



Administrator's Guide to Active Directory, Second Edition

Book

- Planning & Migration
- Managing AD Objects & Services
- Security & Group Policy Administration
- Maintenance & Troubleshooting

CD

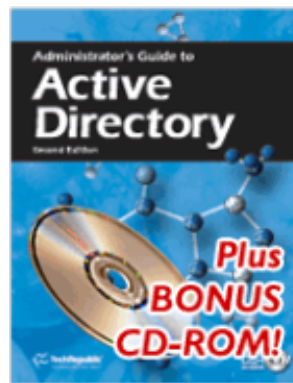
- Overview
- Installation and Support
- Active Directory and DNS
- Rights Management and Security
- Network Documentation
- Microsoft Certification Help

Your entire organization rests on the stability of your Microsoft 2000 Server or Windows Server 2003. And the central nervous system — the basis for the design and management of Windows networks — of that server is Active Directory.

You know Active Directory is complex and intimidating, but now you can make a worthwhile investment of your time and energy and learn it inside and out with TechRepublic's [Administrator's Guide to Active Directory, Second Edition](#). This expanded edition delivers the tips, solutions, and advice to help you install, configure, troubleshoot, and maintain Active Directory with confidence. You'll cover topics like:

- Mastering Active Directory schema administration and design
- Installing and configuring Active Directory
- Organizing and administering Active Directory objects
- Administering group policies
- Securing Microsoft directory services and Kerberos authentication
- Maintaining Active Directory replication
- Understanding and troubleshooting Active Directory DNS errors
- Backing up and restoring critical Active Directory data
- **Bonus:** Transcender exam demonstration and flash card files

From end-user rights and permissions to client systems, **Administrator's Guide to Active Directory, Second Edition** has the resources, guidance, templates, and checklists you need to meet every Active Directory challenge.



[Preview book: "Stretch Active Directory's schema to fit your needs"](#)

ORDER NOW

[Foreword](#)

[Table of Contents](#)

[CD: Table of Contents](#)

Administrator's Guide to Active Directory, Second Edition

Order Your Copy Today!

Understanding Active Directory transactions

May 3, 2001

By Talainia Posey

Active Directory is one of the most essential and complex structures in a Windows 2000 Server environment. Although Active Directory generally works well, things do occasionally go wrong. When that happens, the problems can be difficult to fix. In this article, I'll shed some light on the Active Directory diagnostic and troubleshooting process.

What do I do?

Before you can even think about repairing Active Directory, you need to know how it works. Any time you send a request to Active Directory, you must complete several steps before the request can be fulfilled. Therefore, the first step to repairing Active Directory is to figure out where the process is breaking down and then work on the problem based on the breakdown point.

Many times a problem that appears to be an Active Directory issue is actually caused by an external component. For example, Active Directory is highly dependent on services and protocols, such as Lightweight Directory Access Protocol (LDAP) and DNS. If one of these protocols were to malfunction, it could cause the illusion that Active Directory has problems.

Let's begin by reviewing the process that's involved when a client needs to log in to an Active Directory environment as well as the process used when an application on a client machine requests access to Active Directory. Once you understand the principles involved in how Active Directory works, you can start to troubleshoot problems.

How does an Active Directory request work?

When a client needs to authenticate to Active Directory, it uses a service called the Locator (sometimes called the Domain Controller Locator). The purpose of the Locator is to find a domain controller. Depending on the type of client, the Locator can accomplish this task by using DNS or by using NetBIOS

DANGER!

Keep in mind that the repair process can be dangerous. If you misdiagnose the problem and then act on your erroneous conclusion, you could possibly damage Active Directory worse than it already is. The damage you cause may even be severe enough to prevent Windows from booting. Therefore, even though your system is presently damaged, I strongly recommend making a complete backup of the system before you begin.

names. The Locator works as a remote procedure call (RPC) to the local Net Logon service.

During the early phases of the connection process, the Net Logon service collects some necessary information from the system. This information may include items such as the DNS Server name, the domain name, and the site name. Assuming you're working in a TCP/IP environment (and who isn't these days?), the next step in the process involves the client making a call to the DNS server to read the service resource records.

After the client has read the appropriate DNS records, it sends a datagram packet to the domain controllers through the use of LDAP. This process works similarly to pinging all domain controllers. At this point, every functional domain controller will respond to the datagram. The Net Logon service then sends to the client the name of the domain controller that responded first. The Net Logon service caches all of the information acquired through the domain controller discovery process. Having the information cached prevents the client from having to work through the discovery process at each login. It also increases the odds that the client will use the same domain controller for each login and thus have a consistent view of Active Directory.

Once the client has established communications with a domain controller, it uses LDAP

to query Active Directory. The client must now determine whether the domain controller that responded to the query is the most efficient domain controller to use based on location. It does this by looking at the domain controller's site and subnet information. If the domain controller is found to exist within the same site and subnet as the client, communications will continue. Otherwise, the client will attempt to locate a domain controller within the same site in which it exists.

If the client has already tried to find a domain controller within that site, however, and no domain controller was found, the client will use the domain controller it has already established communications with. If the selected domain controller is found to be the optimal domain controller for the client, the

domain controller's information is cached for future use. If the domain controller isn't optimal, the information is cached anyway, but the cache is cleared after 15 minutes so that the client can attempt to locate a more optimal domain controller.

Now that a functional communication session has been established between the client and the domain controller, the authentication process may begin. The authentication process involves making LDAP calls to the SAM. The SAM then passes the calls along to the DSA, the database layer, and finally to the ESE. You must remember that it's possible for an Active Directory problem to occur at any point along the way. Therefore, it's important to be able to follow the process from beginning to end.

TERMS TO KNOW

Here are a few terms that may come in handy when troubleshooting or dealing with Active Directory. You'll find that just about every Active Directory function uses these components. Therefore, let's review the function of these components:

- ▶ **Directory System Agent (DSA)**—The DSA component provides access to the physical database information located on the hard disk. The DSA is a part of the Local System Authority (LSA). Because of its integration with the LSA, the DSA is able to understand what each Active Directory entry does. For example, when you create an object in Active Directory, it's the DSA that manages the new object and assigns the appropriate attributes to it. The DSA is also responsible for managing internal Active Directory events, such as replication. Clients and internal processes can access the DSA through the Security Accounts Manager (SAM), the LDAP protocol, a remote procedure call (RPC), and in some cases, through a MAPI call.
- ▶ **Database layer**—The database layer is the component that organizes the structure of the database. For each database object, the database layer generates a unique distinguished name, relative distinguished name, and the distinguished nametag that's used by many internal functions. The database layer is responsible for preserving the database schema and for the creation, deletion, and modification of individual records within the database.
- ▶ **The Extensible Storage Engine (ESE)**—The ESE is the mechanism that actually puts the various database components together. The ESE is sometimes referred to as a Jet database. The database's format and functionality is similar to the databases used by WINS servers, certificate servers, and Microsoft Exchange servers, just to name a few.

The ESE works by recording each database transaction in a log file. The log files make it possible to recover database changes that have occurred after the last backup in the event of a database crash. Likewise, the log files also provide a method by which the database engine can verify the consistency of the database.

The ESE itself resides in a file called Esent.dll. The actual database file is called `\\Winnt\Ntds\Ntds.dit`. The log files reside in the same directory as the database file.

Asking the right questions

You can perform tests to discover exactly where the problem lies. These tests can be summarized in a series of questions:

- ▶ Does the network connection work?
- ▶ Does name resolution work?
- ▶ Does the domain controller work?
- ▶ Does authentication work?
- ▶ Does access control work?

Although some of these questions may sound oversimplified and appear to have obvious answers, it's important to work through the troubleshooting process from step one. For example, it might be easy to simply say, "Yes, the network is working." As you know, however, networks can be complicated. It's possible for a small portion of the network to be malfunctioning without displaying any obvious signs. For example, if you use multiple protocols on your network, it's possible for one of the protocols to fail and the network to keep running without generating any obvious signs of the problem (depending on your network's configuration, of course).

Because subtle problems can creep in and disguise themselves as a major problem, you should test for small problems first. After all, it's better to test for and correct a minor problem than to just assume you have a big problem. If you try to fix a big problem that doesn't actually exist, you'll probably end up having a big problem on top of the small problem that your network originally had.

Understanding the basic procedures

Now that you're familiar with the Active Directory structure and with the authentication process, and you know a few questions to ask, let's take a look at some of the general procedures you'll use to troubleshoot Active Directory.

The troubleshooting method I'll use will be based on the questions I posed earlier. It will start with the basic connection and progress all the way to data access, testing each component along the way. When I diagnose Active Directory problems, I follow these procedures:

1. Perform a diagnostic procedure to see if the network is functioning correctly.

After all, you can't expect higher-level communications such as those used by Active Directory to work if your basic connectivity is questionable.

2. Test the network's name resolution.

Remember that regardless of which name-resolution method you use, Active Directory absolutely cannot work without DNS support. Therefore, part of the process will involve testing the DNS services. However, if you use other name-resolution methods, you must test them to make sure they are working correctly as well.

3. Perform some tests on the domain controllers.

Remember that Active Directory is stored on and regulated by the domain controllers.

4. Test the authentication process.

5. Test the process of accessing data.

As you can see, this method starts with the most basic components and works toward the more complex structures. As you go through the steps, chances are you're bound to discover the problem.

Conclusion

In this article, I've familiarized you with the components involved in Active Directory transactions. As I did, I explained the client authentication process. Now that you understand the basics, you'll be able to begin the troubleshooting process. ~

Monitoring Active Directory performance

May 21, 2001

By Jim Boyce

If yours is a relatively small network, Active Directory (AD) performance will likely not be a major concern unless your hardware is simply not up to the task. As the number of domain controllers grows and more users and other objects are added to the directory, performance begins to become a more important consideration. In this article, I'll introduce you to several topics that will help you get started monitoring and managing Active Directory, the network, and domain controllers.

Monitoring network performance

One factor that naturally has a significant effect on Active Directory performance is network performance. LDAP queries, replication, and other directory functions take place across the network, so network performance bandwidth and performance can affect AD performance. The reverse is also true: AD can impose additional load on the network, affecting other network traffic for file and print sharing, streaming audio or video, and other functions.

Windows 2000 provides two primary tools for monitoring network performance. The first of these is Network Monitor, which runs under Windows 2000 Server and enables you to track network throughput. The version of Network Monitor included with Windows 2000 Server tracks only local traffic but can be quite useful for determining network performance at the server. If you need to test network performance to and from other systems, such as between domain controllers, you can upgrade to the version of Network Monitor included with Microsoft Systems Management Server 2.0 SP1 or later. Tracked remote systems must run the Network Monitor agent, which is supported for Windows NT and Windows 2000 but not for Windows 9x clients.

The second tool for monitoring network performance is the System Monitor included with Windows 2000. Choose Start | Programs | Administrative Tools | Performance to

access the System Monitor. In addition to monitoring server performance as described previously (such as CPU utilization and disk performance), you should also establish a baseline and monitor network performance to help you evaluate Active Directory's impact on the network and vice versa. By default, the System Monitor includes a performance object named Network Interface that you can use to monitor network transmission. In particular, you should add and monitor the following counters:

- ▶ **Packets Outbound Discarded:** This counter monitors the length of the outbound packet queue, by number of packets waiting in the queue. A queue with a few items indicates acceptable performance, but longer queues indicate the NIC is waiting for the network and is not keeping pace with the server, indicating a bottleneck.
- ▶ **Bytes Total/Sec:** Use this counter to track the rate at which bytes are sent and received on the interface. A higher number indicates better performance. Track the performance of each network interface to identify high utilization per interface and determine whether you need to use switches to segment the network or increase bandwidth.

If you install the Network Monitor driver on the system, you should see a Network Segment counter object in System Monitor, as well. Include the following counters from the Network Segment object in your monitoring scheme:

- ▶ **Broadcast Frames Received/Sec:** This counter lets you define a baseline over time against which to evaluate variations in network traffic.
- ▶ **% Network Utilization:** This counter provides a good indication of the bandwidth utilization for the local segment and enables you to evaluate the impact of certain network events—such as replication—on network bandwidth. Consider 30 percent utilization a maximum for unswitched

Ethernet. Adjust your acceptable benchmark based on your network topology.

- ▶ **Total Frames Received/Sec:** You can use this counter to monitor network-wide traffic and determine when switches and routers are becoming saturated, indicating a need for additional segmenting.

Monitoring server and domain controller performance

In addition to monitoring network performance, you also need to monitor server and domain controller performance through System Monitor. First, consider monitoring CPU utilization. In the Performance Monitor, select the Processor object and, at a minimum, monitor the % Processor Time counter. On multiprocessor systems you can monitor the total processor utilization or monitor individual CPUs. If the CPU utilization is high, it's a good indication that it's time for an upgrade, either through replacing the server or adding CPUs. You also should monitor available disk space for the volumes containing the directory database files, log files, and SYSVOL folder, which by default are stored in the \NTDS and \SYSVOL folders. Use the LogicalDisk object and monitor the Free Megabytes counter to keep tabs on free space in the target volumes.

In addition to monitoring general server performance items, you also should monitor domain controller performance issues. System Monitor provides two objects that enable you to monitor a broad range of counters for Active Directory. The first of these—the NTDS object—includes the following counters that you'll find useful for monitoring Active Directory performance:

- ▶ **DRA Inbound Bytes Total/Sec:** This counter shows total bytes received through replication per second. Lack of activity indicates that the network is slowing down replication.
- ▶ **DRA Inbound Object Updates Remaining in Packet:** This counter shows the number of object updates received for replication that have not yet been applied to the local server. The value should be low, with a higher value indicating that the hard-

ware is incapable of adequately servicing replication (warranting a server upgrade).

- ▶ **DRA Outbound Bytes Total/Sec:** This counter shows the total bytes sent per second. Lack of activity indicates that the hardware or network is slowing down replication.
- ▶ **DRA Pending Replication Synchronizations:** This counter indicates the replication backlog on the server. This value should be low, with a higher value indicating that the hardware is not adequately servicing replication.
- ▶ **DS Threads In Use:** This counter shows the number of threads in use by Active Directory, with a lack of activity typically pointing to network problems that are preventing client requests from succeeding.
- ▶ **Kerberos Authentications/Sec:** This counter shows the number of Kerberos authentications on the server per second. A lack of activity can indicate network problems that are preventing authentication requests from succeeding.
- ▶ **LDAP Bind Time:** This counter shows the time required for completion of the last LDAP binding, with a higher value pointing to either hardware or network performance problems.
- ▶ **LDAP Client Sessions:** This counter shows the number of connected LDAP client sessions, with a lack of activity pointing to network problems.
- ▶ **LDAP Searches/Sec:** This counter shows the number of LDAP searches per second performed by clients in the directory. A lack of activity points to network problems.
- ▶ **LDAP Successful Binds/Sec:** This counter shows the number of successful LDAP binds per second, with a lack of activity pointing to network problems.
- ▶ **NTLM Authentications:** This counter shows the number of NTLM authentications per second handled by the domain controller (from Windows 98 and Windows NT clients). A lack of activity points to network problems.

The second object that is useful for monitoring Active Directory performance is the Database object. Some of the Microsoft documentation indicates that the Database counters do not install by default, although on the systems I tested, the Database object was installed. If you open the System Monitor and can't find the Database object, use these steps to add it:

1. Create a new folder to contain the Database object's DLL. In this example, assume you create the folder C:\dataperf.
2. Copy the file %systemroot%\System32\Esentprf.dll to the \dataperf folder.
3. Create the following registry keys, if they do not already exist:

```
HKEY_LOCAL_MACHINE\SYSTEM\
CurrentControlSet\Services\ESENT
HKEY_LOCAL_MACHINE\SYSTEM\
CurrentControlSet\Services\ESENT\
Performance
```

Monitoring general server performance, network performance, and NTDS/Database performance will give you a good indication of domain controller and network health.

4. Create the following values under the Performance key:
Open : REG_SZ : OpenPerformanceData
Collect : REG_SZ : CollectPerformanceData
Close : REG_SZ : ClosePerformanceData
Library : REG_SZ : C:\Performance\esentprf.dll
5. Open a command console in %systemroot%\System32 and execute the following command:

```
Lodctr.exe Esentperf.ini
```

After you've loaded the Database object, restart the System Monitor to work with the Database counters. The counters you'll find most useful for monitoring Active Directory performance include:

- ▶ **Cache % Hit.** This counter shows the percentage of database page requests handled by the cache, thereby not causing a file I/O. A lack of activity can indicate that the server has insufficient physical memory.
- ▶ **Cache Page Fault Stalls/Sec.** This counter shows the number of page faults per second that go unserved due to lack of available pages in the database cache. A value other than zero indicates insufficient physical memory in the server.
- ▶ **Cache Page Faults/Sec.** This counter shows the number of page requests per second that cause the database cache to allocate new pages from the cache. This value should be low, with a higher value indicating insufficient physical memory in the server.
- ▶ **File Operations Pending.** This counter shows the number of file operations for the database file(s) currently pending by the operating system. The value should be low, with a higher value indicating insufficient physical memory and/or inadequate CPU availability or performance.
- ▶ **File Operations/Sec.** This counter shows the number of file operations per second generated by the database cache manager against the database files. The value should be low, with a higher value indicating inadequate physical memory in the server.
- ▶ **Log Record Stalls/Sec.** This counter shows the number of log records per second that could not be added to the log buffers because the buffers were full. The value should be zero or close to zero, with a higher value indicating inadequate physical memory in the server.
- ▶ **Log Threads Waiting.** This counter indicates the number of threads waiting on pending log writes. The value should be low, with a higher value indicating insufficient physical memory, poor disk performance, or poor disk structuring.
- ▶ **Table Open Cache Hits/Sec.** This counter shows the number of directory database tables open per second from the cache. A high value indicates better caching,

with a lower value typically indicating inadequate physical memory in the server.

Monitoring replication


Monitoring general server performance, network performance, and NTDS/Database performance will give you a good indication of domain controller and network health. You also should monitor replication to help identify potential problems, such as network congestion, that can affect directory replication. The Microsoft Windows 2000 Resource Kit includes a handful of tools to help you monitor replication:

- ▶ **Netdiag.exe:** This tool performs a wide range of tests to check network connectivity and DNS consistency. The tool has been updated to include additional tests and also to add functionality to existing tests. Netdiag.exe is a console-based command, and you can view its syntax and options by executing Netdiag.exe /? at a console prompt. The help/syntax information is relatively lengthy, so you might want to redirect the output to a text file so you can view it in Notepad.
- ▶ **Repadmin.exe:** This tool lets you view replication topology and force replication events between domain controllers. Use the /showreps switch to display the DC's replication partners, when the last replication

was attempted, and whether or not it was successful. Use the /showconn switch to view connection objects on the DC to determine whether the DC is configured to replicate with the appropriate servers.

- ▶ **Dcdiag.exe:** This tool performs several tests to check the status and health of a DC. These tests verify connectivity, replication, topology integrity, DC roles, and other aspects of the DC's function.
- ▶ **Replmon.exe:** Unlike the previous three tools, Replmon is a Windows-based application. You can use Replmon to view the status and performance of directory replication, force synchronization between DCs, and view replication topology graphically. You can generate status reports that include a wide variety of configuration and performance data on the monitored server.

Conclusion

Contrary to the hopes of hardware manufacturers, sometimes the cure to poor performance doesn't lie in throwing more hardware at a network and hoping things improve. Instead, you should monitor the performance of components of your network and tweak where necessary. In this article, I've shown you how to monitor the performance of Active Directory. 

Taking out the Active Directory trash

April 18, 2001

By Jim Boyce

The size of Active Directory (AD) depends on the number and type of objects it contains. As more objects are added, the directory grows in size. There is generally no appreciable change in performance as the size of the directory grows, and in general, there's little performance reason to reduce the directory size. Storage capacity is, however, a consideration. In this article, I'll take a look at what AD does when you remove entries from it.

What do you want on your tombstone?

When objects are deleted from the directory, they are not immediately removed. Instead, the directory service removes the majority of the object's attributes and tags the object as tombstoned. The tombstone state indicates that the object has been deleted but not removed from the directory, much like a deleted file is removed from the file allocation table but the data is not actually removed from the drive. The directory service moves tombstoned objects to the Deleted Objects container, where they remain until the garbage collection process removes the objects. Garbage collection also defragments the database, essentially rearranging the data to be contiguous, and thereby reducing the size of the database file. The primary consideration isn't performance but rather keeping disk utilization to a manageable size.

Time to take out the garbage

The garbage collection process by default runs every 12 hours on a DC. The length of time tombstoned objects remain in the directory service before being deleted is 60 days (by default). The tombstone lifetime must be significantly longer than the garbage collection frequency to ensure that deletion of objects is replicated to other DCs. These default values ensure that the tombstoned state of the objects is replicated and the objects are deleted from all DCs, because it is extremely unlikely

that it will take 60 days for a single replication to complete.

While you don't need to change the garbage collection interval or the tombstone lifetime, you can do so if your domain structure or replication scheme warrants it. For example, you might prefer to reduce the garbage collection interval to 24 hours to reduce server load and reduce the tombstone lifetime to 30 days to free up disk space more frequently. The maximum garbage collection interval is one-third of the tombstone lifetime. If you set the tombstone lifetime to 30 days, for example, the garbage collection interval will be 10 days, even if you've specified a larger value.

You can use the ADSI Edit tool included with the Windows 2000 Support Tools (located in the Support\Tools folder of the Windows 2000 CD) to modify the settings for garbage collection and tombstone lifetime. The values are attributes of the `cn=Directory Service,cn=Windows NT,cn=Services,cn=Configuration,dc=ForestRootDomain` object, and the attributes to change are `tombstoneLifetime` and `garbageCollPeriod`.

Defragmenting the Active Directory database

When AD performs the garbage collection process, it defragments the database; although it does not free up space on the disk, it simply restructures the existing data within the file. You use the `Ntdsutil.exe` command-line tool included with Windows 2000 to perform the defragmentation. While you can run `Ntdsutil` while the server is online, you must defragment the database with the directory service offline to recover disk space.

To start the server in Directory Services Restore Mode to perform the defragmentation, press [F8] at startup to display the Windows 2000 Advanced Options menu. Select Directory Services Restore Mode and press [Enter]. After the server boots, run the `Ntdsutil` utility to defragment the database. `Ntdsutil`

is an interactive console program that performs several actions on the database.


When you perform a defragmentation, *Ntdsutil* creates a new copy of the *Ntds.dit* database file in a different folder. You then replace the old file with the new one and restart the server. You should retain the old *Ntds.dit* file in case you experience problems with the new file. Also, compare the file size between the old and new files to determine how much space you've freed through the defragmentation.

In addition, you can configure Windows 2000 to log the amount of space that would be freed by an offline defragmentation to the Directory Service event log during garbage collection. You'll need to tweak the registry to accomplish this. Open the Registry Editor and set the value of `HKEY_LOCAL_MACHINE\`

`SYSTEM\CurrentControlSet\Services\NTDS\Diagnostics\Garbage Collection` to 1. Then, check the log after the next garbage collection to verify that the directory service is logging the data.

As explained above, *Ntdsutil.exe* is an interactive utility. Type *Ntdsutil.exe* at a console prompt and then enter `Help` to view the command options. You use the `Files` command to defragment the database.

Conclusion

AD is essentially a big database. As with any database, you have to do a little work to maintain it properly. In this article, I've shown you how to defragment the Active Directory database and how garbage collection works. 

Optimize Active Directory with NTDSUTIL

Jan. 10, 2002

By Talainia Posey

The Active Directory (AD) database is constantly changing, and over time, these changes can cause the database to respond to the system more slowly than necessary. To squeeze the best possible performance out of your servers, you should optimize your AD database. Here are a few techniques that you can use to increase the efficiency of your AD database and improve your overall system response time.

Database defragmentation

As the AD database is used, it becomes fragmented; the process of adding and removing database records leaves holes in the database, which causes the data to become scattered. You can help your server to perform better by defragmenting this database. The defragmentation process condenses the data within the database and reorganizes it in a logical fashion so that your system won't have to waste time

by jumping all over the database to get a piece of data here and a piece of data there.

Because database defragmentation is so important, Windows automatically defragments AD every 12 hours. However, there are times when it's appropriate to manually defragment the database. For example, if you've just made a lot of changes to AD, you should defragment the database. Or if the AD database is somehow damaged, then the defragmentation cycle may not complete. So if you've had to repair AD, you might want to defragment it after the repair, because the database may not have been defragmented in a while.

There are two basic methods of defragmenting your AD database. You can perform an online defragmentation or an offline defragmentation. An online defragmentation is basically the automated process that Windows performs every 12 hours. To see how long ago the database was last defragmented, check the

Directory Service event log. An online defragmentation writes an event number 700 when the process begins, as shown in **Figure A**.

An event number 701 is written to the event log on the domain controller where the defragmentation took place, as shown in **Figure B**.

As I mentioned, an online defragmentation rearranges records within the AD database and fills in holes. It may seem strange, but an online defragmentation never decreases the size of the database. Instead, it simply organizes all of the empty space into one group.

An offline defragmentation would be performed as a maintenance chore. It's referred to as offline because AD isn't actually running during the defragmentation process. To initiate an offline defragmentation, reboot the server. During the early phase of the boot process, press [F5] to access the boot menu. When you see the boot menu, select the Directory Service Restore Mode option. Windows will then boot into what appears to be Safe Mode with network support. However, running in this Directory Service Restore mode is necessary because doing so takes AD offline.

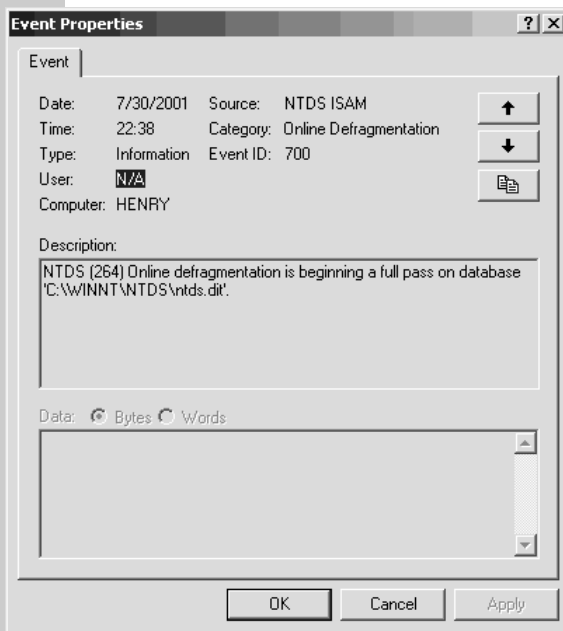
Next, open a command prompt and create a temporary directory on your partition with the most available free space. This partition

must have enough free space to accommodate a copy of your AD. If you're in doubt as to the size of your AD, search your system for the AD database's file name, Ntds.dit, and check its size. As you can see in **Figure C**, on my test system, the database was located in C:\WINNT\NTDS and was about 29 MB in size. The other copy that you see in the figure is an older copy that's no longer in use.

Enter the *NTDSUTIL* command in the command prompt window. When you see the NTDSUTIL prompt, enter the *FILES* command. Next, at the FILE MAINTENANCE prompt, enter the command *COMPACT TO pathoftemporarydatabase*, where you will enter the path to your database. Windows will then begin the defragmentation process, which could take some time, depending on the speed of your computer and the size of your AD database. When the process finishes, you'll have a compacted copy of the database in your temporary directory. Enter the *QUIT* command twice to exit the NTDSUTIL program. Then, overwrite your original database with the compacted copy. When you're done, erase all of the .log files from the original database location.

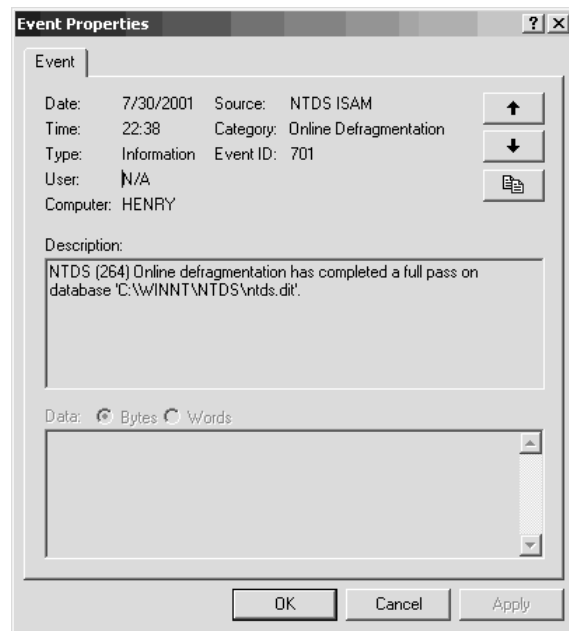
Figure D shows the entire process step-by-step.

Figure A



Event number 700 in the Directory Service Log indicates the defragmentation process has begun.

Figure B



Event number 701 in the Directory Service Log indicates the online defragmentation process has completed.

Be sure to clean up your work by erasing the temporary copy of your AD database. You can then return your system to normal operations by simply rebooting.

Moving AD

By default, Windows 2000 places the AD database in the `\%systemroot%\NTDS` directory, usually `\WINNT\NTDS`. The problem with this is that many times the system drive simply isn't big enough to accommodate AD.

If you've ever worked with Windows NT, you may recall a popular setup technique that involved creating a 2-GB FAT partition and placing the system files into it. The idea was that if a severe system crash were to occur, you could at least get to the system files to fix the problem, because they were on a FAT partition. Unfortunately, if you upgraded a copy of Windows NT that was configured in this way to Windows 2000, you'd have very little space left over, and your AD could quickly outgrow the partition.

The solution in such a case is to move your AD. Moving AD could also be used as a performance-boosting technique if you were to move the databases off of a slow partition and on to a faster partition.

To move AD, boot the system into Active Directory Restore Mode. When the system boots, verify the location of all AD files by opening a command prompt and entering the `NTDSUTIL` command. At the `NTDSUTIL` prompt, enter the `FILES` command. Then, type `INFO` at the prompt, and you'll see a summary of the size and locations of the various AD files as in **Figure E**.

The next step in the process is to select a location to move the database to. Remember that the partition that you select should have ample speed and space to effectively host AD. Also, Windows requires that the partition containing AD must be running the NTFS file system.

Next, enter the `QUIT` command twice to exit the `NTDSUTIL` program. Then, decide which partition should be hosting AD, and create a directory on that partition called `NTDS`. For example, on my test system, I'm creating a directory called `D:\NTDS`.

Enter the `NTDSUTIL` command followed by the `FILES` command. Then, enter the

Figure C

```

C:\Documents and Settings\Administrator.POSEY>ntdsutil
ntdsutil: quit
C:\Documents and Settings\Administrator.POSEY>cd\
C:\>dir ntds.dit /s
Volume in drive C has no label.
Volume Serial Number is A060-D194

Directory of C:\WINNT\NTDS

07/31/2001  09:10a                29,376,512 ntds.dit
              1 File(s)                29,376,512 bytes

Directory of C:\WINNT\system32

12/07/1999  08:00a                6,307,840 ntds.dit
              1 File(s)                6,307,840 bytes

Total Files Listed:
              2 File(s)                35,684,352 bytes
              0 Dir(s)                544,773,632 bytes free
C:\>

```

Figure D

```

C:\>md tempdb
D:\>c:
C:\>ntdsutil
ntdsutil: files
file maintenance: compact to d:\tempdb
Opening database [Current].
Using Temporary Path: D:\
Executing Command: C:\WINNT\system32\esentutl.exe /d "C:\WINNT\NTDS\ntds.dit" /8
/o /1"C:\WINNT\NTDS" /s"C:\WINNT\NTDS" /t"d:\tempdb\ntds.dit" /!0240 /p

Initiating DEFRAGMENTATION mode...
Database: C:\WINNT\NTDS\ntds.dit
Log files: C:\WINNT\NTDS
System files: C:\WINNT\NTDS
Temp. Database: d:\tempdb\ntds.dit

Defragmentation Status < % complete >
 0  10  20  30  40  50  60  70  80  90 100
!-----!-----!-----!-----!-----!
.....

Note:
It is recommended that you immediately perform a full backup
of this database. If you restore a backup made before the
defragmentation, the database will be rolled back to the state
it was in at the time of that backup.

Operation completed successfully in 18.557 seconds.

Spawned Process Exit code 0x0(0)

If compaction was successful you need to:
copy "d:\tempdb\ntds.dit" to "C:\WINNT\NTDS\ntds.dit"
and delete the old log files:
del C:\WINNT\NTDS\*.log
file maintenance: quit
ntdsutil: quit
C:\>d:
D:\>cd\tempdb
D:\tempdb>copy NTDS.DIT c:\winnt\ntds
Overwrite c:\winnt\ntds\ntds.dit? (Yes/No/All): y
1 file(s) copied.
D:\tempdb>c:
C:\>cd\winnt\ntds
C:\WINNT\NTDS>erase *.log
C:\WINNT\NTDS>

```

This is an example of the steps you'll perform to compact and defragment your AD database.

`MOVE DB TO PATH` command, where `PATH` represents the path to your directory. For example, in my case, I'd enter `MOVE DB TO D:\NTDS`. Windows 2000 will then run a script that could take a while to complete.

Figure E

```
Command Prompt - ntdsutil
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\Documents and Settings\Administrator.POSEY>ntdsutil
ntdsutil: files
file maintenance: info

Drive Information:

C:\ NTFS (Fixed Drive ) free(536.8 Mb) total(1.9 Gb)
D:\ FAT32 (Fixed Drive ) free(25.5 Gb) total(25.4 Gb)
E:\ FAT32 (Fixed Drive ) free(3.7 Gb) total(3.8 Gb)
Q:\ NTFS (Network Drive) free(60.2 Gb) total(63.5 Gb)

DS Path Information:

Database : C:\MINNT\NTDS\ntds.dit - 24.1 Mb
Backup dir : C:\MINNT\NTDS\dsadata.bak
Working dir: C:\MINNT\NTDS
Log dir : C:\MINNT\NTDS - 30.0 Mb total
        res2.log - 10.0 Mb
        res1.log - 10.0 Mb
        edb.log - 10.0 Mb

file maintenance:
```

When the moving process finishes, you'll need to move the database's log files. Enter the *MOVE LOGS TO PATH* command where *PATH* represents the path to your directory. This command would look something like *MOVE LOGS TO D:\NTDS*.

The last step of the process is to let Windows 2000 know about the changes that you've made by updating the database path. To do so, enter the command *SET PATH DB PATH*. For example, this command might look something like *SET PATH DB D:\NTDS\NTDS.DIT*. Next, enter the *QUIT* command twice and reboot your machine.

Continued on page 176

Listing A: *Sample text from an AD move*

```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\Documents and Settings\Administrator.POSEY>ntdsutil
ntdsutil: files
file maintenance: info

Drive Information:

C:\ NTFS (Fixed Drive ) free(536.8 Mb) total(1.9 Gb)
D:\ FAT32 (Fixed Drive ) free(25.5 Gb) total(25.4 Gb)
E:\ FAT32 (Fixed Drive ) free(3.7 Gb) total(3.8 Gb)
Q:\ NTFS (Network Drive) free(60.2 Gb) total(63.5 Gb)

DS Path Information:

Database : C:\WINNT\NTDS\ntds.dit - 24.1 Mb
Backup dir : C:\WINNT\NTDS\dsadata.bak
Working dir: C:\WINNT\NTDS
Log dir : C:\WINNT\NTDS - 30.0 Mb total
        res2.log - 10.0 Mb
        res1.log - 10.0 Mb
        edb.log - 10.0 Mb

file maintenance: quit
ntdsutil: quit

C:\Documents and Settings\Administrator.POSEY>d:

D:\>md ntds

D:\>ntdsutil
ntdsutil: files
file maintenance: move db to d:\ntds
```

Listing A: Continued

Opening database [Current].

```
D:\>REM - *****
```

```
D:\>REM - Script to move DS DB file
```

```
D:\>REM - *****
```

```
D:\>d:
```

```
D:\>cd \
```

```
D:\>mkdir "ntds"
```

```
A subdirectory or file ntds already exists.
```

```
D:\>cd "ntds"
```

```
D:\ntds>move "C:\WINNT\NTDS\ntds.dit" "d:\ntds\ntds.dit"
```

```
D:\ntds>C:\WINNT\system32\ntdsutil.exe files "set path DB  
  \"d:\ntds\ntds.dit\""
```

```
quit quit
```

```
C:\WINNT\system32\ntdsutil.exe: files
```

```
file maintenance: set path DB "d:\ntds\ntds.dit"
```

```
file maintenance: quit
```

```
C:\WINNT\system32\ntdsutil.exe: quit
```

```
D:\ntds>C:\WINNT\system32\ntdsutil.exe files "set path backup  
  \"d:\ntds\DSADATA
```

```
BAK\"" quit quit
```

```
C:\WINNT\system32\ntdsutil.exe: files
```

```
file maintenance: set path backup "d:\ntds\DSADATA.BAK"
```

```
file maintenance: quit
```

```
C:\WINNT\system32\ntdsutil.exe: quit
```

```
D:\ntds>C:\WINNT\system32\ntdsutil.exe files "set path working dir  
  \"d:\ntds\""
```

```
quit quit
```

```
C:\WINNT\system32\ntdsutil.exe: files
```

```
file maintenance: set path working dir "d:\ntds"
```

```
file maintenance: quit
```

```
C:\WINNT\system32\ntdsutil.exe: quit
```

```
D:\ntds>C:\WINNT\system32\ntdsutil.exe files info quit quit
```

```
C:\WINNT\system32\ntdsutil.exe: files
```

```
file maintenance: info
```

Drive Information:

```
C:\ NTFS (Fixed Drive ) free(560.9 Mb) total(1.9 Gb)
```

```
D:\ FAT32 (Fixed Drive ) free(25.3 Gb) total(25.4 Gb)
```

Listing A: Continued

```
E:\ FAT32 (Fixed Drive ) free(3.7 Gb) total(3.8 Gb)
Q:\ NTFS (Network Drive) free(60.2 Gb) total(63.5 Gb)
```

DS Path Information:

```
Database   : d:\ntds\ntds.dit - 24.1 Mb
Backup dir  : d:\ntds\DSADATA.BAK
Working dir : d:\ntds
Log dir     : C:\WINNT\NTDS - 30.0 Mb total
              res2.log - 10.0 Mb
              res1.log - 10.0 Mb
              edb.log  - 10.0 Mb
```

file maintenance: quit

C:\WINNT\system32\ntdsutil.exe: quit

D:\ntds>REM - *****

D:\ntds>REM - Please make a backup immediately else restore

D:\ntds>REM - will not retain the new file location.

D:\ntds>REM - *****

Opening database [Current].

If move database was successful,

please make a backup immediately else restore

will not retain the new file location.

file maintenance: move logs to d:\ntds

Opening database [Current].

D:\>REM - *****

D:\>REM - Script to move DS log files

D:\>REM - *****

D:\>d:

D:\>cd \

D:\>mkdir "ntds"

A subdirectory or file ntds already exists.

D:\>cd "ntds"

D:\ntds>move "C:\WINNT\NTDS\res2.log" "d:\ntds\res2.log"

D:\ntds>move "C:\WINNT\NTDS\res1.log" "d:\ntds\res1.log"

D:\ntds>move "C:\WINNT\NTDS\edb.log" "d:\ntds\edb.log"

Listing A: Continued

```
D:\ntds>C:\WINNT\system32\ntdsutil.exe files "set path logs \"d:\ntds\""  
quit quit
```

```
C:\WINNT\system32\ntdsutil.exe: files  
file maintenance: set path logs "d:\ntds"  
file maintenance: quit  
C:\WINNT\system32\ntdsutil.exe: quit
```

```
D:\ntds>C:\WINNT\system32\ntdsutil.exe files info quit quit  
C:\WINNT\system32\ntdsutil.exe: files  
file maintenance: info
```

Drive Information:

```
    C:\ NTFS (Fixed Drive ) free(590.8 Mb) total(1.9 Gb)  
    D:\ FAT32 (Fixed Drive ) free(25.0 Gb) total(25.4 Gb)  
    E:\ FAT32 (Fixed Drive ) free(3.7 Gb) total(3.8 Gb)  
    Q:\ NTFS (Network Drive) free(60.2 Gb) total(63.5 Gb)
```

DS Path Information:

```
    Database   : d:\ntds\ntds.dit - 24.1 Mb  
    Backup dir : d:\ntds\DSADATA.BAK  
    Working dir: d:\ntds  
    Log dir    : d:\ntds - 30.0 Mb total  
              edb.log - 10.0 Mb  
              res1.log - 10.0 Mb  
              res2.log - 10.0 Mb
```

```
file maintenance: quit  
C:\WINNT\system32\ntdsutil.exe: quit
```

```
D:\ntds>REM - *****
```

```
D:\ntds>REM - Please make a backup immediately else restore
```

```
D:\ntds>REM - will not retain the new file location.
```

```
D:\ntds>REM - *****
```

```
Opening database [Current].  
If move log files was successful,  
please make a backup immediately else restore  
will not retain the new file location.
```

```
file maintenance: set path db d:\ntds\ntds.dit  
file maintenance:
```

Continued from page 172

Because it's impossible to fit the move process into a screen capture, I've included the text output from an actual move. Note, however, that after you complete a move, you need to immediately back up the system, because you may not be able to restore your old system backups, since the backup will be looking for your AD in the wrong location. With that said, **Listing A** (page 172) shows the text of the sample move.

Conclusion

As time goes on, your AD will become fragmented. When this happens, you can greatly boost performance by defragmenting the database. Although there are no handy GUI tools to accomplish this, once you get used to using the NTDSUTIL utility, you'll be able to defragment AD with little trouble. 