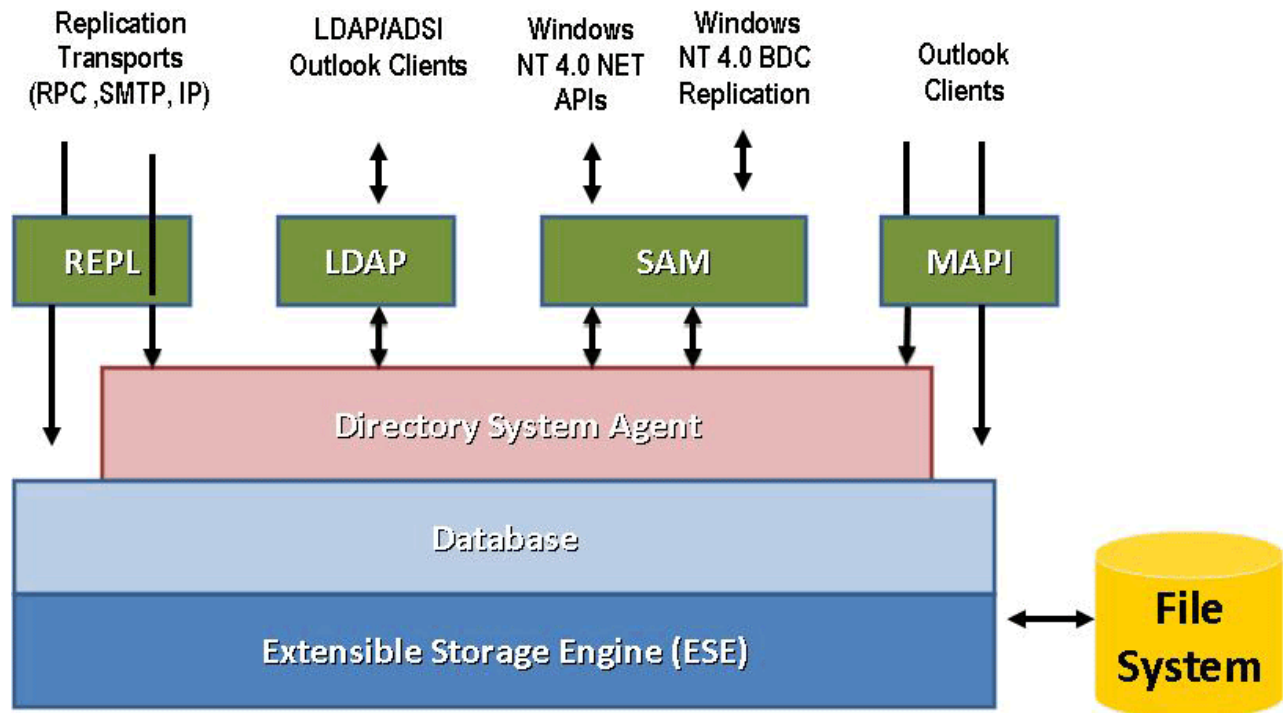


# Active Directory Offline Defrag

Gary Olsen

For the most part, the Active Directory database just works. It performs automatic online defrag, rarely becomes corrupt and never requires manual database recovery procedures like Exchange Server does. Still, it's important for administrators to understand the basics of an Active Directory database – how it works and certain important maintenance procedures.

Figure 1 shows where the database (NTDS.DIT file) sits in the Active Directory architecture. You can see how protocols like LDAP interface with the Directory Service Agent (DSA). The DSA is responsible for functions such as schema enforcement of updates, access control enforcement, object identification, referrals and functional level definition. It is associated with a GUID -- specifically the objectGUID attribute of NTDSsettings object -- that is used in replication to identify replication partners. This is exposed in the Repadmin/showrepl command and is the GUID that is mapped to the server's FQDN in the DNS Alias record.

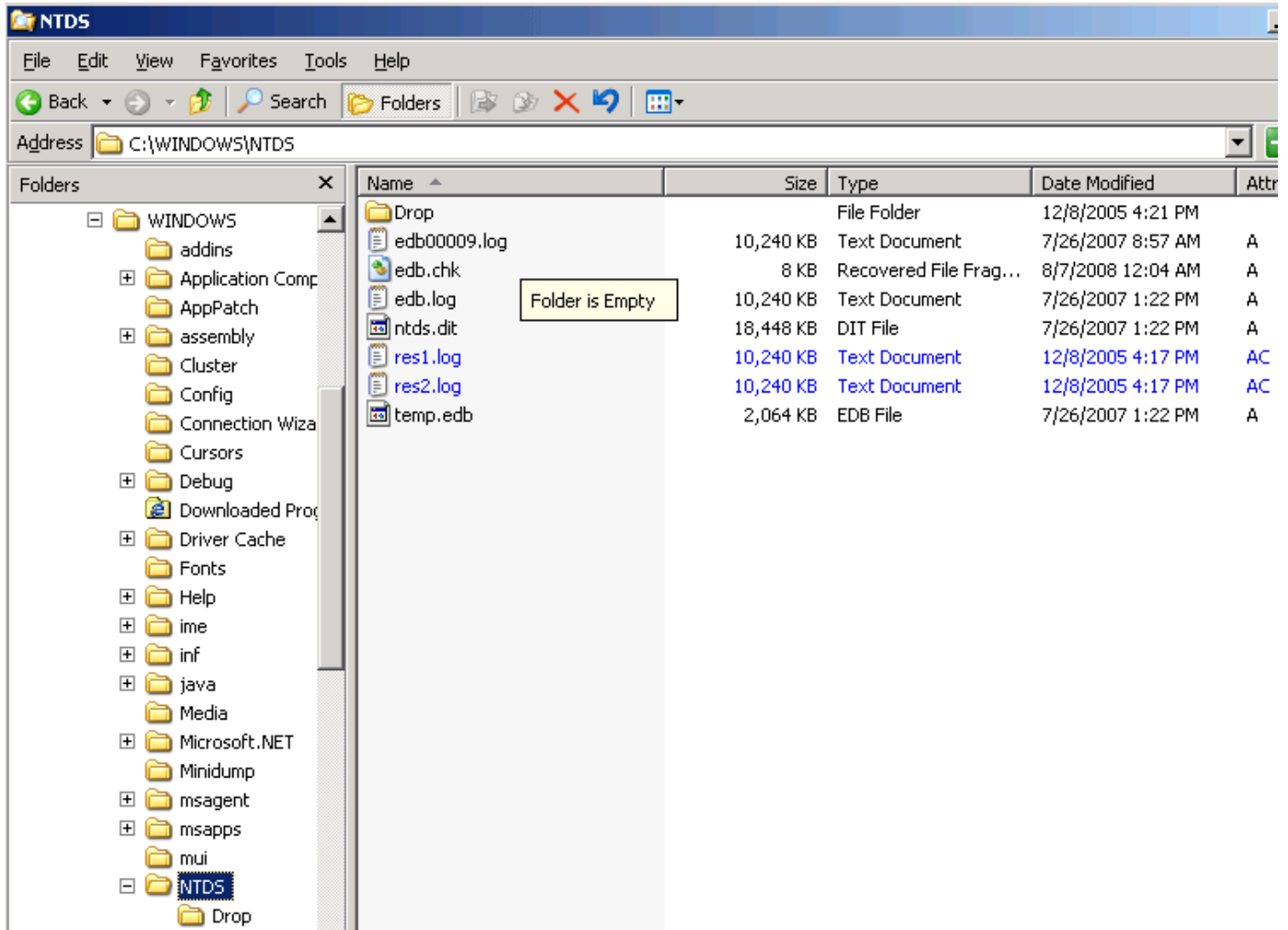


There is also a database GUID that is the invocationID attribute of the NTDSsettings object. The database layer is responsible for the creation, deletion and modification of objects, as well as the retrieval of objects, attributes and the schema cache. The schema defines rules for the organization of the database in terms of classes and attributes.

The NTDS.dit -- located in the %systemroot%\ntds directory -- exists on every Windows server installation. It is a basic transactional Jet database just like Exchange Server, and it is recommended to store the database and logs on separate physical disks. The location for these files is determined during Dcpromo, but you can change the location using the NTDSUtil program described later in this article. Figure 2 shows a typical NTDS directory.

# Active Directory Offline Defrag

Gary Olsen



NTDS.DIT is the database. During Dcpromo, it is enhanced with data from other domain controllers if it's joined to an existing domain or starts a new domain. As objects and attributes are added, deleted or modified, the database gains "whitespace" (unused space).

Adding users, computers, printers and other objects along with defining various attributes will cause the database to grow, and it can be anywhere from a few MB in size to several GB. Being able to load the database entirely in addressable memory will significantly improve operations such as authentication. Therefore it is normally recommended for DCs with 4 GB or more of physical RAM to use the \3 GB switch in Boot.ini to expand the user mode section of memory, which permits more of the NTDS.DIT to fit in there.

Windows Server 2003 made an important change that significantly reduced the database size. While a security descriptor value is stored for every object in Windows 2000 Server, Windows Server 2003 uses single instance descriptors, allowing multiple objects to use a single descriptor.

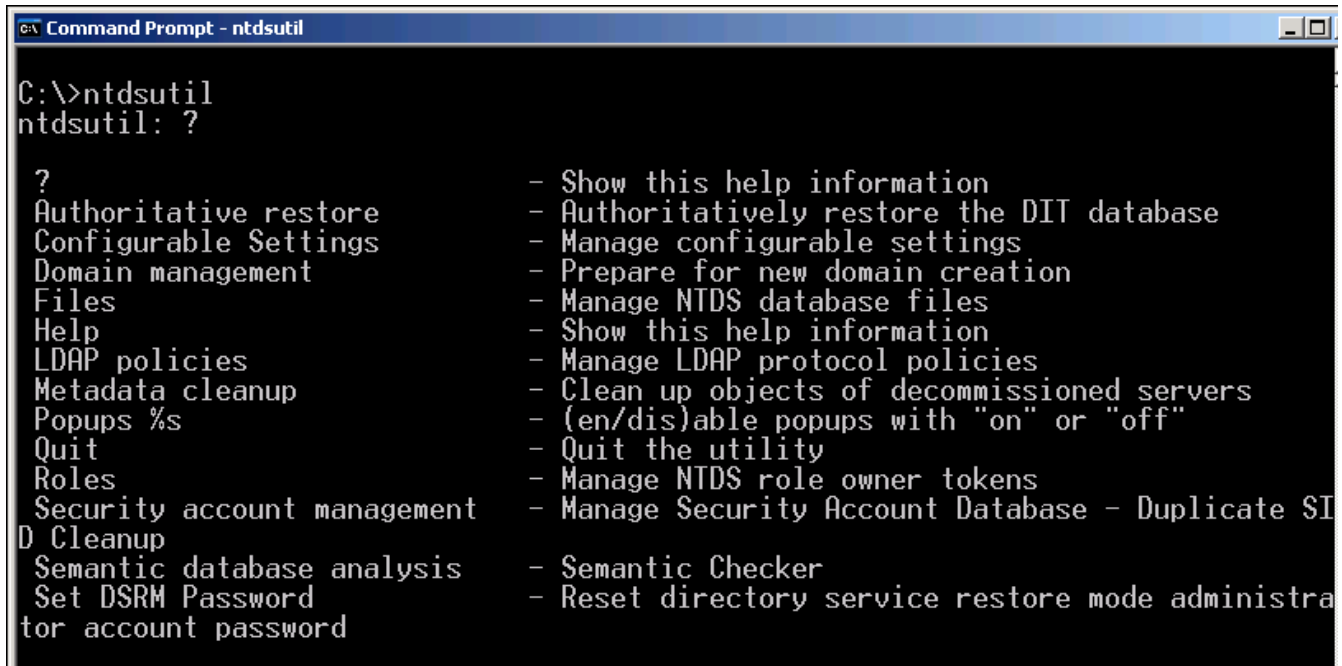
## Defragging The Active Directory Database

It's important to defragment the Active Directory database for best performance. Normally there is an online defragmentation that occurs about twice a day on the database, but this is more of a backup than defrag. It can give admins a false sense of security, thinking that the database has been defragged without any downtime.

# Active Directory Offline Defrag

Gary Olsen

The only way to truly defragment the Active Directory database, remove whitespace and decrease its size is with an offline defrag. This requires you to take Active Directory offline by booting a DC into Directory Service Restore Mode (DSRM), which boots up the DC in safe mode without mounting the AD database. Once booted, enter the NTDSUtil program and you will be able to perform a number of actions that are not possible with Active Directory online, including those in the File menu as shown in Figure 3.



```
C:\>ntdsutil
ntdsutil: ?

? - Show this help information
Authoritative restore - Authoritatively restore the DIT database
Configurable Settings - Manage configurable settings
Domain management - Prepare for new domain creation
Files - Manage NTDS database files
Help - Show this help information
LDAP policies - Manage LDAP protocol policies
Metadata cleanup - Clean up objects of decommissioned servers
Popups %s - (en/dis)able popups with "on" or "off"
Quit - Quit the utility
Roles - Manage NTDS role owner tokens
Security account management - Manage Security Account Database - Duplicate SI
D Cleanup
Semantic database analysis - Semantic Checker
Set DSRM Password - Reset directory service restore mode administra
tor account password
```

Significant functions include:

- Header -- Dumps the Jet database header with information, such as database location, the state (i.e., clean or dirty shutdown, last backup, database size, etc.). This is shown in Figure 4.
- Checksum -- Performs a Jet physical integrity check, a good thing to do before defragmenting. See Figure 2.
- Integrity -- Performs a logical integrity check of the Jet database. After running this command you will see a message recommending that you run the Semantic Database Analysis function. This option is found in the main NTDSUtil menu. This is a good one to run if you get events in the system event log indicating database inconsistency, corruption or other errors. The Semantic Database Analysis function will be covered in more detail in a future article.
- Compact to %S -- This command is the defrag operation that squeezes the whitespace out of the database.

# Active Directory Offline Defrag

Gary Olsen

```

c:\> Select Command Prompt - ntdsutil

Initiating FILE DUMP mode...
  Database: C:\WINDOWS\NTDS\ntds.dit

  File Type: Database
  Format ulMagic: 0x89abcdef
  Engine ulMagic: 0x89abcdef
  Format ulVersion: 0x620,9
  Engine ulVersion: 0x620,9
  Created ulVersion: 0x620,9
  DB Signature: Create time:10/22/2002 18:49:57 Rand:1841498499 Computer:
    cbDbPage: 8192
    dbtime: 542695 (0x847e7)
    State: Dirty Shutdown
  Log Required: 10-10 (0xa-0xa)
  Streaming File: No
    Shadowed: Yes
    Last Objid: 127
  Scrub Dbtime: 0 (0x0)
  Scrub Date: 00/00/1900 00:00:00
  Repair Count: 0
  Repair Date: 00/00/1900 00:00:00
  Last Consistent: (0xA,AC,EF) 07/26/2007 13:22:36
  Last Attach: (0xA,AD,160) 07/26/2007 13:22:38
  Last Detach: (0x0,0,0) 00/00/1900 00:00:00
    Dbid: 1
  Log Signature: Create time:12/08/2005 15:17:27 Rand:1256826 Computer:
  OS Version: (5.2.3790 SP 1)

Previous Full Backup:
  Log Gen: 0-0 (0x0-0x0)
  Mark: (0x0,0,0)
  Mark: 00/00/1900 00:00:00

Current Incremental Backup:
  Log Gen: 0-0 (0x0-0x0)
  Mark: (0x0,0,0)

```

Note that there are other commands available that allow you to move the database and log files, set the online backup directory path and even perform a soft database recovery.

To perform the actual defrag of the Active Directory database, the following steps are required: Create a directory that will temporarily store the compacted (defragmented) database. In our example here, we have created C:\ESE-Backup.

From NTDSUtil, go to the File menu option. In the File Maintenance prompt, enter:

```
Compact c:\ESE-Backup
```

When the compacting operation completes, you will see the following text:

If compaction was successful, you need to:

```
copy "c:\ese-backup\ntds.dit"
```

```
"C:\WINDOWS\NTDS\ntds.dit"
```

and delete the old log files:

```
del C:\WINDOWS\NTDS\*.log
```

## Active Directory Offline Defrag

Gary Olsen

It follows that the next step is to copy the new compacted ntds.dit in c:\ese-backup to c:\windows\ntds\ntds.dit (overwrite the old file), and then delete the 4 logs. You will only have the edb.chk, temp.edb and ntds.dit files left.

Restart the DC in normal mode.

If you want to play with this in a test domain, you can record the size of the NTDS.DIT and then create a large number of users -- say 10,000 -- in the domain. Note the increased size of the NTDS.DIT. Then delete the users and follow the procedure just described to compact the database and replace the old one with the whitespace for the 10,000 deleted users. Compare the size of the new, compacted database with the size after you created the users. You could also wait for an online defrag of the database to occur (recorded with an event in the system event log) and see if that changes the NTDS.DIT size. Note that in a production situation, you want to perform database integrity checks to ensure stability.

It is not necessary to perform this offline defrag on a regular basis, but it's good to do it after significant changes have been made -- such as the removal of a large number of users or groups -- to keep the database at an efficient size.

Previously, I described the basic architecture of the Active Directory database (NTDS.DIT), with details on how and why to perform an offline defrag. I now want to talk about how to handle database errors by looking at some common events that indicate database corruption and how to fix them.

Before getting into the nitty gritty of database repair, let me just say that in my 8+ years of working with Active Directory, I've never had to manually rebuild the database with the ESEUtil.exe tool, as I often did with Exchange Server. While there are Active Directory database errors that pop up occasionally, they are usually pretty easy to resolve.

As mentioned in the previous article, Active Directory uses a Jet database, which is a transactional database. When a change is made to the database, LSASS.exe writes the change to a page in the memory buffer, then writes it to a log file. The default log file is %SystemRoot%\NTDS\Edb.log. The Extensible Storage Engine (ESE) can create a new log file when the current log is filled. LSASS.exe then waits for the log file to be committed to the database (NTDS.DIT).

If Active Directory stops ungracefully, the uncommitted transactions in the log files will be replayed with the transactions committed to disk to make the database consistent. Note that circular logging is enabled by default, which allows data to be overwritten in the existing logs. In the %system%\NTDS directory, you will see the following files:

- Edbxxxx.log (i.e. Edb00009.log) -- This is the log file containing transactions that could not be held in the EDB.log. These are created sequentially.
- EDB.log -- Contains the newest transactions or database changes that have not been committed to the database.
- EDB.chk -- Keeps the database checkpoint and knows which transactions have and have not been committed, so when a recovery is needed, EDB.chk keeps it all straight.
- Res1.log and Res2.log -- Placeholders, 10 MB each, to prevent the disk from being full and having no room to create more log files. If circular logging is enabled, there is no danger of this.
- NTDS.DIT -- The Active Directory database, stored independently on each domain controller.

# Active Directory Offline Defrag

Gary Olsen

All of the database logs are 10 MB in size, while the size of NTDS.DIT depends on the amount of objects stored in Active Directory. Note that there is no physical limit on the number of objects in the database. A colleague of mine once built an Active Directory domain with 100,000 objects (mostly users) and the performance remained pretty flat. It is unknown if there is a limit, testifying to the scalability of Active Directory.

## Database Errors

As previously noted, an ungraceful shutdown of a domain controller will require the database to be rebuilt by LSASS.exe when the DC is rebooted. You may see a message prior to or during logon that says something like "Active Directory is rebuilding indices." This is a notice of database recovery.

While it is technically possible to use ESEUTIL.exe to verify the database and commit the pending transactions in the logs, I have never needed to do that, and having spent the past eight years troubleshooting Active Directory problems for clients, I've never seen nor heard of anyone else having to do it either. Active Directory is quite self-healing.

There are occasions when the NTDS.DIT will become corrupt. Usually "corrupt" is a word you use when you can't figure out what the problem is, but occasionally Active Directory will become genuinely corrupted. In my previous article, I discussed several operations in NTDSUTIL.exe, when booted into Directory Service Restore Mode (DSRM). These commands are in the File Maintenance menu:

- Integrity -- This detects low-level corruption (using ESEUTIL.exe /g).
- Move DB to :\ -- This allows you to move the NTDS.DIT or logs if you experience disk corruption, run out of space or desire to put them on separate drives when DCPromo put them on the same drive.
- Compact to :\ -- This performs an offline defrag of the NTDS.DIT (described in detail in my previous article).
- Recover -- Performs a "soft" recovery of the logs on an unexpected shutdown of the DC (but again I've never seen this have to be done).

## Semantic Database Analysis Checker

This function is located in NTDSUTIL.exe in the main menu, and it must be used offline. Unlike other database operations, I've used this one a lot. Basically any time you see evidence of or suspect database corruption, you can run this tool to fix the problem.

There are a number of events that will indicate database corruption. Some are obvious; some are not. Database corruption could be the cause of Event 1265 (Source: NTDS KCC) or Event 1645 (Source: NTDS Replication), which results in replication failure. Of course these could also be caused by SPN problems, DNS errors, etc., but running the Semantic database analysis is worth a try if you can't find the problem. There is a good description of how to use this option in this Microsoft KB article.

A more obvious error is Event ID 467, Category: Database Corruption. Here are the steps you can take to resolve this and similar events indicating corruption:

- Boot into DSRM and go to NTDSUTIL and go to the File Maintenance menu.
- Run the Integrity command (it will probably confirm database corruption).
- Run the Recover command.
- Run Semantic database analysis with the Go Fixup option.

## **Active Directory Offline Defrag**

**Gary Olsen**

In the case of Event 467, you may see a description that the index is corrupted. In order to repair this, try defragging the database (with the Compact To option in File Maintenance).

If these options fail, it would be advisable to simply demote and re-promote the domain controller. This destroys the old database and gets a fresh copy from another DC. Note that if replication is broken, then you will have to manually demote the DC using the DCPromo/ForceRemoval command.