

# Disaster Recovery: Active Directory Users and Groups

By Gil Kirkpatrick

## At a Glance:

- Mechanics of replication and object linking
- Using NTDSUTIL to back up and restore
- Authoritative and non-authoritative restores

Active Directory is one of the most critical services in a Windows network. To avoid downtime and loss of productivity, it's essential that you have effective disaster recovery plans in place for problems related to Active Directory. This point may sound obvious, but it's amazing how many administrators don't have a plan for one of the most common Active Directory® failure scenarios: accidental deletion of data.

Accidental deletion of objects is one of the most common root causes of service failure. When I do seminars and conferences, I often ask who has had an Active Directory failure due to accidental deletion of data. And every time, nearly everyone raises his hand.

To understand why data recovery is so complex, you first need to understand the following: how Active Directory stores and replicates objects, how it deletes objects, and the mechanics of authoritative and non-authoritative restores.

## Storing Objects

Active Directory is a specialized object database that implements the X.500/LDAP data model. The data store (called the Directory Information Tree or DIT) is based on the Extensible Storage Engine (ESE), an indexed sequential access method (ISAM) database engine. Conceptually, Active Directory stores the DIT in two tables: the data table (which contains the actual Active Directory objects and attributes), and the link table (which contains the relationships between objects).

Each Active Directory object is stored in a separate row in the data table, with one column per attribute. The data table contains all the entries for all of the replicas stored on the domain controller (DC). On a normal DC, the data table contains entries from the domain NC (naming context), the configuration NC, and schema NC. On a global catalog, the data table contains entries for each object in the forest.

Active Directory uses the distinguished name tag (DNT)—a 32-bit integer—to uniquely identify each row in the data table. The DNT, used to refer to objects internally, is much smaller than other identifiers like the distinguished name (DN) and the objectGUID (a 16-byte binary structure). But unlike the objectGUID, the DNT is a local identifier, and is different on each DC.

## How Active Directory Links Objects

Active Directory manages two kinds of relationships between objects in the DIT: the parent-child relationship (also referred to as the container relationship) and the reference relationship (also referred to as the link relationship). To implement the parent-child relationship, Active Directory stores an additional column in the data table called the parent distinguished name tag, or PDNT. This column always contains the DNT of the object's parent.

Each attribute in Active Directory is defined by an attributeSchema object in the Active Directory Schema container. Certain attributes in Active Directory are defined as link attributes, as determined by an even, non-zero value in the linkID attribute of the attributeSchema object. Link attributes establish a relationship between objects in the directory and can be single-valued or multi-valued. The member attribute of a group object is an example of a multi-valued link attribute—it establishes a link between the group object and its member objects.

# Disaster Recovery: Active Directory Users and Groups

By Gil Kirkpatrick

Even though it appears that the member attribute of a group contains the DNs of the members (as displayed by the Active Directory Users and Computers snap-in, for instance), this is not how Active Directory stores them. When you add the DN of a member object to a group's member attribute, Active Directory stores the object's DNT, not its DN. Since the DNT doesn't change, even when an object is renamed, you can rename a user object and Active Directory won't have to sort through all the groups in the system to update the DN in each of the member attributes. This is how Active Directory maintains referential integrity within the DIT. Figure 1 shows a representation, though greatly simplified, of how the data table and link table relate to each other. These tables show that the three user objects—Molly Clark, Alexander Tumanov, and Makoto Yamagishi—are all members of the Senior Engineers group.

**Figure 1 How data and link tables relate**

Data Table			
DNT	PDNT	Object Class	Cn
14529	14401	organizationalUnit	Engineers
14530	14529	Group	Senior Engineers
14531	14529	User	Molly Clark
14532	14529	User	Alexander Tumanov
14533	14529	User	Makoto Yamagishi
Link Table			
Attribute	DNT	Forward Link	
Member	14530	14531	
Member	14530	14532	
Member	14530	14533	

These links are called forward links. Similarly, Active Directory also provides backward link attributes. These provide a reference from the linked-to object back to the object that refers to it, meaning the object with the forward link. The memberOf attribute for users and groups is an example of a back link attribute. The attributeSchema object that describes a back link attribute has a linkID value that is one greater than the even-numbered linkID value of the corresponding forward link attribute. For instance, the member attribute in the Windows Server® 2003 R2 schema has a linkID value of 2; the memberOf attribute that serves as the back link has a linkID value of 3. For more information Figure 2 provides a list of the linked attributes defined by default in the Windows Server 2003 R2 schema.

# Disaster Recovery: Active Directory Users and Groups

By Gil Kirkpatrick

**Figure 2 Link attributes in the Windows Server 2003 R2 schema**

Forward Link Attribute	linkID	Back Link Attribute	Linked
member	2	memberOf	3
manager	42	directReports	43
owner	44	ownerBL	45
siteObject	46	siteObjectBL	47
nonSecurityMember	50	nonSecurityMemberBL	51
queryPolicyObject	68	queryPolicyBL	69
privilegeHolder	70	isPrivilegeHolder	71
managedBy	72	managedObjects	73
hasPartialReplicaNCs	74		
hasMasterNCs	76	masteredBy	77
syncMembership	78		
serverReference	94	serverReferenceBL	95
bridgeheadTransportList	98	bridgeheadServerListBL	99
netbootServer	100	netbootSCPBL	101
frsComputerReference	102	frsComputerReferenceBL	103
fRSMemberReference	104	fRSMemberReferenceBL	105
fRSPrimaryMember	106		
siteLinkList	142		
siteList	144		
msCOM-PartitionLink	1040	msCOM-PartitionSetLink	1041
msDS-NC-Replica-Locations	1044		
msFRS-Hub-Member	1046		
msCOM-UserPartitionSetLink	1048	msCOM-UserLink	1049
msDS-SDReferenceDomain	2000		
msDS-HasInstantiatedNCs	2002		
msDS-NonMembers	2014	msDS-NonMembersBL	2015

# Disaster Recovery: Active Directory Users and Groups

By Gil Kirkpatrick

msDS-MembersForAzRole	2016	msDS-MembersForAzRoleBL	2017
msDS-OperationsForAzTask	2018	msDS-OperationsForAzTaskBL	2019
msDS-TasksForAzTask	2020	msDS-TasksForAzTaskBL	2021
msDS-OperationsForAzRole	2022	msDS-OperationsForAzRoleBL	2023
msDS-TasksForAzRole	2024	msDS-TasksForAzRoleBL	2025
msDS-HasDomainNCs	2026		
msSFU30PosixMember	2030	msSFU30PosixMemberOf	2031
msDS-hasMasterNCs	2036	msDs-masteredBy	2037
msDS-ObjectReference	2038	msDS-ObjectReferenceBL	2039
msDFSR-ComputerReference	2050	msDFSR-ComputerReferenceBL	2051
msDFSR-MemberReference	2052	msDFSR-MemberReferenceBL	2053

Back link attributes are always multi-valued and they're maintained automatically by Active Directory. In fact, you can't directly modify a back link attribute. Even though it appears that you can modify the memberOf attribute of a user or group through the Active Directory Users and Computers MMC snap-in, the snap-in is actually modifying the member attribute of the corresponding group, and Active Directory updates the memberOf attribute behind the scenes. This is why you don't need permissions on the user object to add the user to a group; you are really only modifying the member attribute of the group object. Because each DC manages its back link attributes locally, changes to back links are never replicated. Only the change to the forward link attribute, such as the member attribute of a group, is replicated.

On a normal DC, the data table contains entries for domain objects as well as objects from the Configuration and Schema containers. But some group types can contain references to objects that reside in another domain. How does Active Directory store a DNT for an object that is not in its data table? The answer lies with the Infrastructure Master FSMO (Flexible Single Master Operations) role owner and something called a phantom object.

## Phantom Objects

When you add a member from one domain to a group in another domain, Active Directory automatically creates a special object in the data table called a phantom, which contains the objectGUID, objectSid, and DN of the new member. This provides a DNT that can be stored in the member attribute of the group. If a domain controller is a global catalog, it will not need to create a phantom because it already has an entry in its data table for each object in the forest.

The DC that holds the Infrastructure FSMO role periodically checks the entries in its data table against a global catalog and when it finds that an object has been moved, renamed, or deleted, it updates the phantoms in the data table and replicates the change to the other DCs in the domain. And by virtue of a reference count, the infrastructure master can also remove phantoms that are no longer referred to by any forward link attribute in the domain.

Phantoms allow DCs to manage references to objects in other domains within the forest, but forward link attributes can also refer to objects that are outside the forest—for instance, in a trusted domain. In

# Disaster Recovery: Active Directory Users and Groups

By Gil Kirkpatrick

this case, Active Directory creates an object called a foreign security principal (FSP) in the CN=ForeignSecurityPrincipals container in the domain NC. The FSP contains the foreign object's Security Identifier (SID) and other attributes that identify the object in the foreign domain, but there is no process to ensure that the FSP is kept up to date. For the purposes of data recovery, we treat FSPs as we would any other Active Directory object.

## Deleting Objects

Here, I focus primarily on restoring users and their group memberships. However, the same principles apply to recovering other linked attributes.

When Active Directory deletes an object, it doesn't physically delete the object from the DIT. Instead, it marks the object as deleted by setting its isDeleted attribute to true, which renders the object invisible to normal directory operations. Active Directory removes all attributes that are not designated to be saved, as defined by the schema, and changes the relative distinguished name (RDN) of the object to <old RDN>\0aDEL:<objectGUID>. It then moves the object to the CN=Deleted Objects container for the NC. (There are some classes of objects in the Configuration NC that Active Directory does not move to the Deleted Objects container.) Active Directory removes any forward links to other objects that the deleted object holds—which reduces their reference count in the link table. If there are other objects that contain forward links to the now deleted object, Active Directory removes those links as well.

The resulting object is called a tombstone. Active Directory replicates this tombstone to other DCs, where the same changes are made. Note that Active Directory does not replicate the changes made to forward links that refer to the deleted object. Each DC makes the equivalent change locally, so there is no need to replicate it. This has consequences for recovering group memberships, as I will discuss later in the article.

Active Directory maintains tombstoned objects in the DIT as determined by the tombstoneLifetime attribute of the CN=Directory Service,CN=Windows NT,CN=Services,CN=Configuration,DC=<root domain> object. The garbage collection process on each DC removes tombstones that are older than the configured tombstone lifetime. By default, the tombstone lifetime is 60 days for Windows® 2000, Windows Server 2003, and Windows Server 2003 R2. It is 180 days for Windows Server 2003 SP1.

The tombstone lifetime has a significant bearing on the restore process. You cannot restore from a backup that is older than the tombstone lifetime. Because objects that have been deleted and then garbage collected from the domain no longer have tombstones, the deletion operation will never re-replicate to the restored DC. The deleted objects will then remain on the restored DC as lingering objects and the restored DC will never properly converge with the other DCs in the domain.

## Replicating Objects

Whenever a domain controller performs an update operation of any sort—for instance adding an object or modifying an attribute—the DC assigns a unique 64-bit number to the update operation, called an update sequence number (USN). Active Directory tags the objects and attributes that are updated with the USN to help determine whether they need to be replicated.

Active Directory replicates objects on an attribute-by-attribute basis. That is, if you modify an attribute of an object, Active Directory will replicate just that attribute, not the entire object. To do this, Active Directory keeps track of the changes it makes to each attribute with replication metadata. The replication metadata for an attribute includes:

- The local USN, which identifies the change operation on the local DC.

# Disaster Recovery: Active Directory Users and Groups

By Gil Kirkpatrick

- The invocationID of the DC that originated the change (specifically, the invocationID attribute of the DC's corresponding nTDSSettings object), which identifies a particular generation of the DIT on a domain controller.
- The USN of the original operation as it exists on the originating DC.
- A time stamp that contains the DC system time for when the originating change was made.
- A 32-bit sequential version number that is incremented each time the value is changed.

When a destination DC requests changes from its source DC partner, it sends the USN of the last successfully replicated change to the source DC along with an up-to-dateness vector that includes the largest originating USN the destination DC has seen from each DC that has a replica of the NC being replicated. The source DC uses this information to send only those updates that the destination DC has not already seen.

As the destination DC processes the incoming attribute updates, it checks the version number of each attribute. If the version number of an incoming attribute is greater than the version the DC already has for that attribute, the DC stores the incoming value. If the incoming version number is equal to the version the DC already has, the DC compares the timestamps and uses the attribute with the latest timestamp. If the timestamps are the same, the destination DC chooses the value with the largest invocationID. This guarantees that every DC will eventually settle on the same value for every replicated attribute.

## Linked Value Replication

In Windows 2000, Active Directory replicated multi-valued attributes in the same fashion as single-valued attributes. This caused problems for large, dynamic group objects whose multi-valued member attribute could change frequently on different DCs. If an administrator added a user to a group on one DC and a different administrator added a different user to the group on another DC within the replication latency window, Active Directory would choose the later addition and completely lose the earlier addition. Microsoft addressed this problem in Windows Server 2003 with a process called linked value replication (LVR).

With Windows Server 2003 forest functional level or interim forest functional level, Active Directory replicates the individual values of multi-valued forward link attributes separately, with each value having its own replication metadata. This effectively solves the problem found in Windows 2000 where nearly simultaneous updates of group membership on different DCs could cause data to be lost.

There is one point to be aware of, however. Raising the forest functional level does not automatically fix up existing multi-valued link attributes with the new replication metadata. Only values that are added after raising the forest functional level will have the new metadata. This will have a significant effect on recovering group memberships, as you'll see in a moment.

## Backing Up

Windows includes the very basic NTBACKUP utility, which can be used to perform a system state backup of a DC. The system state of a domain controller includes its registry, SYSVOL, Active Directory DIT files, and critical system files. Most third-party backup utilities also have the ability to backup and restore the system state of a DC.

To perform a system state backup to a disk file, use the following command:

# Disaster Recovery: Active Directory Users and Groups

By Gil Kirkpatrick

```
NTBACKUP backup systemstate /F "<filename>"
```

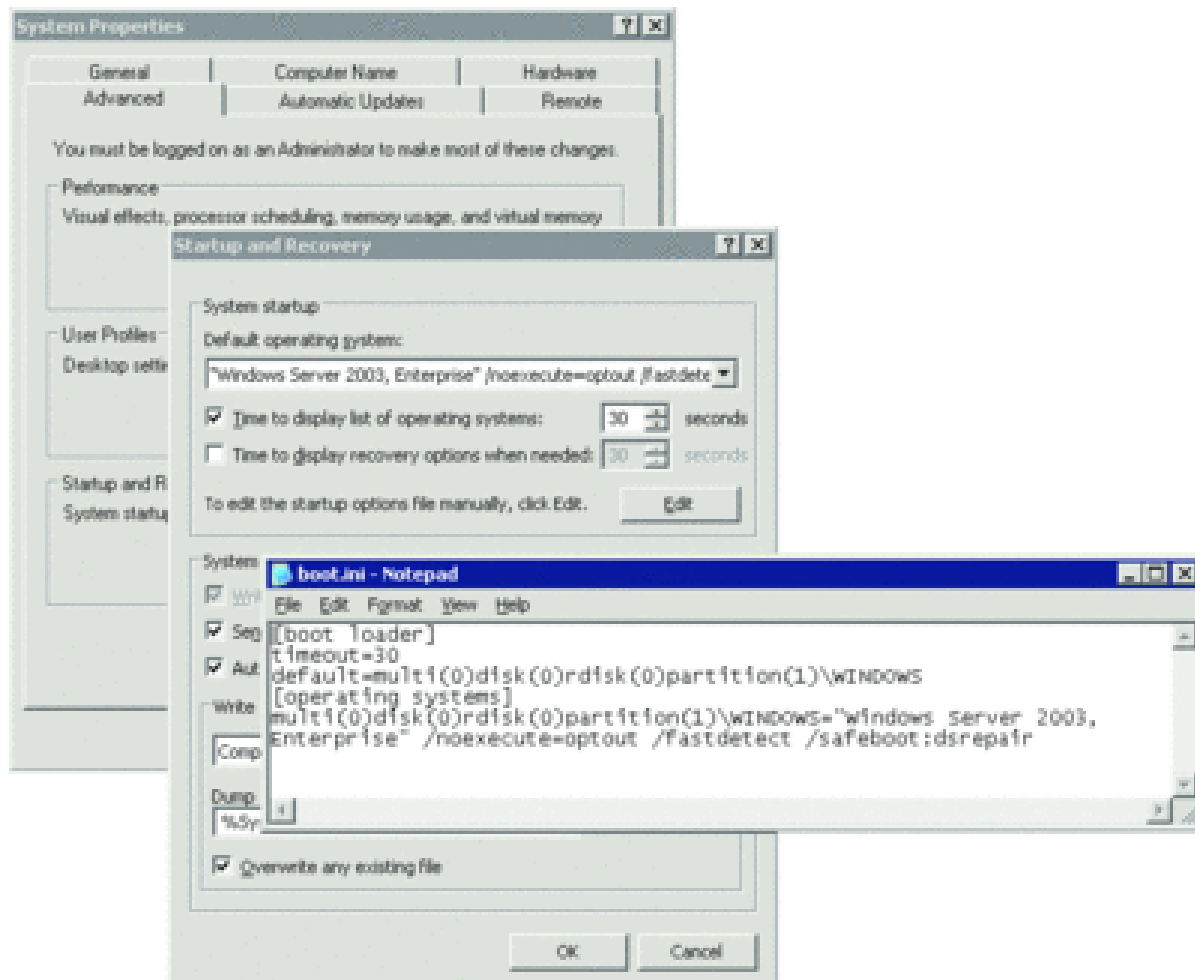
Here, <filename> is the name of the backup file to be created and should use the .bkf extension.

## Performing a Non-Authoritative Restore

Restoring deleted Active Directory objects from backup is a two-step process. First, you reboot the DC into Directory Services Restore mode (DSRM) and then you restore the entire Active Directory DIT from the system state backup using the Windows NTBACKUP utility or an equivalent third-party product. This process will overwrite the entire DIT.

There are two ways to boot a DC into DSRM. If you have access to the system console of the DC, shut down and restart the DC and press F8 when prompted to bring up the Windows boot menu. Select Directory Services Restore from the menu and enter the DSRM password.

If you are managing the server remotely, you won't be able to access the Windows boot menu. Instead, you can change the system boot options by selecting Properties from My Computer, clicking the Advanced tab, and pressing the Settings button located under Startup and Recovery. Press the Edit button in the System startup section to edit the boot.ini file, and add the switch /SAFEBOOT:DSREPAIR to the end of the line, as shown in Figure 3. (For more information about boot.ini switches, see [microsoft.com/technet/sysinternals/information/bootini.msp](http://microsoft.com/technet/sysinternals/information/bootini.msp).)



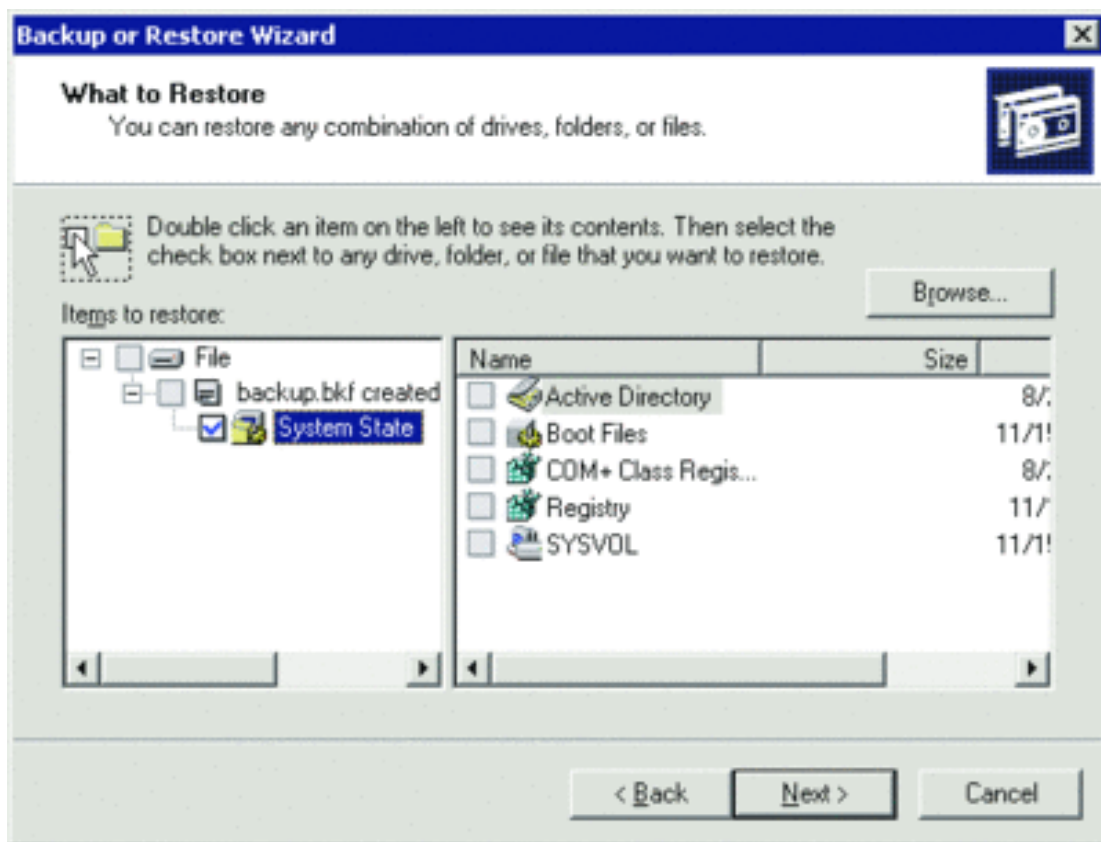
# Disaster Recovery: Active Directory Users and Groups

## By Gil Kirkpatrick

**Figure 3 Setting boot options for DSRM**

When you reboot the server, it will come up in DSRM. Remember that you have to remove the /SAFEBOOT switch from boot.ini when you want to restart the DC in normal mode.

Once you've logged in using the DSRM password, restore the system state backup using the NTBACKUP command again, but without specifying any parameters. (You can't perform a restore using NTBACKUP from the command line.) When the wizard comes up, select Restore files and settings and click Next. Then select the backup file and check the System State box as shown in Figure 4.



**Figure 4 Using the Backup or Restore Wizard to restore system state**

If you were to boot the DC back into normal mode at this point, the Active Directory replication process would bring the restored domain controller back into sync with the other DCs in the domain, and all of the restored data would be overwritten with current data. Clearly, this isn't your goal. Instead, you need a way to force the objects being restored to replicate out to the other domain controllers in the domain.

### Performing an Authoritative Restore

NTDSUTIL also increases the version number of each attribute by 100,000 for each day between the date of the backup and the date of the restore. Unless there are attributes that are being updated

# Disaster Recovery: Active Directory Users and Groups

By Gil Kirkpatrick

more than 100,000 times a day (a pretty unlikely scenario), the version number of the restored attributes will be much greater than the version numbers held by other DCs, and the authoritatively restored object will replicate to the other DCs. The other objects that were restored non-authoritatively from backup will be ultimately overwritten by the existing data from the other domain controllers.

After you've completed the non-authoritative restore, but before you reboot into normal mode, you use the NTDSUTIL program to perform an authoritative restore of the objects you want to recover. Despite the name, authoritatively restoring an object does not "restore" it; it simply ensures that Active Directory will replicate the object to the other DCs. To do this, NTDSUTIL assigns the next available USN to the local USN of the attributes of the object. This causes the object to be sent to replication partners the next time they synchronize. To restore a single object, make sure the DC is booted in DSRM, and follow these steps:

1. Open a command window and type:

**ntdsutil**

2. At the ntdsutil prompt, type:

**authoritative restore**

3. At the authoritative restore prompt, type:

**restore object "<DN of object to be restored>"**

For example, if you want to restore the Molly Clark account from the Eng OU in the DRNET domain, you would enter:

**restore object "CN=Molly Clark,OU=Eng,DC=DRNET,DC=com"**

If you want to authoritatively restore an entire directory subtree, for instance an OU, you would instead enter:

**restore subtree "OU=Eng,DC=DRNET,DC=com"**

(NTDSUTIL also provides a restore database command that authoritatively restores the entire domain as well as the configuration and schema NCs. Restoring the entire domain is fraught with peril and I don't recommend you use that option. If you need to restore an entire domain, you should restore one domain controller and repromote the other DCs in the domain as described in "Planning for Active Directory Forest Recovery",

4. When prompted, confirm that the authoritative restore should increase the version numbers of the respective objects and their attributes.
5. Exit ntdsutil (you'll need to type quit two times).
6. Reboot the DC into normal Active Directory mode.

# Disaster Recovery: Active Directory Users and Groups

By Gil Kirkpatrick

The next time the DC replicates with its partners, the user you restored will replicate out. But restoring the user object is only half the problem. When you introduce object links like those between a group and its members, the situation is more complicated. There are a few fundamental problems you may face during and after the restore, which I will describe in the next few sections.

First, let's review what happens when you delete an object that has back links. Say you delete a user object that is a member of one or more groups. Each domain controller that has a copy of the user object will convert it into a tombstone and remove any references from the link table, thereby removing the user object from any group memberships in the user's domain. (Remember that removing the user from group memberships is not a replicated change since each DC updates the group membership locally. The version number and local USN of the group's member attribute remain unchanged.) A short time later, the phantom objects will be removed from the link tables in other domains, again without updating the replication metadata of the group's member attribute.

When you non-authoritatively restore the DIT on a domain controller in the user's domain, you recover the user object along with all of the group memberships in groups in the domain, so the restored DC is self-consistent. And after you use the NTDSUTIL utility to authoritatively restore the user, the user object replicates out to all the other DCs in the domain.

But because the replication metadata of the current groups in the domain is unchanged, the member attributes of the groups on the restored DC are inconsistent with those on the other DCs. And there is nothing to make them converge on a common state. Thus, the user's memberships will not be restored on the other DCs in the domain.

## **Problem: Group Memberships within the Domain Don't Restore**

Authoritatively restoring the user object does not recover the user's group memberships. Why not? Because the membership relationship is stored and replicated using the member attribute of the group objects (the forward-links), not the memberOf attribute of the user (the back-link). The problem is how to find the user's old group memberships and, once you know them, how to recover them properly.

Microsoft has made incremental improvements to the process of recovering a user's group memberships, so the technique you use depends on the version of Active Directory you are running. The following section applies primarily to Windows 2000 Active Directory.

Determining the user's old group memberships is pretty easy: simply inspect the backlink attribute on the restored DC—in this case, the memberOf attribute of the user object. The memberOf attribute will contain all of the memberships to local and global groups in the user's domain. You can use the Active Directory Users and Computers MMC snap-in (ADUC), or you can use the LDIFDE utility, which is included with Windows Server, to list the restored user's group memberships.

The following LDIFDE command line will list the groups in the DRNET domain that Molly Clark is a member of, storing the results in the output.ldf file:

```
C:\> ldifde -r "(distinguishedName=CN=Molly Clark,  
OU=Engineering,DC=DRNET,DC=Local)" -l memberOf -p Base -f output.ldf
```

Note that you must boot the DC into normal mode to use any LDAP tools and, again, you must disable inbound replication; otherwise the data you restored would be overwritten. The easiest way to disable inbound replication is to use the REPADMIN command:

# Disaster Recovery: Active Directory Users and Groups

## By Gil Kirkpatrick

```
REPADMIN /options <dcname>+DISABLE_INBOUND_REPL
```

Here, <dcname> is the name of the DC you are restoring to. And don't forget to re-enable replication using `-DISABLE_INBOUND_REPL` when you are finished.

If you are recovering only a few users, simply adding the user back to the groups manually using ADUC is pretty easy. If you are recovering more than a few users, there are some tools that can automate some of the process. The Microsoft GROUPADD utility (available from Microsoft Product Support Services) can accept the LDIF file you created to list the user's old group memberships, and in turn generate an LDIF file that recreates those memberships. For instance, you would use this GROUPADD command to process the LDIF file we created in the earlier example for Molly Clark:

```
C:\> groupadd /after_restore output.ldf
```

This command will create a new LDIF file for each domain that Molly Clark had group memberships in with the name `groupadd_<domain>.ldf` (where <domain> is the fully qualified domain name of the domain whose groups will be updated). You would import the LDIF file created above with the following command:

```
C:\> ldifde -i -k -f groupadd_child.drnet.net.ldf
```

With Windows Server 2003, Microsoft improved NTDSUTIL to take advantage of the additional metadata that is present in the member attribute to support link-value replication (LVR). If the restored user object had been a member of any groups in the domain, and the user's group membership was stored with LVR metadata, then NTDSUTIL increases the version number of the corresponding value of the member attribute, which then causes the restored membership to replicate out.

The Windows Server 2003 SP1 version of NTDSUTIL incorporates the GROUPADD functions and will automatically create LDIF files as it performs the authoritative restore of the user object. Figure 5 shows the new version of NTDSUTIL, and Figure 6 shows the contents of the automatically created LDIF file.

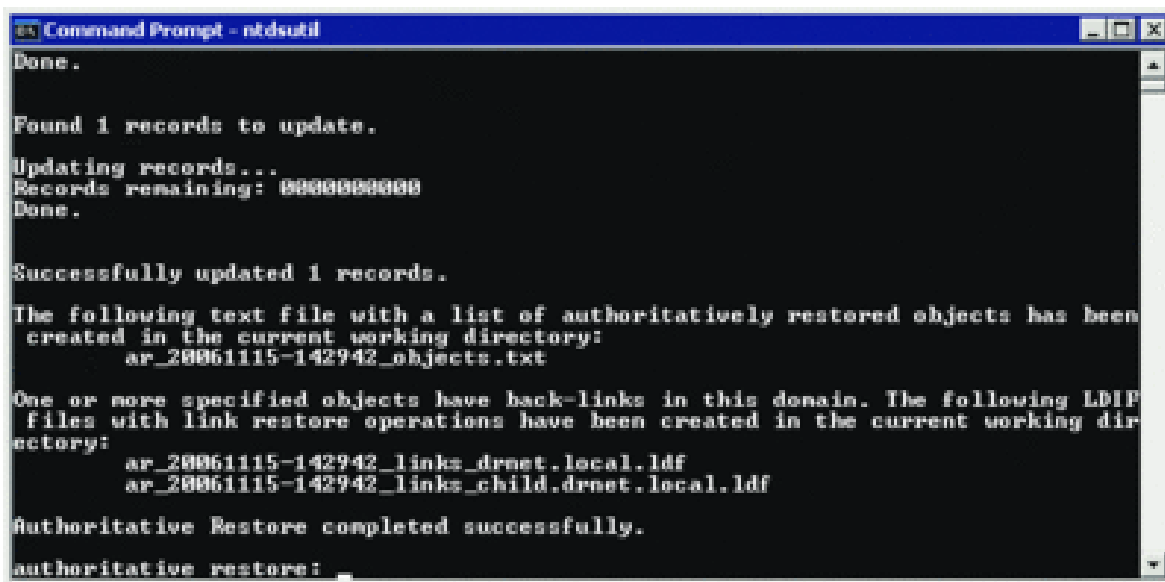
### Figure 6 Contents of LDIF file created by NTDSUTIL

```
dn: CN=EngDL,OU=Eng Groups,DC=drnet,DC=local
changetype: modify
delete: member
member: CN=Molly Clark,OU=Eng,DC=drnet,DC=local
-
dn: CN=EngDL,OU=Eng Groups,DC=drnet,DC=local
changetype: modify
add: member
member: CN=Molly Clark,OU=Eng,DC=drnet,DC=local
-
dn: CN=EngGG,OU=Eng Groups,DC=drnet,DC=local
changetype: modify
delete: member
member: CN=Molly Clark,OU=Eng,DC=drnet,DC=local
-
```

# Disaster Recovery: Active Directory Users and Groups

By Gil Kirkpatrick

```
dn: CN=EngGG,OU=Eng Groups,DC=drnet,DC=local
changetype: modify
add: member
member: CN=Molly Clark,OU=Eng,DC=drnet,DC=local
-
dn: CN=EngUG,OU=Eng Groups,DC=drnet,DC=local
changetype: modify
delete: member
member: CN=Molly Clark,OU=Eng,DC=drnet,DC=local
-
dn: CN=EngUG,OU=Eng Groups,DC=drnet,DC=local
changetype: modify
add: member
member: CN=Molly Clark,OU=Eng,DC=drnet,DC=local
-
```



```
Command Prompt - ntdsutil
Done.

Found 1 records to update.
Updating records...
Records remaining: 0000000000
Done.

Successfully updated 1 records.

The following text file with a list of authoritatively restored objects has been
created in the current working directory:
ar_20061115-142942_objects.txt

One or more specified objects have back-links in this domain. The following LDIF
files with link restore operations have been created in the current working dir
ectory:
ar_20061115-142942_links_drnet.local.ldf
ar_20061115-142942_links_child.drnet.local.ldf

Authoritative Restore completed successfully.
authoritative restore: _
```

**Figure 5 New NTDSUTIL with GROUPADD capabilities built in**

If you are restoring an entire OU that contains a number of users and groups, adding the users back to their groups manually is quite tedious. Another way to recover the restored group memberships is to authoritatively restore the groups themselves.

There are two problems with authoritatively restoring groups, though. The first problem is fairly obvious: if you restore a group, the membership in that group will revert to its state as of the time of the backup. This means that any changes you have made to the group since the last backup will have to be reapplied to the group. The second problem is a little more subtle and has to do with the way Active Directory replication works. After an authoritative restore of both users and groups, there is no guarantee in which order they will replicate out. If a group object replicates to a DC before the restored user object, the replicating domain controller will automatically remove the user reference from the group because the user object does not yet exist on that DC. When the user object replicates in later, it will not be added to the group.

The easiest solution to this problem is to perform the authoritative restore of the groups a second time. After you perform the first authoritative restore, reboot into normal mode and make sure that

# Disaster Recovery: Active Directory Users and Groups

By Gil Kirkpatrick

replication takes place properly. Then reboot back into DSRM and run NTDSUTIL to perform an authoritative restore of the groups the user was a member of. This guarantees that when you boot back into normal mode, the user object will have replicated out before the group objects referring to it replicate.

## Problem: Group Memberships in Other Domains Don't Restore

The "which groups was this user a member of" problem is actually more difficult than I've described. The user you're restoring may have been a member of domain local and universal groups in other domains and those group memberships will not be restored when you do the non-authoritative restore. So how do you know what groups the user belonged to in other domains? The answer is in the global catalog. Along with its own domain's data, the global catalog contains a read-only copy of the data from the other domains in the forest.

To take advantage of the global catalog's forest-wide data, you must perform the non-authoritative restore on a global catalog, which means you must have backed up a global catalog to begin with. Now, when you run LDIFDE to identify the user's group memberships, you can find out the user's universal group memberships from other domains.

When you list the group memberships of the user you are recovering, connect to the global catalog port 3268 instead of the default 389, and specify the root domain of the forest as the base of the search. LDIFDE will display the recovered user's group memberships, including membership in universal groups in all the domains in the forest. Here's how to do this:

```
C:\> ldifde -r "(distinguishedName=CN=Don Clark,  
OU=Engineering,DC=DRNET,DC=Local)" -t 3268 -l memberOf -p Base -f  
output.ldf
```

If you run GROUPADD or the new NTDSUTIL on a global catalog, you will produce one LDIF file for the user's domain, and one LDIF file for each domain in which the restored user was a member of a universal group. When you import these LDIF files, you will restore all the group memberships for the user. Well, almost all—which brings us to the next problem.

## Problem: Recovering Domain Local Group Memberships in Other Domains

There are three kinds of groups in a Windows Active Directory environment. Global groups can only contain members in the same domain, but can be used as a member within domain local groups in its own domain and other domains in the forest. The member attribute of global groups does not appear in the global catalog, but this is not an issue because global groups only contain members from their own domain. Universal groups can contain members from any domain and can be used as members in other universal groups in the forest and in domain local groups in its own domain and other domains in the forest. The member attribute of universal groups is replicated to global catalogs. Domain local groups can contain members from any domain in the forest, but cannot be used as members in groups in other domains. More importantly, the member attribute of domain local groups, like that of global groups, does not appear in the global catalog. The result is that there is no easy way to recover the user's membership in domain local groups in other domains.

Before Windows Server 2003 SP1, the only way to recover domain local group memberships in foreign domains was to restore a DC in each domain, manually search the domain data for any domain local groups that contained the restored user, and then add the user back to the groups you identified. In a large environment with lots of domains, this approach is prohibitively time-consuming.

# Disaster Recovery: Active Directory Users and Groups

By Gil Kirkpatrick

The Windows Server 2003 SP1 version of NTDSUTIL can help. When you run NTDSUTIL on a domain controller, the utility creates a text file that contains the DN and GUID of the restored user objects. Then for each foreign domain, you can non-authoritatively restore a single DC, copy the text file to the DC, and run NTDSUTIL to generate a new domain-specific LDIF file that adds the recovered user back to the domain local groups it was a member of.

To do this, perform the following steps on a DC in each foreign domain:

1. Boot the DC in the foreign domain into DSRM.
2. Use NTBACKUP to restore a copy of the DIT that contains the restored user's group memberships.
3. Copy the .txt file created by NTDSUTIL to the current DC.
4. Open a command window and type ntdsutil.
5. Type authoritative restore.
6. Type create LDIF file(s) from <file name> (where <file name> is the name of text file).
7. Type quit two times to exit ntdsutil.
8. Reboot the DC to normal Active Directory mode.
9. Type ldifde -i -f <ldif filename> (where <ldif filename> is the name of the LDIF file you just created).

And now you have restored all the deleted user's group memberships.

## Step-by-Step

Recovering Active Directory users and their group memberships, particularly in a multi-domain environment, is complicated. The specific steps required to properly recover group memberships depend on the version of Windows you are running.

If you are running Windows 2003 SP1, you would take the following steps:

1. Boot a GC into DSRM and perform a system state restore using a backup that contains the deleted user.
2. Use NTDSUTIL to perform an authoritative restore of the deleted user. NTDSUTIL will create a text file containing the restored object DNs and GUIDs, and one or more LDIF files to restore the user's group memberships.
3. Use LDIFDE -i -f <LDIF filename> (where <LDIF filename> is the name of the LDIF files created in step 2) to import the group memberships in the current domain and other domains.
4. Reboot the global catalog into normal mode.
5. On a DC in each foreign domain, boot into DSRM and perform a system state restore using a backup that contains the group memberships of the restored user.
6. Run NTDSUTIL using the create ldif files command.
7. Reboot the DC into normal mode.
8. Using LDIFDE -i -f <filename> (where <filename> is the name of the LDIF file you created in step 6) to restore the group memberships in the foreign domain.
9. At this point you can optionally force replication with REPADMIN /syncall.

If you are running a version of Windows Server 2003 without SP1 installed, or if you are running Windows 2000, there are some additional steps involved. Since the older version of NTDSUTIL doesn't create LDIF files, use the GROUPADD utility to create them. The process is:

1. Boot a global catalog into DSRM and perform a system state restore using a backup that contains the deleted user.
2. Disable the NIC or unplug the cable to prevent inbound replication.

# Disaster Recovery: Active Directory Users and Groups

By Gil Kirkpatrick

3. Reboot the global catalog in normal mode.
4. Use LDIFDE `-r "(distinguishedName=<dn>)" -t 3268 -l memberOf -p Base -f membership.ldf` to dump the membership of the user with the distinguished name `<dn>`.
5. Use GROUPADD `/after_restore membership.ldf` to create LDIF files.
6. Use LDIFDE `-i -f <filename>` (where `<LDIF filename>` is the name of the LDIF file created by GROUPADD in Step 5) to import the group memberships in the current domain and other domains.
7. Re-enable inbound replication using REPADMIN `/options <dcname> - DISABLE_INBOUND_REPL`.
8. On a DC in each foreign domain, boot into DSRM and perform a system state restore using a backup that contains the group memberships of the restored user.
9. Reboot the DC into normal mode.
10. Using LDIFDE `-i -f <filename>` (where `<filename>` is the name of the LDIF file created by GROUPADD in step 5) to restore the group memberships in the foreign domain.
11. At this point, you can optionally force replication with REPADMIN `/syncall`.

The only thing left now for the pre-Windows Server 2003 SP1 environment is to recover the foreign domain local group memberships for the restored user. Your only choices are to manually restore the domain local group memberships or to restore a DC from backup and authoritatively restore the domain local groups.

## Summary

Even though it's quite easy to accidentally delete users or even OUs from Active Directory, properly recovering the deleted users and their group memberships can be surprisingly complex, time-consuming, and error-prone. To ensure that you can recover from these sorts of disasters as quickly as possible, you have to understand the mechanics of object linking, replication, deletion, and authoritative restores.

Do you think you can get all the steps right the first time you try this in your production environment? To make sure you're ready the next time you have to recover the CEO's user object, have a written plan prepared for recovering deleted objects. And be sure to practice the plan at least once or twice before you have to try it on real data. Your boss (and your CEO) will appreciate it.