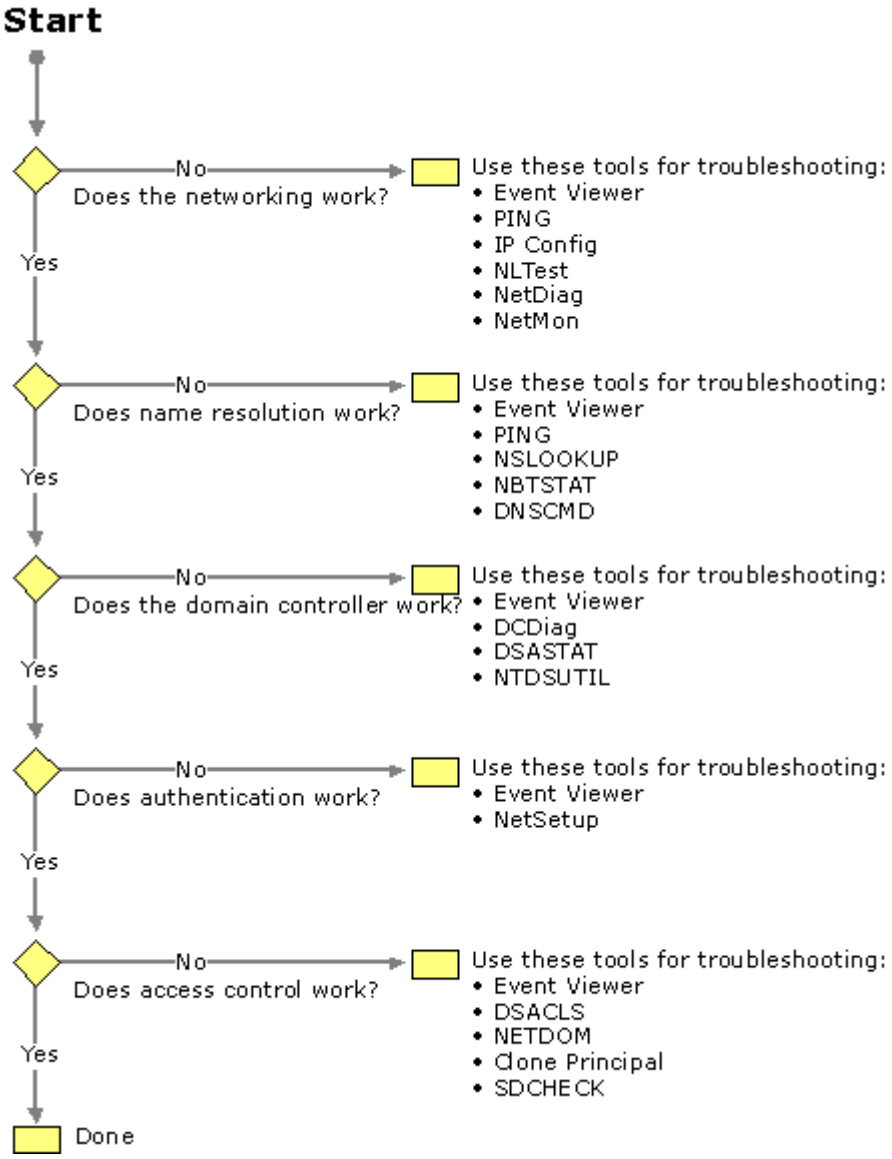
After the client has established a communications path to the domain controller, it can establish the logon and authentication credentials and, if necessary for Windows 2000 platforms, set up a secure channel. Then the client is ready to perform the normal queries and search for information against the 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.



If your browser does not support inline frames, click here to view on a separate page.

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.

Note 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
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.
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.
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.
