Active Directory Diagnostic Tool

Active Directory Diagnostic Tool (Ntdsutil.exe)

Ntdsutil.exe is a command-line tool that provides management facilities for Active Directory™ , the Microsoft® Windows® 2000 directory service. Use Ntdsutil to perform database maintenance of Active Directory, to manage and control single master operations, and to remove metadata left behind by domain controllers that were removed from the network without being properly uninstalled. This tool is intended to be used by experienced administrators. By default, Ntdsutil is installed in the Winnt\System32 folder.

Invoking Ntdsutil Commands and Parameters

Ntdsutil provides menus that list the set of commands for the program. At any level, you can enter **?** or **Help** (or even **h** if there are no other options that start with **h** at that level!) to read the list of commands for that part of the program. The command **quit** (or **q**, if sufficiently unique) is the universal command to return to the prior menu. The command **quit**, when used at the outermost level, exits the program.

For more information about Ntdsutil menu commands, see "List of Menu Commands" later in this document.

How to Use Ntdsutil Menu Commands

You can invoke Ntdsutil from the command prompt with no arguments. Rather than support and extend an ever-increasing set of cryptic command-line arguments, the tool parses keyboard input after you invoke it. The commands have been made as conversational as possible. For example, you can type the following:

list roles for connected server

connect to server xxx

For convenience, Ntdsutil allows you to be cryptic. You need only specify enough of each word to make it unique with respect to any other words that you can enter at that time. Thus, as you become more familiar with the tool, you might type the following:

li r f c s

rather than:

list roles for connected server

How Ntdsutil Processes Command Input

Ntdsutil processes as input all the arguments that you type when you start the program. For example, if you type the following:

ntdsutil help connections help quit quit

Ntdsutil does the following steps:

1. Invokes Ntdsutil.exe.
2. Displays its Help information.
3. Enters the Connections submenu.
4. Displays its Help information.
5. Closes the **Connections** submenu and returns to the top-level menu.

6. Quits the program.

How to Use Arguments with Ntdsutil Commands

Some commands take arguments that are shown as either **%s** or **%d** in Help. As you probably know, **%d** is the decimal number specifier and **%s** is the string specifier for the C-language program print and scan commands. When you enter a command whose Help indicates you use either a **%d** or **%s**, enter a number or string respectively. For example, one Help listing shows the following:

connect to server %s

You would type this command as follows:

connect to server xxx

where xxx is the character string you want to substitute for %s. If your string has spaces in it, enclose it in quotation marks as follows:

connect to server "xxx yyy"

How to Automate Ntdsutil Commands

You can automate Ntdsutil by creating batch files or scripts that contain a series of Ntdsutil commands. Many Ntdsutil commands that perform writes, open by default a message that asks users if they really want to perform a particular operation. When these messages appear, the program will pause and wait for keyboard input. Use the Popups %s command to disable these messages when running Ntdsutil from a batch file or script. For example, to disable these messages, type the following:

popups no

To reenable the display of these messages, type the following:

popups yes

It is good practice to disable these messages only when you are scripting Ntdsutil commands and to reenable them as soon as you finish scripting.

Managing Active Directory Files

The Microsoft® Windows® 2000 directory service is implemented on top of an indexed sequential access method (ISAM) table manager. This is the same table manager used by Microsoft® Exchange Server, the file replication service, the security configuration editor, the certificate server, Windows Internet Name Service (WINS), and other Windows 2000 components. The version of the database that Windows 2000 uses is called extensible storage engine (ESENT).

ESENT is a transacted database system that uses log files to support rollback semantics to ensure that transactions are committed to the database. Ideally, data and log files should be located on separate drives to improve performance and support recovery of the data if a disk fails.

The data file is called Ntds.dit. The Files menu of Ntdsutil provides commands for managing the directory service data and log files.

ESENT provides its own tool for certain database file management functions called Esentutl.exe, which is also installed in the Winnt\System32 folder. Several of the Ntdsutil file management commands invoke Esentutl, reducing the need to learn that tool's command-line arguments. In the cases where Ntdsutil invokes Esentutl, it brings up a separate window configured with a large history so that you can scroll back to see all of the Esentutl progress indicators.

The Windows 2000 directory service opens its files in exclusive mode. This means the files cannot be managed while the system is operating as a domain controller.

To manage directory service files

Start the computer.

When the **Starting Windows** progress bar appears, press F8.

From the **Windows 2000 Advanced Options Menu**, select **Directory Services Restore Mode**.

Note
Starting the computer in Directory Services Restore Mode causes your domain controller to temporarily operate as a stand-alone server. This causes some services to fail, especially those that are integrated with the directory service. When operating in this mode, the security accounts manager (SAM) uses a minimal set of user and group definitions stored in the registry. If your domain controller is not physically secure, you should set the administrative password for the Directory Services Restore Mode.

Table 1 lists and describes the file management commands.

Table 1 File Management Commands

Command	**Description**
Compact to %s (where %s identifies an empty target directory)	Invokes Esentutl.exe to compact the existing data file and writes the compacted file to the specified directory. The directory can be remote, that is, mapped by means of the **net use** command or similar means. After compaction is complete, archive the old data file, and move the newly compacted file back to the original location of the data file. ESENT supports online compaction, but this compaction only rearranges pages within the data file and does not release space back to the file system. (The directory service invokes online compaction regularly.)
Header	Writes the header of the Ntds.dit data file to the screen. This command can help support personnel analyze database problems.
Info	Analyzes and reports the free space for the disks that are installed in the system, reads the registry, and then reports the sizes of the data and log files. (The directory service maintains the registry, which identifies the location of the data files, log files, and directory service working directory.)
Integrity	Invokes Esentutl.exe to perform an integrity check on the data file, which can detect any kind of low-level database corruption. It reads every byte of your data file; thus it can take a long time to process large databases. Note that you should always run Recover before performing an integrity check.
Move DB to %s (where %s identifies a target directory)	Moves the Ntds.dit data file to the new directory specified by **%s** and updates the registry so that, upon system restart, the directory service uses the new location.
Move logs to %s (where %s identifies a target directory)	Moves the directory service log files to the new directory specified by **%s** and updates the registry so that, upon system restart, the directory service uses the new location.
Recover	Invokes Esentutl.exe to perform a soft recovery of the database. Soft recovery scans the log files and ensures all committed transactions therein are also reflected in the data file. The Windows 2000 Backup program truncates the log files appropriately.

	Logs are used to ensure committed transactions are not lost if your system fails or if you have unexpected power loss. In essence, transaction data is written first to a log file and then to the data file. When you restart after failure, you can rerun the log to reproduce the transactions that were committed but hadn't made it to the data file.
Repair	Invokes Esentutl.exe to perform a low-level repair of the data file. Use the repair command only on the advice of qualified service personnel, as it can cause data loss. Furthermore, this can only repair what ESENT knows about. This means that its notion of repair might eliminate some data that is key to the safe operation of the directory service.
Set path backup %s (where %s identifies a target directory)	Sets the disk-to-disk backup target to the directory specified by **%s**. The directory service can be configured to perform an online disk-to-disk backup at scheduled intervals.
Set path DB %s (where %s identifies a target directory)	Updates the part of the registry that identifies the location and file name of the data file. Use this command only to rebuild a domain controller that has lost its data file and that is not being restored by means of normal restoration procedures.
Set path logs %s (where %s identifies a target directory)	Updates the part of the registry that identifies the location of the log files. Use this command only if you are rebuilding a domain controller that has lost its log files and is not being restored by means of normal restoration procedures.
Set path working dir %s	Sets the part of the registry that identifies the directory service's working directory to the directory specified by **%s**.

Example using Ntdsutil to perform Offline Defragmentation of NTDS.DIT:

The database file cannot be compacted while Active Directory is mounted. An ntds.dit file that has been defragmented offline (compacted), can be much smaller than the ntds.dit file on its peers. To defrag ntds.dit offline:

Back up the Active Directory using Windows 2000 Backup. W2K backup natively supports backing up Active Directory while online. This occurs automatically when you select the option to back up everything on the computer in the Backup Wizard, or independently by selecting to back up **System State** in the backup wizard.

- Reboot
- Select the appropriate installation from the boot menu, and press F8 to display the **Windows 2000 Advanced Options** menu.
- Choose **Directory Services Restore Mode** and press ENTER. Press ENTER again to start the boot process.

  Logon using the password defined for the local Administrator account in the offline SAM.
- Click Start, Programs, Accessories, and then click Command Prompt.
- At a command prompt, type **ntdsutil** then press ENTER.

  Once **ntdsutil** starts, type the following commands within the utility:
- At the Ntdsutil prompt type **files** then press ENTER.
- At the File Maintenance prompt type **info** then press ENTER.

  This will display current information about the path and size of the Active Directory database and its log files.
- At the File Maintenance prompt type **Compact to *drive:\directory*** then press ENTER.

Be sure that the drive specified has enough drive space for the compacted database to be created. I know, you don't know how big the compacted version will be, but if there is enough space for the uncompacted version, you should be OK. A gotcha!: You must specify a directory path and if the path name has spaces, the command will not work unless you use quotation marks compact to "c:\my new folder"

- At the File Maintenance prompt type **quit** then press ENTER.
- At the Ntdsutil prompt type **quit** then press ENTER to return to the command prompt. A new compacted database named Ntds.dit can be found in the folder you specified.
- Copy the **new** ntds.dit file over the old ntds.dit file.

  You have successfully compacted the Active Directory database.

  If you believe in belts and suspenders, you could copy the old uncompacted database somewhere else before I overwrote it with the new compacted version.
- Reboot and see if all is normal.

Example using Ntdsutil to move a database or log file:

- Reboot the domain controller and press F8 to display the **Windows 2000 Advanced Options** menu.
- Select **Directory Services Restore Mode** and then press ENTER.

  Select the correct installation, and then press ENTER to start the boot process.

  Logon using the administrator account and password you specified during the promotion process.

  When you ran Dcpromo.exe to install Active Directory, it requested a password to be used for the Administrator password for Active Directory Restore Mode. This password is not stored in Active Directory. It is stored in an NT4-style SAM file and is the only account available when the AD is corrupted.
- Click Start, Programs, Accessories, and then click Command Prompt.
- At a command prompt, type **ntdsutil** then press ENTER.

  Once **ntdsutil** starts, type the following commands within the utility:
- At the Ntdsutil prompt, type **files** then press ENTER.
- At the File Maintenance prompt:

  To move a database, type **move db to %s** then press ENTER, where %s is the drive and folder where you want the database moved.

  To move log files, type **move logs to %s** then press ENTER, where %s is the drive and folder where you want the log files moved.

  To view the log files or database, type **info** then press ENTER.

  To verify the integrity of the database at its new location, type **integrity** then press ENTER.
- At the File Maintenance prompt type **quit** then press ENTER.
- At the Ntdsutil prompt type **quit** then press ENTER to return to a command prompt.
- Restart the computer in Normal mode.

When you move the database and log files, you must back up the domain controller.

Using the Connections Menu

Several Ntdsutil operations send LDAP or RPC commands to a particular server. The **Connections** menu appears as a selection within several other menus in Ntdsutil and provides a way to connect to a specific server or domain. By default, your connection is authenticated using the credentials of the logged-on user. You can also specify the credentials to use when authenticating to a server.

Table 2 lists and describes the connection commands.

Table 2 Connection Commands

Command	Description
Clear creds	Clears any previously defined credentials and disconnects any previous connections to prevent ambiguity about which credentials are in use during subsequent commands.
Connect to domain %s (where %s identifies a target domain)	Finds any domain controller for the domain specified by **%s** and connects to it using the default credentials or any credentials specified earlier by **Set creds**.
Connect to server %s (where %s identifies a target domain controller)	Connects to the domain controller specified by **%s** using the default credentials or any credentials specified earlier by **Set creds**.
Info	Displays the credentials now in use and the current connection state.
Set creds %s %s %s (where the first %s represents the domain, the second %s represents the user name, and the third %s represents the password)	Sets the credentials for use in subsequent **Connect To** commands. Use the literal string "Null" to specify a null password.

Selecting an Operation Target

Several Ntdsutil operations require you to identify a particular site, server, or domain by its object in the Configuration container. Instead of requiring that you enter the full distinguished names of objects, many of these operations provide a numbered list of valid selections.

The **Select operation target** submenu appears as a selection within several other menus in Ntdsutil and provides a way to query an existing and operating domain controller about significant objects in the Configuration container. The general model is to connect to a server and list the objects that it knows about. The objects are displayed to the console and numbered from zero onward. You select a particular object by entering its number instead of its distinguished name.

Table 3 lists and describes the select operation target commands.

Table 3 Select Operation Target Commands

Command	Description
Connections	Invokes the **Connections** submenu.
List current selections	Lists the currently selected site, domain, and server.

List domains	Lists all domains that have a corresponding cross-reference object in the partitions container. Note that some of these domains might not exist if the last domain controller for a domain was removed without performing the demotion properly.
List domains in site	Lists the domains that have domain controllers in the currently selected site.
List roles for connected server	Lists all the operations master roles that the server to which you are connected knows about and displays the domain controllers that are the current operations master role owners. Due to replication latency, the server to which you are connected might not have an up-to-date view of the current role owners.
List servers for domain in site	Lists all the servers known for the currently selected site and domain.
List servers in site	Lists all the domain controllers known to be in the currently selected site.
List sites	Lists all the sites in the forest.
Select domain %d	Selects the domain specified by **%d**.
Select server %d	Selects the server specified by **%d**.
Select site %d	Selects the site specified by **%d**.

Managing Operations Master Roles

Although Active Directory is based on a multimaster administration model, some operations support only a single master. For multimaster operations, conflict resolution ensures that after the system finishes replicating, all replicas agree on the value for a given property on a given object. However, some data, for which adequate conflict resolution is not possible, is key to the operation of the system as a whole. This data is controlled by individual domain controllers called operations masters. These domain controllers are referred to as holding a particular operations master role.

Note
Operations masters are sometimes referred to as Flexible Single Master Operations (FSMOs).

There are five operations master roles: some are enterprisewide, and some are per domain. The following paragraphs describe these five roles:

Schema Operations Master

There is a single schema operations master role for the entire enterprise. This role allows the operations master server to accept schema updates. There are other restrictions on schema updates.

Relative ID Master

There is one relative ID master per domain. Each domain controller in a domain has the ability to create security principals. Each security principal is assigned a relative ID. Each domain controller is allocated a small set of relative IDs out of a domainwide relative ID pool. The relative ID master role allows the domain controller to allocate new subpools out of the domainwide relative ID pool.

Domain-Naming Master

There is a single domain-naming master role for the entire enterprise. The domain-naming master role allows the owner to define new cross-reference objects representing domains in the Partitions container.

PDC Operations Master

There is one primary domain controller (PDC) operations master role per domain. The owner of the PDC operations master role identifies which domain controller in a domain performs

Microsoft® Windows NT® version 4.0 PDC activities in support of Windows NT 4.0 backup domain controllers and clients using earlier versions of Windows.

Infrastructure Master

There is one infrastructure master role per domain. The owner of this role ensures the referential integrity of objects with attributes that contain distinguished names of other objects that might exist in other domains. Because Active Directory allows objects to be moved or renamed, the infrastructure master periodically checks for object modifications and maintains the referential integrity of these objects.

An operations master role can only be moved by administrative involvement; it is not moved automatically. Additionally, moving a role is controlled by standard Windows 2000 access controls. Thus a corporation should tightly control the location and movement of operations master roles. For example, an organization with a strong IT presence might place the schema role on a server in the IT group and configure its access control list (ACL) so that it cannot be moved at all.

Operations master roles require two forms of management: controlled transfer and seizure.

Use controlled transfer when you want to move a role from one server to another, perhaps to track a policy change with respect to role location or in anticipation of a server being shut down, moved, or decommissioned.

Seizure is required when a server that is holding a role fails and you do not intend to restore it. Even in the case of a server recovered from a backup, the server does not assume that it owns a role (even if the backup tape says so), because the server cannot determine if the role was legitimately transferred to another server in the time period between when the backup was made and the server failed and was recovered. The restored server assumes role ownership only if a quorum of existing servers is available during recovery and they all agree that the restored server is still the owner.

The Roles submenu in Ntdsutil is used to perform controlled transfer and recovery of operations master roles. Controlled transfer is simple and safe. Because the source and destination servers are running, the system software guarantees that the operations master role token and its associated data is transferred atomically. Operations master role seizure is equally simple but not as safe. You simply tell a particular domain controller that it is now the owner of a particular role.

Caution
Do not make a server a role owner by means of seizure commands if the real role holder exists on the network. Doing this could create irreconcilable conflicts for key system data. If an operations master role owner is temporarily unavailable, do not make another domain controller the role owner. This could result in a situation where two computers function as the role owner, which might cause irreconcilable conflicts for key system data.

The commands listed in Table 4 are found in the **Roles** submenu and perform controlled transfer and recovery of operations master roles.

Table 4 Roles Commands

Command	Description
Abandon all roles	Instructs the domain controller to which you are connected to give away all operations master roles it owns. This command is not guaranteed to succeed because eligible role recipients might be currently unreachable or because the domain controller to which you are connected is the last domain controller for the domain.
Connections	Invokes the **Connections** submenu.
Seize domain naming master	Forces the domain controller to which you are connected to claim ownership of the domain-naming operations master role without regard to the data associated with the role. Use only for recovery purposes.

Seize infrastructure master	Forces the domain controller to which you are connected to claim ownership of the infrastructure operations master role without regard to the data associated with the role. Use only for recovery purposes.
Seize PDC	Forces the domain controller to which you are connected to claim ownership of the PDC operations master role without regard to the data associated with the role. Use only for recovery purposes.
Seize RID master	Forces the domain controller to which you are connected to claim ownership of the relative ID master role without regard to the data associated with the role. Use only for recovery purposes.
Seize schema master	Forces the domain controller to which you are connected to claim ownership of the schema operations master role without regard to the data associated with the role. Use only for recovery purposes.
Select operation target	Invokes the **Select operation target** submenu.
Transfer domain naming master	Instructs the domain controller to which you are connected to obtain the domain-naming role by means of controlled transfer.
Transfer infrastructure master	Instructs the domain controller to which you are connected to obtain the infrastructure operations master role by means of controlled transfer.
Transfer PDC	Instructs the domain controller to which you are connected to obtain the PDC operations master by means of controlled transfer.
Transfer RID master	Instructs the domain controller to which you are connected to obtain the relative ID master role by means of controlled transfer.
Transfer schema master	Instructs the domain controller to which you are connected to obtain the schema operations master role by means of controlled transfer.

Example using Ntdsutil to seize a particular FSMO role:

You can seize particular operations master roles using the Ntdsutil tool as in the following example:

- At a command prompt, type **ntdsutil** then press ENTER.

  Once **ntdsutil** starts, type the following commands within the utility:

- At the Ntdsutil prompt type **roles** then press ENTER.
- At the FSMO Maintenance prompt type **connections** then press ENTER.
- At the Server connections prompt type **connect to *OneOfYourDCs*** then press ENTER.

  binding to OneOfYourDCs ...

  Connected to OneOfYourDCs using credentials of locally logged on user

- At the Server connections prompt type **quit** then press ENTER.
- At the FSMO Maintenance prompt type **seize RID master** then press ENTER.

  Server "OneOfYourDCs" knows about 5 roles

  Schema – CN=NTDS Settings,CN=server04,CN=Servers, CN=New-York,CN=Sites,CN=Configuration,DC=MyDomain,DC=com

  Domain – CN=NTDS Settings,CN=server04,CN=Servers, CN=New-York,CN=Sites,CN=Configuration,DC=MyDomain,DC=com

  PDC – CN=NTDS Settings,CN=server05,CN=Servers, CN=Chicago,CN=Sites,CN=Configuration,DC=MyDomain,DC=com

  RID – CN=NTDS Settings,CN=server05,CN=Servers, CN=Chicago,CN=Sites, CN=Configuration,DC=MyDomain,DC=com

  Infrastructure – CN=NTDS Settings,CN=server12,CN=Servers, CN=San-Francisco,CN=Sites,CN=Configuration,DC=MyDomain,DC=com

- At the FSMO Maintenance prompt type **quit** then press ENTER.

- At the Ntdsutil prompt type **quit** then press ENTER.

Managing Orphaned Metadata

The directory service maintains various metadata for each domain and server known to the forest. Normally, domains and domain controllers are created by means of promotion using the Active Directory Installation wizard provided with the Windows 2000 operating system and are removed by means of demotion using the same tool. You can invoke the Active Directory Installation wizard by typing **dcpromo** at the command prompt or by selecting the **Active Directory** option that is displayed on the welcome page of Configure Your Server.

Promotion and demotion are designed to correctly clean up the appropriate metadata. In the directory, however, you might have domain controllers that were decommissioned incorrectly. In this case, their metadata is not cleaned up. For example, a domain controller has failed, and rather than attempting to restore it, you decide to retire the server. This leaves some information about the retired domain controller in the directory. The general model of operation is to connect to a server known to have a copy of the offending metadata, select an operation target, and then delete it.

Caution
Do not delete the metadata of existing domains and domain controllers.

Table 5 lists and describes the metadata cleanup commands.

Table 5 Metadata Cleanup Commands

Command	Description
Connections	Invokes the **Connections** submenu.
Remove selected domain	Removes the metadata associated with the domain selected in the **Select operation target** submenu.
Remove selected server	Removes the metadata associated with the domain controller selected in the **Select operation target** submenu.
Select operation target	Invokes the **Select operation target** submenu.

Performing an Authoritative Restore

When a domain contains more than one domain controller, Active Directory replicates directory objects, such as users, groups, organizational units, and computers, to all the domain controllers in that domain.

When you are restoring a domain controller by using backup and restore programs, such as Ntbackup or those from third-party providers, the default mode for the restore is nonauthoritative. This means that the restored server is brought up-to-date with its replicas through the normal replication mechanism. For example, if a domain controller is restored from a backup tape that is two weeks old, when you restart it, the normal replication mechanism brings it up-to-date with respect to its replication partners.

Authoritative restore allows the administrator to recover a domain controller, restore it to a specific point in time, and mark objects in Active Directory as being authoritative with respect to their replication partners. For example, you might need to perform an authoritative restore if an administrator inadvertently deletes an organizational unit containing a large number of users. If

you restore the server from tape, the normal replication process would not restore the inadvertently deleted organizational unit. Authoritative restore allows you to mark the organizational unit as authoritative and force the replication process to restore it to all of the other domain controllers in the domain.

Table 6 lists and describes the authoritative restore commands.

Table 6 Authoritative Restore Commands

Command	Description
Restore database	Marks the entire Ntds.dit (both the domain and configuration naming contexts held by the domain controller) as authoritative. The schema cannot be authoritatively restored.
Restore database verinc %d	Marks the entire Ntds.dit (both the domain and configuration naming contexts held by the domain controller) as authoritative and increments the version number by **%d**. Use this option only to **%d** authoritatively restore over a previous, incorrect, authoritative restore, such as an authoritative restore from a backup that contains the problem you want to restore over.
Restore subtree %s	Marks subtree (and all children of subtree) as being authoritative. The subtree is defined by using the fully **%s** distinguished name of the object.
Restore subtree %s verinc %d	Marks subtree (and all children of subtree) as being authoritative and increments the version number by **%d**. The subtree is defined by using the fully distinguished name of the object. Use this option only to authoritatively restore over a previous, incorrect, authoritative restore, such as an authoritative restore from a backup that contains the problem you want to restore over.

Managing Domains

Ordinarily a user must belong to the Enterprise Administrators group to create child domains and to promote servers as domain controllers. Often the staff member who installs the hardware and software on domain controllers is not the same person who requires high levels of administrative privilege. The domain management commands allow administrators who are members of the Enterprise Administrators group to precreate cross-reference and server objects in the directory.

Table 7 lists and describes the domain management commands used by Ntdsutil.

Table 7 Domain Management Commands

Command	Description
List	Lists all the naming contexts that exist in the enterprise, the schema and configuration naming contexts, as well as all domain naming contexts.
Precreate %s1 %s2	Creates a cross reference object for the domain **%s1** allowing a server named **%s2** to be promoted as the domain controller for that domain. The domain name must be specified by using a fully distinguished name, and the server must be named by using the fully qualified DNS name.

Example using Ntdsutil to find FSMO role holders:

- At a command prompt, type **ntdsutil** then press ENTER.
- Once **ntdsutil** starts, type the following commands within the utility:
- At the Ntdsutil prompt type **domain management** then press ENTER.
- At the Domain management prompt type **connections** then press ENTER.
- At the Server connections prompt type **connect to server *oneofyourDCs***

  Binding to oneofyourDCs ...

  Connected to oneofyourDCs using credentials of locally logged on user
- At the Server connections prompt type **quit**
- At the Domain management prompt type **select operation target**
- At the Select operation target type **list roles for connected server**

  ....

  info for your domain listing the fsmo role holders

  .....
- At the Select operation target prompt type **quit**
- At the Domain management prompt type **quit**
- At the Ntdsutil prompt type **quit**

  Disconnecting from oneofyourDCs ...

Managing Lightweight Directory Access Protocol Policies

To ensure that domain controllers can support service level guarantees, you need to specify operational limits for a number of Lightweight Directory Access Protocol (LDAP) operations. These limits prevent specific operations from adversely impacting the performance of the server and also make the server resilient to denial of service attacks.

LDAP policies are implemented by using objects of the class queryPolicy. Query Policy objects can be created in the container Query Policies, which is a child of the Directory Service container in the configuration naming context.

For example: cn=Query Policies,cn=Directory Service,cn=Windows NT,cn=Services <*configuration naming context*>.

A domain controller uses the following three mechanisms to apply LDAP policies:

A domain controller might refer to a specific LDAP policy. The nTDSASettings object includes an optional attribute queryPolicyObject, which contains the distinguished name of a Query Policy.

In the absence of a specific query policy being applied to a domain controller, the domain controller applies the Query Policy that has been assigned to the domain controller's site. The ntDSSiteSettings object includes an optional attribute queryPolicyObject, which contains the distinguished name of a Query Policy.

In the absence of a specific domain controller or site Query Policy, a domain controller uses the default query policy named Default-Query Policy.

A Query Policy object includes the multivalued attributes LDAPIPDenyList and LDAPAdminLimits. Ntdsutil allows the administrator to set the LDAP administration limits and IP Deny list for the Default-Query Policy object.

The LDAP administration limits (with defaults in parentheses) are the following:

InitRecvTimeout	Initial receive time-out (120 seconds).
MaxConnections	Maximum number of open connections (5000).
MaxConnIdleTime	Maximum amount of time a connection can be idle (900 seconds).
MaxActiveQueries	Maximum number of queries that can be active at one time (20).
MaxNotificationPerConnection	Maximum number of notifications that a client can request for a given connection (5).
MaxPageSize	Maximum page size supported for LDAP responses (1000 records).
MaxQueryDuration	Maximum length of time the domain controller can execute a query (120 seconds).
MaxTempTableSize	Maximum size of temporary storage allocated to execute queries (10,000 records).
MaxResultSetSize	Maximum size of the LDAP Result Set (262144 bytes).
MaxPoolThreads	Maximum number of threads created by the domain controller for query execution (4 per processor).
MaxDatagramRecv	Maximum number of datagrams that can be processed by the domain controller simultaneously (1024).

Table 8 lists and describes the LDAP policies commands.

Table 8 LDAP Policies Commands

Command	Description
Cancel	Cancels any uncommitted modifications of the LDAP administration limits to the default query policy.
Commit	Commits all modifications of the LDAP administration limits to the default query policy.
List	Lists all supported LDAP administration limits for the domain controller.
Set %s1 to %s2	Sets the value of the LDAP administration limit **%s1** to the value **%s2**.
Show values	Shows the current and proposed values for the LDAP administration limits.

Example using Ntdsutil to set LDAP policies:

The default Active Directory Services Interface (ADSI) duration is two minutes. After that period, ADSI queries will stop responding. This is controlled by the MaxQueryDuration setting. You can use the Ntdsutil.exe utility to increase this duration:

- At a command prompt, type **ntdsutil** then press ENTER.
- Once **ntdsutil** starts, type the following commands within the utility:
- At the Ntdsutil prompt type **ldap policies** then press ENTER.
- At the ldap policies prompt type **connections** then press ENTER.
- At the connections prompt type **connect to *yourservername*** then press ENTER.
- Type **q** then press ENTER to exit back to Lightweight Directory Access Protocol (LDAP) policies .
- To display current values, type **show values** then press ENTER.
- Type **set maxqueryduration to *numberofsecondswanted*** then press ENTER.
- Type **commit changes** then press ENTER.
- To display current values, type **show values** then press ENTER.

- Exit by typing **q** press ENTER and typing **q** press ENTER.

Managing the IP Deny List

To provide higher levels of security for the domain controller, you can apply an IP Deny List that prevents the domain controller from accepting LDAP queries from clients with specified IP addresses. Similar to the LDAP administration limits, the IP Deny List only alters the Default LDAP Policy object. The default LDAP Policy is applied to any domain controller that has not had a specific LDAP policy applied to it or to the site in which it belongs.

Table 9 lists and describes the Ntdsutil menu commands on the IP Deny List.

Table 9 Ntdsutil IP Deny List Menu Commands

Command	Description
Add %s1 %s2	Adds an entry to the IP Deny List. The first parameter **%s1** is either the host component or network component of an IP address. If a host component is specified, the second parameter **%s2** is specified as NODE; whereas if the network component is specified, the second parameter is the subnet mask. For example, to deny access from a host with an address of 192.168.100.10, the command is: **Add 192.168.100.10 NODE** To deny access from all hosts with a network address of 192.168.100.0, the command is: **Add 192.168.100.0 255.255.255.0** The entries that you specify by using the add command are not applied until you commit them by using the **Commit** command.
Cancel	Cancels any uncommitted additions or deletions.
Commit	Commits all additions or deletions to the LDAP policy object.
Delete %d	Deletes the specified entry with the index number **%d**. Use the show command to display entries with the respective index number.
Show	Shows all IP addresses that are included in the IP Deny List.
Test %s	Determines whether the IP address specified by **%s** is allowed or denied access to the domain controller. For example, given an IP Deny List entry of 192.168.100.0 255.255.255.0, when tested with an address of 192.168.100.10, access is denied.

Managing Security Accounts

Each security account (users, groups, and computers) is identified by a unique security identifier (SID). Use a SID to uniquely identify a security account and to perform access checks against resources, such as files, file directories, printers, Exchange™ mailboxes, Microsoft® SQL Server databases, objects stored in Active Directory, or any data that is protected by the Windows 2000 security model.

A SID is made up of header information and a set of relative identifiers that identify the domain and the security account. Within a domain, each domain controller is capable of creating accounts and issuing each account a unique security identifier. Each domain controller maintains a pool of relative IDs that is used in the creation of security identifiers. When 80 percent of the relative ID pool is consumed, the domain controller requests a new pool of relative identifiers from the

relative ID operations master. This ensures that the same pool of relative IDs is never allocated to different domain controllers and prevents the allocation of duplicate security identifiers. However, because it is possible (but rare) for a duplicate relative ID pool to be allocated, you need to identify those accounts that have been issued duplicate security identifiers so that you prevent undesirable application of security.

One cause of duplicate relative ID pools is when the administrator seizes the relative ID master role while the original relative ID master is operational but temporarily disconnected from the network. In normal practice, after one replication cycle, the relative ID master role is assumed by just one domain controller, but it is possible that before the role ownership is resolved, two different domain controllers might each request a new relative ID pool and be allocated the same relative ID pool.

Table 10 lists and describes the menu commands for security account management.

Table 10 Security Account Management Menu Commands

Command	Description
Check Duplicate SID	Checks the domain for any objects that have duplicate security identifiers.
Cleanup Duplicate SID	Deletes all objects that have duplicate security identifiers and logs these entries into the log file.
Log File %s	Sets the log file to **%s**. If a log file is not explicitly set, the log file defaults to Dupsid.log.

Using Semantics Database Analysis

Unlike the file management commands described earlier, which test the integrity of the database with respect to the ESENT database semantics, the semantic analysis analyzes the data with respect to Active Directory semantics. It generates reports on the number of records present, including deleted and phantom records.

Note
End users should not use this command except when Microsoft requests them to use it as an aid to fault diagnosis.

Table 11 lists and describes the menu commands for semantic database analysis.

Table 11 Semantic Database Analysis Menu Commands

Command	Description
Get %d	Retrieves record number **%d** from the Ntds.dit.
Go	Starts the semantic analysis of the Ntds.dit. A report is generated and written to a file named Dsdit.dmp.*n*, in the current directory, where *n* is an integer incremented each time that you carry out the command.
Verbose %s	Toggles verbose mode on or off.

List of Menu Commands

Table 12 lists the main menu and submenu commands. The **help**, **quit** and **?** commands are not included.

Note
Menu commands preceded by an asterisk are functional only when the domain controller is operating in Directory Services Restore Mode.

Table 12 Menu Commands

Main Menu Command	Submenu Options
* Authoritative restore	Restore database Restore database verinc %d Restore subtree %s Restore subtree %s verinc %d
Domain management	Connections List Precreate %s %s Select operation target
* Files	Compact to %s Header Info Integrity Move DB to %s Move logs to %s Recover Repair Set path backup %s Set path DB %s Set path logs %s Set path working dir %s
IPDeny List	Add %s %s Cancel Commit Connections Delete %d Show Test %s
LDAP policies	Cancel Changes Commit Changes Connections List Set %s to %s Show Values
Metadata cleanup	Connections Remove selected domain Remove selected Naming Context Remove selected server Select operation target
Popups %s	
Roles	Connections Seize domain naming master Seize infrastructure master Seize PDC Seize RID master

	Seize schema master Select operation target Transfer domain naming master Transfer infrastructure master Transfer PDC Transfer RID master Transfer schema master
Security account management	Check Duplicate SID Cleanup Duplicate SID Connect to server %s Log File %s
* Semantic database analysis	Get %d Go Go Fixup Verbose %s