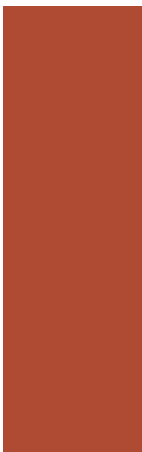


A Definitive Guide to Active Directory Disaster Recovery

**By
Gil Kirkpatrick &
Guido Grillenmeier**



Copyright © 2005 NetPro Computing, Inc. All rights reserved.

This white paper is for informational purposes only. NetPro makes no warranties, express or implied, in this document.

NetPro Computing, NetPro, RestoreADmin and the NetPro logo are either registered trademarks or trademarks of NetPro Computing, Inc. in the United States and/or other countries.

Microsoft, Active Directory, Windows NT, Windows 2000 and Windows Server 2003 are either registered trademarks or trademarks of Microsoft Corporation. Other product and company names mentioned herein may be the trademarks of their respective owners.

NetPro Computing, Inc. • 4747 N 22nd Street, Suite 400 • Phoenix, AZ 85016-4774 • USA

ADDR-WP-1005-100

About the Authors

Gil Kirkpatrick **CTO, NetPro**

A true expert in the design and development of large-scale distributed software for enterprise networks, Mr. Kirkpatrick is the recipient of a 2005 Microsoft Most Valuable Professional (MVP) award and has more than 26 years of software development expertise. Gil is a recognized authority and renowned speaker on commercial network directories, including Banyan StreetTalk, Novell eDirectory, Microsoft Active Directory and Microsoft Identity Integration Server (MIIS), and is the author of Active Directory Programming, published by MacMillan USA.

Guido Grillenmeier **Senior Consultant, Enterprise Microsoft Services, HP Consulting**

Based in Germany, Guido joined HP in 1996 and deals primarily with global Windows 2000/2003 deployments and migrations, designing and implementing efficient Active Directory security and delegation models for HP customers. Guido has further specialized in Disaster Recovery methodologies for AD and is working very closely with Microsoft to investigate and understand this critical task. He is an MVP for Microsoft's Directory Services/Active Directory.

More Information

More information can be found in the books "Windows Server 2003 Security Infrastructures" Part 1 and Part 2 – written by Guido Grillenmeier and Jan De Clercq. These books will be published by Digital Press (Elsevier Reed) in the first half of 2006. They provide insights into the security features and security infrastructure components of the Windows Server 2003 operating system and highlight the security principles an architect should remember when designing and operating a secure Windows Server 2003 infrastructure. The books focus on the security updates that are provided as part of Windows Server 2003 Service Pack 1 (SP1) and the Windows Server 2003 R2 release. Part 1 covers Windows Security Fundamentals (basic security concepts, authentication and authorization services). Part 2 covers Advanced Windows Security Services (including identity management, public key infrastructure, security management services and Active Directory disaster recovery methodologies).

CONTENTS

INTRODUCTION	1
ACTIVE DIRECTORY SECURITY AND COMPLIANCE	2
Business Continuity Planning	2
Active Directory Risk Assessment	2
Common AD Threats	3
Other Ways of Classifying Threats	5
Summary	5
ACTIVE DIRECTORY DATA RECOVERY CHALLENGES	6
How AD Links Objects	6
Managing Object-Links	8
Deleting Objects with Object-Links	9
Viewing Object-Link Attributes	10
Authoritative Restores	11
The Problem with Recovering Objects in AD	14
Additional Challenges	19
How to Prepare Yourself for Enabling a Full Restore of Object-Links	20
Changes in Windows Server 2003 AD with Respect to Recovery of Object-Links	21
SP1 Improvement: Updated NTDSUTIL with Improved Authoritative Restores	24
Summary	27
Tombstone Reanimation	28
Why Recover Tombstones?	30
The Tombstone Recovery Challenge	30
SP1 Improvement: Longer Tombstone Lifetime	32
Group Policy Objects Recovery	33
Automated GPO Backup	36
Summary	36
ACTIVE DIRECTORY SERVICE RECOVERY CHALLENGES	37
Domain Controller Recovery	38
SP1 Improvement: Updated NTDSUTIL for easier server metadata removal	40
SP1 Improvement: Install From Media DCPROMO option retains DNS application partitions	41

Domain Recovery	42
SP1 Improvement: Report if a directory partition has not been backed up recently	43
Forest Recovery	44
REPLICATION LAG-SITES	46
Sample Lag-Site Setup	47
Replication Schedule	47
The Lag-Site DCs	48
Challenges with Lag-Sites	50
SP1 Improvement: Better protection for false restores of DCs as Virtual Servers	51
Summary	53
THIRD-PARTY BACKUP AND RESTORE SOLUTIONS	54
NetPro's RestoreADmin	54
CONCLUSION	56
NETPRO CONTACT INFORMATION	57

INTRODUCTION

Microsoft Active Directory (AD) is the most critical security component in your Windows network, and it has now become an important subject of IT compliance audits. Maintaining AD service continuity in the face of various types of failures, and recovering from those failures quickly and efficiently without extended impact are vital to maintaining a secure and compliant Windows infrastructure.

This white paper describes the some typical failure scenarios for AD and outlines the process for recovering from them.

ACTIVE DIRECTORY SECURITY AND COMPLIANCE

Regulatory compliance for the IT organization is essentially a security problem. The regulations that affect IT, such as the Sarbanes-Oxley Act (SOX), the Health Care Insurance Portability and Accountability Act (HIPAA), the Gramm-Leach-Bliley Act (GLBA), and California Senate Bill 1386 (SB 1386) all require explicit and verifiable controls that ensure the confidentiality, integrity, and availability of IT systems. Because AD provides authentication and authorization services to systems running in the Windows environment, these regulations apply directly to the management of AD.

For instance, SOX requires controls (policies and procedures) that ensure the accuracy of a public company's financial reports as provided to the Securities and Exchange Commission (SEC). If an organization's accounting system runs on a Windows server on the network, AD most likely provides the authentication and authorization services that control access to the accounting data. If the organization's accounting department keeps financial data in spreadsheets stored on network shares, AD controls access to these files as well. If the organization can't ensure that AD is healthy and secure, then the confidentiality, integrity, and availability of the company's financial data is questionable. Obviously, SOX has a clear and defined impact on the way IT organizations manage AD.

Business Continuity Planning

Two of the most important ways to ensure security are through **Business Continuity Planning (BCP)** and **Disaster Recovery Planning (DRP)**. Briefly, BCP is the process of determining how to keep the business functions of an organization running in the event of a significant IT systems failure. An important component of the BCP is Disaster Recovery, which is the process of restoring a failed system to an operational state.

Each business-critical IT system needs a BCP to outline how the business will function if that system fails, and a DRP to show how to restore the system to an operational state. Typically you will develop a single BCP for each system, but you may create several DRPs, depending upon the types of system failures that you expect. The Contingency Planning Guide for Information Technology Systems published by the National Institute of Standards and Technology (NIST) as Special Publication 800-34 provides a detailed description of the BCP and DRP and how they fit into an organization's overall contingency plans.

Developing a BCP requires that you first identify and classify the threats to the environment. The next section briefly explains this process.

Active Directory Risk Assessment

The first step in developing a BCP is to identify and classify the threats that might cause a significant IT systems failure. However, all threats are not created equal and you must also factor in the likelihood of a threat occurring, as well as its potential impact on the business. This provides a risk assessment for each threat, which you can use to evaluate the threats that you need to mitigate.

For example, if you identify an external hacker breaking into the email system as a threat, you might determine that the impact to the business is very high. However, because you already have effective firewalls and VPN systems in place, the likelihood of a hacker breaking in is quite small, so the consequent risk to the business is relatively low. On the other hand, you may determine that an administrator mis-configuring an AD group policy object (GPO) has a medium impact on the organization. But because it is more likely to occur, the risk is considerably higher than that of an outside hacker gaining access to the company's email.

AD domain controllers (DCs) are subject to the same sorts of threats as most computer systems, such as fire and flood, hard disk crashes and the like. But because of its distributed nature, the failure of a single DC is rarely a major problem. On the other hand, a single administrative change to AD can replicate throughout the environment, causing widespread service failure. The risk assessment for AD is therefore somewhat different than typical computer systems.

Common AD Threats

Every AD deployment is unique, and each organization will classify its risks differently. Nonetheless, there are numerous common threats to AD that all organizations must consider. The following illustrates some “typical” AD threats and the risks they may impose. Although this list is not complete, and the mitigation steps and responses are not exhaustive, it provides some guidance into the kinds of threats that you should consider. We cover several mitigation techniques and responses later in this document.

- **Inadvertent Data Deletion (user, OU, computer)**
 - **Description:** A user or a data administrator deletes an important object inadvertently, causing authentication failures.
 - **Probability:** High
 - **Impact:** Severe, but limited in scope
 - **Mitigation:** Depending on root cause, an improved delegation model could help. If simply a “fat-finger” error, improved tools, e.g. command line tools for batch operations, should help.
 - **Response:** Recover deleted items from backup.
- **Administrative Misconfiguration**
 - **Description:** An AD administrator mis-configures some aspect of AD, such as a replication setting or a GPO.
 - **Probability:** Medium
 - **Impact:** Potentially severe; potentially global scope.
 - **Mitigation:** Make sure minimal privileges are granted to administrators. Improve delegation model. Implement a stronger change control process.

-
- **Response:** If discovered soon enough, stop replication. Reapply correct settings, or recover from backup.
 - **Single DC Failure**
 - **Description:** A single DC fails, either because of a software or hardware problem.
 - **Probability:** Medium
 - **Impact:** Low (see next scenario). Authentication will fail over to another DC.
 - **Mitigation:** Use higher quality or redundant hardware. Use proactive system diagnostics and preventative maintenance. Consider multiple DCs for critical sites (e.g., hub sites) and ensure there are at least two DCs per domain
 - **Response:** Rebuild DC from backup, or promote a new DC.
 - **Critical DC failure**
 - **Description:** A single critical DC fails, e.g. the only DC in a central site, or one that holds an important FSMO role, such as the PDC Emulator.
 - **Probability:** Low
 - **Impact:** Medium to high. If the only DC within a site, this could make it impossible for users in the site to logon. If PDC emulator fails, specific functions that rely on this role will fail as well, such as management of Domain based DFS roots.
 - **Mitigation:** Use higher quality or redundant hardware. Use proactive system diagnostics and preventative maintenance. Consider multiple DCs for critical sites (e.g., hub sites) and ensure there are at least two DCs per domain
 - **Response:** Seize or move FSMO role if necessary. Rebuild DC from backup, or promote a new DC.
 - **Site failure**
 - **Description:** All the DCs in a site fail, perhaps because of a data center power outage, or some replicated configuration or corruption problem.
 - **Probability:** Low
 - **Impact:** High
 - **Mitigation:** Stronger change control policies. Redundant data center power sources. Ensure geographic distribution of DCs for each domain in forest.
 - **Response:** Rebuild (or promote) one new DC for each domain in the site. Add additional DCs as needed. Seize FSMO roles as needed.

- **Domain or forest failure**

- **Description:** All the DCs in a domain or forest fail, perhaps because of some replicated configuration or corruption problem.
- **Probability:** Extremely low
- **Impact:** Severe
- **Mitigation:** Stronger change control policies.
- **Response:** Root cause analysis and potentially forest recovery. See the Forest Recovery section of this paper for more information about recovering a forest.

Other Ways of Classifying Threats

There are other ways to classify threats to AD. For instance, you can classify malicious threats by the level of access a user has in AD:

- **Unauthenticated users** are those users who have an IP address inside the firewall, but do not otherwise have any access to AD.
- **Authenticated users** are those that can login to AD, but do not have any rights to make modifications.
- **Data administrators** are authenticated users that have been granted access to update some portion of AD, such as an OU.
- **Service administrators** are authenticated users that have been granted access to configure AD or other critical services.
- **People with physical access to DCs** may not have explicit access to AD, but because they have physical access to the DCs, they can issue various types of attacks on AD.

Classifying the threats this way results in a different list of vulnerabilities and risks that you should consider.

Summary

We've described the notion of a Business Continuity Plan, and shown how a Disaster Recovery Plan fits into the BCP. We've also identified several different situations that would require some sort of DRP to restore Active Directory to a functional state. In many cases, the appropriate disaster recovery response for AD is to restore deleted or corrupted data from backup. While simple in concept, properly restoring all or part of AD from backup is at best tedious and error prone.

The next section discusses the challenges involved in AD data recovery.

ACTIVE DIRECTORY DATA RECOVERY CHALLENGES

Active Directory is a sophisticated distributed, partitioned, and replicated database that is completely integrated with the Windows security system. Restoring lost or corrupted data in Active Directory, while simple in concept, has several pitfalls that make data recovery a risky proposition if you don't understand how Active Directory manages its data.

This part of the white paper describes how objects in AD (Windows 2000 and Windows Server 2003) are linked together and why there are issues when performing a disaster recovery, i.e. an authoritative restore of deleted objects in AD. It also covers what has to be done during the restore of these objects to fully recover all relevant information. Realize that there are quite a few changes with Service Pack 1 (SP1) for Windows Server 2003, which will be taken into account where appropriate.

Windows Server 2003 SP1 – The Security and AD Recovery Service Pack

With the release of Service Pack 1 for Windows Server 2003 late in Q1 2005, Microsoft made several important changes to the core operating system. Besides the very obvious and often discussed changes required to address security issues, quite a few changes were made under the hood to address many other issues as well. This white paper will highlight the various advancements Microsoft has made in the AD Backup and Recovery area with this service pack.

As an over view, the AD Backup and Recovery related changes are listed below. We'll discuss each item in more detail throughout the white paper:

1. Improvements in NTDSUTIL for group membership consistency on authoritative restore
2. Addition of the `siDHistory` attribute to the tombstone object
3. New default tombstone lifetime
4. Report if a directory partition has not been backed-up recently
5. Better protection for false restores of Domain Controllers as Virtual Servers
6. Improvements in NTDSUTIL for server metadata removal
7. Retains DNS application partitions on Install from Media DCPROMO option

Watch for additional sidebars to find the details on these changes in this white paper.

How AD Links Objects

One feature of Active Directory is the ability to implicitly link objects together. This linkage feature, called Object-Links, shows up in several places in Active Directory, but the most well-known example is the linkage maintained between user objects and the groups of which they are members. Each group has a multi-valued attribute named `member` that contains the group's members, and each user object has a multi-valued attribute named `memberOf` that contains the groups the user is a member of. Active Directory automatically keeps these links updated, even after the linked objects are moved, renamed, or deleted.

Object-Links, such as the link between the `member/memberOf` attributes for group membership, are composed of a Forward-Link attribute (the `member` attribute of the group in this case) and a Back-Link attribute (the `memberOf` attribute), and they only exist on objects within the same AD forest. They are stored using the distinguished names of the respective objects to reference each other (e.g. `CN=User1, OU=Accounts, DC=MyDom, DC=com`).

This may seem menacing at first, as the distinguished names (DN) of the objects change when moving objects from one OU to another, or when renaming it. In the example above, when the object User1 in AD is renamed to JoeSmith, this implies that all the user's references would need to be updated

on all linked objects, i.e. all group-memberships of the user would need to be updated with the changed distinguished name.

In reality, the AD database is made up of an object table and a link table. All objects of all naming contexts hosted by a specific DC are stored in the object table along with their distinguished names and a unique identifier called Distinguished Name Tag (DNT), a 32-bit unsigned integer which does not support re-usability. To store links (references) between objects, the link table only uses the DNTs of the objects, which are then resolved to the correct distinguished name of the object when reading the respective linked attribute via LDAP. Essentially, this maintains referential integrity of the objects and their respective links.

A naming context (also referred to as a partition of the database), is a sub-tree of the AD forest hierarchy, however all partitions on a DC share the same object table and link table. Besides the Configuration and Schema naming contexts, not all DCs necessarily host the same domain naming contexts. For example, a DC that is promoted to a Global Catalog (GC) will host all domain naming contexts in a forest, while a normal DC will only host the domain naming context of its own domain. As a result, only a GC's object table will be populated with all objects in an AD forest, but as is well known, it will only replicate and store a subset of the attributes of objects from other domains.

If an AD object such as a group, references another security principal in a trusted external domain (such as a user object in an NT4 domain or in another AD forest), AD creates a placeholder object with the SID of the foreign object in the *ForeignSecurityPrincipals* container of the domain. AD then uses the Domain Name of the placeholder object to reference the foreign object in the Domain Local Group. (e.g., CN=S-1-5-21-1564825003-1728003367-934742191-1815,CN=ForeignSecurityPrincipals,DC=MyDom,DC=com)

Don't confuse this with adding objects from one domain to a group in another domain of the *same* forest, as all objects in the same forest already have a valid forest DN. The challenge in this case is that not all of these objects are known to all DCs, since only GCs store a representation of all of the objects in all domains of a forest, including their DNs. So how would a normal DC be able to add a user to one of its Domain Local groups, when it does not have a representation of that user in its object table, even though this is required as a reference to create the respective links in the link table?

This is where *phantom records* come in to play. Phantom records are used as placeholder entries to allow the storage of the necessary references between objects in a DC's link-table, such as when adding that user from the other domain to the own domain local group. A phantom record merely consists of the respective object's DN, SID and GUID. And yes, the DN of these records and their links in the link table of a DC are what get updated by the mysterious *Infrastructure Master FSMO* role by periodically comparing the phantom records to the relevant "foreign" objects stored in a GC. But contrary to Foreign Security

Principals, phantoms are not real objects – they are really just hidden entries in a DC's object table, and are not easily visible with the exception of viewing a group's foreign domain membership.

Managing Object-Links

When managing Object-Links, you are only managing the Forward-Links, while the DCs themselves take care of maintaining the appropriate Back-Links to the respective objects. Active Directory owns the attributes containing the Back-Links and will not let you edit them. More importantly, AD replicates only the Forward-Link attributes to other DCs (e.g. the `member` attribute of a group-object). AD does not replicate the Back-Link attributes of the respective partner object (e.g. the `memberOf` attribute of a user-object). Each DC re-creates the Back-Link attributes on each DC when storing the links in the link table, after the Forward-Links have been replicated.

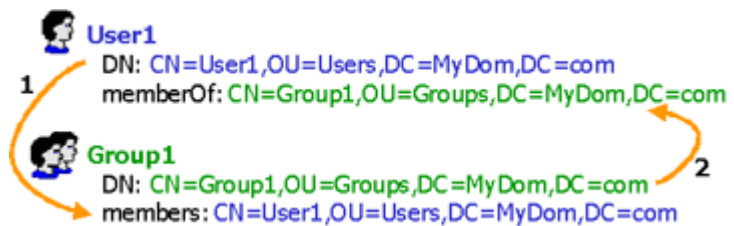


Figure 1: How Group Memberships in AD are Stored

Figure 1 shows the relationship between the `member` and `memberOf` attributes. When adding a user to a group in AD, an administrator only changes the `member` attribute of the group (adds the Forward-Link of the user to the multi-valued `member` attribute of the group). In this case, AD stores the distinguishedName of the user (CN=User1,OU=Accounts,DC=MyDom,DC=com) in the `member` attribute of the group. AD also increments the version number of the group's `member` attribute by 1. At the same time, the AD stores the next Update Sequence Number (USN) of the DC with the attribute, so that this attribute change will be replicated to other DC/GCs in AD^[1].

It's important to realize that Windows 2000 treats all changes to the multi-valued `member` attribute of a group the same way as a single valued attribute (such as description). So if Group1 already contained 1,000 members, the addition of User1 would trigger replication of the whole `member` attribute with its 1,001 links to the other DCs in the domain. Windows Server 2003's version of AD is enhanced by adding version-numbers and USNs to each value in the `member` attribute of groups, leveraging a new feature called *Link-Value-Replication* (LVR). In this case, AD only replicates the link of the added user

^[1] When the value of an attribute in AD is changed, both its version number as well as the Update Sequence Number (USN) of a DC (which changes with any change made on the DC) are incremented by 1. It is the change in USN number that causes replication, not the version number. Version numbers are used to indicate which value is authoritative in case of a replication conflict. The value with the higher version wins.

object. More details on Windows Server 2003 and LVR will be covered later in this paper.

After you updated the `member` attribute of Group1 by adding User1 to the group, AD automatically updated the respective `memberOf` attribute of the user-object with the Back-Link of the group (`CN=Group1,OU=Groups,DC=MyDom,DC=com`). The other DC/GCs in the forest do the same once they have received the updated group-membership information through the replication process.

Similarly, when you remove User1 from Group1, you really only change the Forward-Link in the Group-object. Again, this results in AD incrementing the **version** number and **USN** of the group's `member` attribute, which forces replication of this change to other DC/GCs. The group's Back-Link to the `memberOf` attribute of the respective user object is automatically removed by the DCs.

Administrators are easily fooled by the AD Users & Computers (ADUC) snap-in when managing a user's group-membership via the `memberOf` tab of the user-object because it looks like they are directly adding a group to the user's `memberOf` attribute, which would be the Back-Link. In reality, the snap-in is adding users to the `member` attribute of the group, which is the Forward-Link. This also explains why administrators do not need to have any permissions to manage a user when adding them to a group – instead, they only require sufficient permissions to modify the `member` attribute of a group.

Deleting Objects with Object-Links

When AD deletes an object, the mechanism to update Object-Links works differently. In most scenarios, an administrator deletes a user for a good reason, such as six months after the user has left the company. In this case, let's assume the user belonged to a group in the same forest.

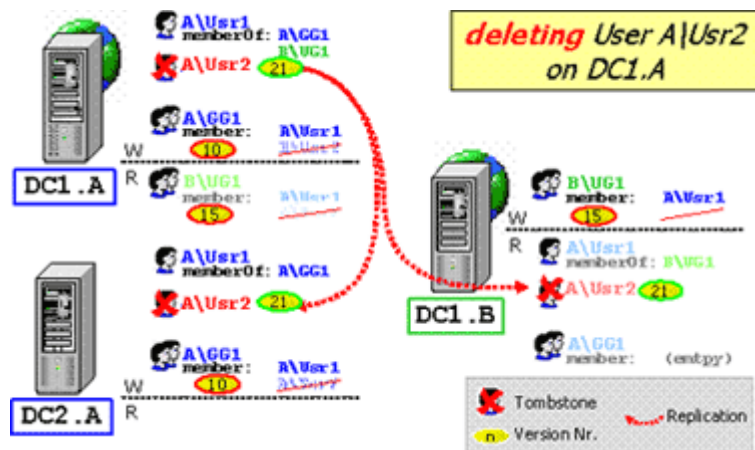


Figure 2: Deleting a User in AD

At the time of deletion, AD does not fully remove the user-object, but instead it removes most of the attributes of the object (except for those the Schema indicates should be retained. Refer to the chapter on Tombstone Reanimation for more information this.) AD renames the deleted object to `<oldname>\0ADEL:<GUID>` and moves it to the hidden Deleted Objects system container of the partition. This is called a *tombstone*. AD increments the **version** number and USN on the deleted **user object**, so that this tombstone will be replicated to other DC/GCs where the object will be deleted ^[2].

Lets look at a slightly more complicated example. Figure 2 above shows two DCs from domain A and one from domain B (*the naming of the objects and their links has been adjusted to allow better visibility*). Prior to the deletion of Ustr2 in domain A, this user belonged to domain A's global group GG1 as well as domain B's universal group UG1. Once the user-object is deleted, i.e. converted to a tombstone object, the internal clean-up process on each DC removes the user object's references from the link table on the DC, so that the deleted object is no longer referenced. This ensures that no stale Object-Links remain within the AD forest, as displayed in Figure 2. This includes updating the Forward-Links of all objects linked to the deleted object (e.g. the group's `member` attribute). **Note, that although this is a change of the group's member attribute, AD does not update the version number on this attribute.** Thus, the group's `member` attribute has the same version number on a DC that knows of the user-deletion as one that has not yet received the tombstones, even though their content is not the same. This fact is very important with respect to the restoration of deleted objects in AD. The groups will be in sync again once the tombstones replicate to the other DCs.

Viewing Object-Link Attributes

Every attribute of an object in AD is defined by an entry in the Active Directory schema. The schema entry determines exactly how AD should handle the attribute in terms of replication, deletion, LDAP queries, and so on. The schema object for attributes that are used as part of an Object-Link contain an attribute named **linkID**. The schema definition for Forward-Link attributes will have an even number (n) in the linkID, while the matching Back-Link attribute will have an odd number (n+1) as the linkID.

In the user/group example, the `member` attribute of a group has a linkID of 2 while the `memberOf` attribute (typically a user, or a group) has a linkID of 3. This shows that the attribute containing the Forward-Link is identified by the even value of n (2 in this case), while the corresponding attribute containing the Back-Link is identified by the odd n+1 value (3 in this case).

^[2] Sometime later, after the tombstone's lifetime has expired (60 days by default), the garbage collection services on each DC finally removes the object from the AD database.

There are two more important attribute-pairs containing Object-Links that are relevant for administering objects in AD and are important to review with respect to disaster recovery of AD objects:

- **manager + directReports**

- Can be used to store the relationship between managers and their team-members. The manager must be a user or contact from within the same AD forest.

- Link-Info:

```
manager (linkID = 42, Forward-Link)
directReports (linkID = 43, Back-Link)
```

- When AD deletes a user or contact object (a manager), it will clean up the "manager" attribute of each member of the team automatically.

- **managedBy + managedObjects**

- Can be used to store information on which user (or contact) is responsible for which resources (groups, OUs, computers etc.) in AD. The responsible person of the resource must be a user or contact from within the same AD forest. The managedObjects attribute is not visible in the GUI of the user or contact object, but can be viewed via LDAP queries.

- Link-Info:

```
managedBy (linkID = 72, Forward-Link)
managedObjects (linkID = 73, Back-Link)
```

- When AD deletes a user or contact object that has been configured to manage various resources, it cleans up the managedBy attribute of the respective resource objects as well.

Authoritative Restores

To recover from accidental or malicious deletions of objects in AD, Windows 2000 and Windows Server 2003 allow the restore of the entire AD database or any part of it by performing an *authoritative restore*. An authoritative restore of AD objects basically increments the version number of an object and all of its attributes by 100,000 for every day between the time of backup and the time of restore. If the backup was performed on Monday and a restore with this backup was performed on Friday, AD would increase the version number on any authoritatively restored objects by 4 x 100,000. Under normal circumstances^[3], this will guarantee that the object has a higher version number than the ones currently existing on other DCs, and will ensure the replication of the authoritatively restored object to all other DCs, overwriting the tombstone of the object. Note that if you authoritatively restore an object that has not been

^[3] An abnormal circumstance would be, that the object or any attribute of the object to be restored would have changed more than "100.000 x days since backup" times on other DCs in between the last replication of the database on the DC, where the authoritative restore is being performed.

deleted, the restored object will overwrite any unreplicated changes made to the object on other DCs. Therefore it is a best practice to authoritatively restore only those objects that have actually been deleted. If the entire domain database were to be authoritatively restored, it would overwrite any unreplicated changes performed on other DCs.

You use the NTDSUTIL command line tool in **Directory Services Restore Mode** (DSRM) to perform an authoritative restore. Performing the authoritative restore requires that the objects and all of the attributes to be restored already exist within the AD database.

One way to ensure the deleted data does exist in the AD database of a DC, is to use a DC that has not yet replicated the deletion of the objects due to replication latency to that DC and rebooting this DC into DSRM mode. In this case, no system-state restore is required (see section on Replication Lag Sites for more information).

Another way you can accomplish this is by using a backup system to recover a previous version of the AD database on a previously backed up DC, i.e. by performing a system-state restore of the DC. This is what's referred to a non-authoritative restore. The Microsoft NTBACKUP tool that comes with the OS allows easy backup and recovery of the system-state of a DC. Performing a system-state backup will ensure backup of all critical system files including the registry, the AD database files and the SYSVOL folder. You can automate these backups by using the command-line interface of NTBACKUP, for example:

```
NTBACKUP backup systemstate /F "D:\Backup\DC1_SysState_2005-09-15.bkf"
```

To perform a system-state restore, you must first boot the DC into Directory Services Restore Mode. Realize that after booting a DC into DSRM mode, the Restore Password is required to logon to the machine – this password is initially set when a server is promoted to a DC. If unknown now, it can be reset at the command-prompt prior to booting the DC:

Windows 2000	c:\setpwd ⇨ <enter new PW>
Windows Server 2003	C:\NTDSUTIL ⇨ set DSRM password ⇨ reset Password on server NULL ⇨ <enter new PW> ⇨ <confirm new PW> ⇨ q ⇨ q

The DC is booted into DSRM mode either by manually pressing F8 during the boot-sequence and choosing the appropriate boot option, or by adding the option `"/safeboot: dsrepair"` to the appropriate line in the boot.ini file of the DC prior to booting it.

TIP: Editing the boot.ini file and adding the `"/safeboot:dsrepair"` option to the boot-entry allows an unattended boot sequence of the DC into DSRM and can thus be performed completely remotely.

You then either logon to the DC's console directly or via Terminal Services, using "administrator" as the username along with the DC restore password. Finally, you can perform the non-authoritative restore via NTBACKUP using the UI. Ensure you do NOT reboot the DC after the system-state restore has finished.

The actual routine to perform an **authoritative restore** after either having restored the AD data base from tape or file, or by booting a "good" DC (one which hasn't yet replicated the tombstones of the deleted objects) into DSRM mode, is as follows:

1.	DC must still be in Directory Services Restore Mode.
2.	Open the Command Prompt.
3.	Type <code>ntdsutil</code> <enter>
4.	Type <code>authoritative restore</code> <enter>
5.	<div>Type <code>restore subtree part_to_restore</code> <enter> Where: <i>part_to_restore</i> is the Distinguished Name of the AD object that needs to be restored. For example, if the "Accounts" OU in the mydom.com domain is to be restored, the command is: <code>restore subtree "OU=Accounts,DC=MyDom,DC=com"</code> <i>Note: It is also possible to restore single objects by using the restore object command in the same manner. Furthermore, the entire AD database can be restored authoritatively by using the restore database command, but this is not recommended and should only be used as a last resort.</i></div>
6.	Acknowledge the authoritative restore to increase the version numbers of the respective objects and their attributes.
7.	Exit ntdsutil by typing <code>quit</code> <enter> twice.
8.	Reboot the DC to normal AD mode.

When rebooting the DC into normal mode, the objects you authoritatively restored will replicate to the other DCs in the Domain and all GCs in the forest, overwriting any tombstone objects created by the deletion. Although this is standard, there is more to the story.

The Problem with Recovering Objects in AD

There are two types of problems with authoritatively restoring objects that contain links to other objects in AD:

1. When authoritatively restoring objects containing Forward-Links (such as groups and their `member` attribute), the links are only successfully replicated out, and thus restored in the domain, if the referenced object exists on the DC that replicates in the link values.
2. When authoritatively restoring objects containing Back-Links (such as users and their `memberOf` attribute), the appropriate Forward-Links in the linked objects are not restored correctly^[4].

The first problem relates to the fact that we cannot guarantee the replication order of objects that need to be replicated between DCs. If both objects of a linked-pair are deleted and authoritatively restored, and the object containing the Forward-Link is replicated first to another DC before the referenced object is replicated, the creation of the Forward-Link will fail during replication. This can be resolved by repeating the authoritative restore of the objects to invoke another replication. You must perform the second authoritative restore operation on the same DC you used for the first restore, but instead of restoring the DC's system state from tape, only repeat the authoritative restore operation so as to increase the version numbers of the same objects once more.

- The following Technet article available from Microsoft explains the first problem and how to resolve it:

[Q280079 - Authoritative Restore of Groups Can Result in Inconsistent Membership Information Across Domain Controllers.](#)

- The second problem was just recently addressed by Microsoft, but still lacks a valid solution to the problem:

[Q84001 – How to restore deleted user accounts and their group memberships in Active Directory.](#)

Because the first problem is easily resolved by performing the authoritative restore twice, this white paper mainly concentrates on the second problem, what it means, and how it can be resolved.

What does this mean?

For example, when restoring a user object that was a member of several groups in its own domain or in other domains of the same AD forest (i.e., containing `memberOf` Back-Links), the authoritative restore will not correctly recover the group-memberships of this user (i.e., the member Forward-Links of the group-objects).

^[4] As mentioned before, there are some changes in Windows Server 2003 AD, which minimize this impact somewhat, but depending on the domain structure, the issue still remains. See section "Changes in Windows Server 2003 AD with respect to recovery of Object-Links" for details.

This means that restoring a user or group object from an authoritative backup will not restore its membership information. This is true for Domain Local Groups, Global Groups and Universal Groups, although the nature of the groups has a direct impact on the efforts involved to solving this problem, as we will see. When Exchange 200X comes into play, all Distribution Lists are mail-enabled groups in AD (usually mail-enabled Universal Groups). Thus, an unsuccessful recovery of the group-memberships will have a direct impact on most messaging infrastructures based on Exchange 200X.

The same is true for a user who is configured in AD as the manager of a team. If this user is deleted and then restored to AD, the information regarding which team members the user manages is not correctly recovered. Furthermore, if users are defined as being responsible for certain resources in AD by setting the managedBy attribute on those objects, these relationships are also lost after authoritatively restoring the user object. This does not directly impact the security of the objects in AD, but it may impact your business workflow if you leverage this information through other processes within your company (e.g. to generate corporate Org-Charts based on the directReports information stored in AD).

From a security and administrative perspective, the loss of the group-membership is by far the most critical for an AD infrastructure.

Why does this happen?

As detailed previously, when an object with a Back-Link gets deleted (e.g. a user who is a member of groups in the same forest), a referenced object's Forward-Link (e.g., the `member` attribute of a group) is cleaned automatically **without changing the version number** on the respective attribute.

When a restore of an AD database is performed from tape, we perform a **non-authoritative** restore of the entire AD database on the respective DC (you can only restore the whole NTDS.dit file). You do this while the DC is booted into directory restore mode.

After the database restore, the DC contains an "older" version of the domain NC, the configuration and schema NC. If the DC was also a GC, the read-only partial replicas of the other domains in the AD forest are also restored to an "older" version. An obvious prerequisite for a successful authoritative restore of objects, the backup operation needs to have been performed prior to the deletion of objects in AD. This is the case in Figure 3.

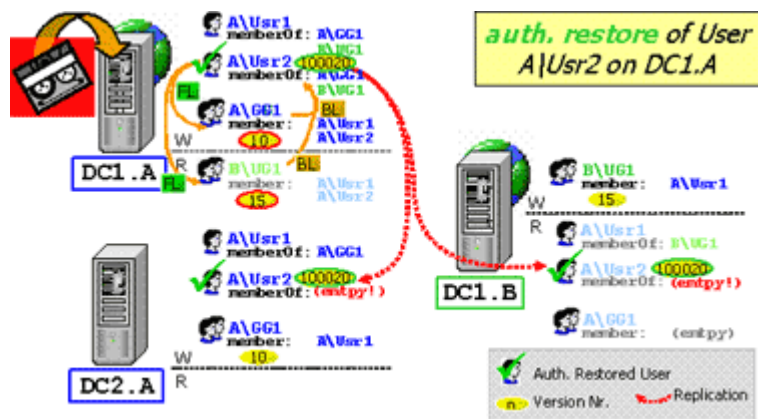


Figure 3: Authoritative Restore of User Object in AD

The DC with the restored AD database also contains the group objects with the correct `member` attributes (Forward-Links), as well as the user objects with the correct `memberOf` attributes (Back-Links), as shown on DC1.A. Assuming that the backup used for the AD database restore was not very old, it is unlikely that there were any major changes to the group memberships in the domain between the last backup and this restore. At this point, the USNs of the groups haven't changed, and the version numbers of their `member` attributes on the restored DC are identical with the version numbers of the `member` attributes on the other DCs that have replicated the user tombstones. The problem, however, is that the other DCs have "cleaned" the user object from their groups, causing the membership information of the groups to be inconsistent between the DCs (as shown on DC2.A and DC1.B in Figure 3). Essentially, the fact that the Back-Links are not replicated between DCs is the real cause of this problem.

At this point, you run the NTDSUTIL command to authoritatively restore the deleted user objects. Because the groups have not been deleted, you only authoritatively restore the deleted user objects. AD will only increment by 100,000 the version numbers on the user objects and their attributes (assuming the restore was performed from today's backup).

After you reboot the DC in "normal" mode, it will replicate with other DCs and GCs of the forest. AD will replicate the user objects to the other DC/GCs, thus overwriting the tombstones. The `memberOf` attribute is not replicated, but is calculated separately by each DC. And although the user's group membership exist correctly on the restored DC, AD will not replicate the "recovered" group memberships to the other DCs, because the USN for this group (the `member` attribute) has not changed since the backup. In fact, all DCs will have the same version number in the `member` attribute of the group objects. As a result, the

authoritatively restored users will only be a member of the default **Domain Users** group on all other DCs and GCs within the AD forest^[5].

A Solution to the Problem

The quick solution that comes to mind is to perform an authoritative restore of the objects that contain the Forward-Links (the groups in this case), along with objects that contain the Back-Links. After all, it should be easy to authoritatively restore all the groups which the deleted users belonged to, as long as you are authoritatively restoring the users themselves.

But *which* groups? Typically users will be members of groups throughout the enterprise, not only within OUs that may belong to a specific location or organizational part of a company. In multi-domain environments the users will very likely also be in groups from the other domains. This is especially true for mail-enabled groups leveraged by Exchange 200X as distribution lists.

If you don't know, which groups the deleted users were a member of, you potentially need to authoritatively restore ALL groups in every domain in AD. This would include restoring a DC of every domain prior to being able to perform the authoritative restore of the groups on the respective DC. Obviously, this is not realistic in a large enterprise.

The good news, however, is that you DO know most of the groups that a deleted user belonged to after you have performed the restore of the AD database on a DC/GC of the user's domain. To ensure success, the restored DC needs to be a **GC**^[6] so that the read-only partial replicas of all other domains in the AD forest are also restored on this machine. The DC will then contain most of the required information to perform a full recovery of the linked data in the AD forest. As a lone exception, you cannot use a GC to retrieve the members of the Domain Local Groups of the remote domains, as AD does not store these as part of the partial attribute set of a GC.

Nevertheless, the recovered GC does contain the membership information of the domain's Local and Global Groups, as well as all Universal Groups of the forest. You can use this information to recover the original group memberships of the recovered user accounts (or nested groups).

This is where the Back-Links come in. The `memberOf` attribute of every user-object contains the Back-Link reference of each group that a user belongs to.

^[5]The **Domain Users** group is a special group in AD, as it neither stores the users in the member attribute of the group nor does it have a real Back-Links to the User-Object. This is due to the fact, that – by default – it is set as the "Primary Group" for every user-object in AD. The membership of the Domain Users group is not read from the member attribute, but instead is calculated whenever it needs to be enumerated (e.g. when viewing the group-memberships via the GUI). This is interesting, as this fact allows the Domain Users group to contain many more users than usually supported for groups in W2K AD (an W2K AD group can usually not contain more than 5.000 users – but the Domain Users group contains all users of a domain, which can be thousands). The same is true for the **Domain Computers** and the **Domain Controllers** groups.

^[6] If you were restoring a normal DC, it would not contain the Universal Group memberships from other domains in the forest.

These are stored as the distinguished name of the groups, and you can read them from the user-objects via LDAP queries. The same is true for the `directReports` and `managedObjects` attributes, which contain the Back-Links of the team-members of a manager and the AD objects managed by a user or contact.

Using the Restored DC to Recover Object-Links

In order to read the AD database of a restored DC via scripts, you must reboot the DC into the "normal" AD mode. To avoid overwriting group-membership data on the recovered DC, you must prevent the DC from replicating with its partners after rebooting. You can do this by simply disabling the NIC on the DC prior to the reboot. Disabling the NIC creates an additional challenge by making it difficult to perform the restore activities remotely. Please note, there is no need for this step on DCs running Windows Server 2003 SP1, as is covered later.

You can now read the Back-Link information from all restored user-objects (e.g. the `memberOf` attribute), and save this information into a file for later reference. After creating the reference file, you should be re-enable replication by re-enabling the NIC. You then need to make sure that the deleted user objects replicate to the other DCs and GCs in the remote domains. At this time, there is an inconsistency of the group-objects within AD, as all other DCs and GCs in the forest have not automatically re-populated the group memberships of the restored user objects.

The following describes the recovery procedure step-by-step:

1. Reboot DC1 to Directory Restore Mode
2. Restore AD database from backup to DC1
3. Perform authoritative restore of deleted objects via NTDSUTIL
4. Disable the NIC on DC1 (this will disable replication of the restored DC with other DCs in the AD forest)
5. Reboot DC1 to normal AD mode
6. Dump membership Back-Link information from object's `memberOf` attribute into reference files (using LDIFDE, for instance)
7. Re-activate replication on DC by enabling the NIC on DC1
8. Using the information in the reference files, re-add objects to the correct groups on DC2, thus increasing the version and USN number of the `member`-attribute and causing replication of the group
9. Repeat the above for Universal Groups from other domains on a DC of the respective domain (this will usually require Enterprise Admin rights)

The last two steps leverage the reference files and re-add the users to the respective groups on a different DC from the same domain and other DCs from the remote domains. This will avoid a potential loss of other changed group memberships.

Additional Challenges

There are several differences between the group-types with respect to availability of the Back-Link data. This section clarifies these differences.

Global Groups:

- Can only contain users of the same domain.
- Can be joined to Domain Local Groups in its own and other domains of the AD forest (as well as outside of the AD forest to local groups of trusted domains or to local groups on member systems – but in terms of Back-Link recovery, only the groups within an AD forest are of interest).
- In W2K Native Mode, Global Groups can also be nested inside other Global Groups of the same domain or within Universal Groups.
- Membership (Forward-Link) is only replicated within its own Domain NC.
 - *This is not an issue for the Back-Link recovery, as all members of a Global Group are also part of the same NC –therefore they will always be available after the restore of a DC, including the required Back-Link information.*

Universal Groups:

- Can contain users and Global or Universal Groups from any part of the AD forest (but no users or groups from external trusted domains).
- Can be a member of Domain Local Groups in its own and other domains of the AD forest (as well as outside of the AD forest to local groups of trusted domains or to local groups on member systems – but in terms of Back-Link recovery, only the groups within an AD forest are of interest).
- Can also be nested inside other Universal Groups within the same AD forest.
- Membership (Forward-Link) is replicated as part of the GC.
 - *This means, that also the Back-Links of the members of Universal Groups from any domain are available on GCs.*
 - *To correctly restore the Universal Group memberships of users in a multi-domain environment via the described Back-Link recovery process, the AD database must be recovered on a GC.*

Domain Local Groups:

- Can contain user and group objects from any domain within the AD forest, as well as objects from any trusted domain (but in terms of Back-Link recovery, only the groups within an AD forest are of interest).
- In W2K Native Mode, Domain Local Groups can also be nested inside other Domain Local Groups.
- Membership (Forward-Link) is only replicated within its own domain NC.
 - *This means that the Back-Link information used to determine the object's group-memberships is not created on user-objects of a remote domain.*
 - *This also means that deleted and authoritatively restored user or group objects of the same AD forest cannot be re-added back to Domain Local Groups of remote domains by means of analyzing their Back-Links.*

How to Prepare Yourself for Enabling a Full Restore of Object-Links

The following options can help prevent the loss of Object-Links (group memberships) as a result of the deletion of objects in AD:

- Ensure, that recent backups of the System-State of at least one DC/GC^[7] of every domain in the AD forest are available.
 - *This is not an option – this is a general best practice!*
- Try to not add users from remote domains directly to Domain Local Groups, so that in case of user-deletion, Domain Local Groups don't need to be updated.
 - *Because it is currently not possible to apply rules at the OS level as to what type of objects an operator is allowed to add to specific groups, this may not be a feasible approach.*
- Keep one DC of every domain of the AD forest in a special "Lag-Site", which only has a nightly (or longer) replication window. These DCs can potentially be used for an immediate authoritative restore of objects after they have been deleted, without requiring a tape-backup solution.
 - *Depending on the number of domains involved, this can be a rather expensive solution, but would save some time when restoring the objects. Using virtual servers for this task is a valid solution to keep the costs to a minimum, but does not circumvent the actual problem of Object-Link recovery. See the section called "Replication Lag-Sites" in this white paper for more information.*

^[7] You should always backup a GC, to allow restoring the DC along with the GC read-only partitions, which eases the recovery of Universal Group memberships

There are also several options to ensure full recovery of memberships of **Domain Local Groups** as part of a disaster recovery process:

- You can periodically dump members of Domain Local Groups from every domain in the AD forest to reference files. You can use these files to ensure complete recovery of Domain Local Group memberships in case of a disaster recovery.
 - *This is the preferred choice, as you could perform the dump as part of the normal backup on a dedicated DC.*
- In the event of a disaster (in this case, accidental deletion of many objects in AD), perform a restore of one DC for every domain in the AD forest to analyze the memberships of the remote Domain Local Groups.
 - *This can be rather difficult to do in an AD forest with many domains.*
- Document and test your disaster recovery plans!
 - *Again, this is not an option – this is a prerequisite for a successful restore!*

Changes in Windows Server 2003 AD with Respect to Recovery of Object-Links

Windows Server 2003 introduces quite a few updates to the AD. These include improvements on the general speed of replication as well as the replication mechanism itself. Link-value attributes – such as the members of a group – are now stored and replicated on a value-by-value basis when changes occur, as opposed to replicating the entire list of members of a group as is done in Windows 2000 AD. This is called **Linked Value Replication (LVR)**.

Like many other attributes in AD, the `member` attribute of a group is actually a **multi-valued attribute** – i.e. it can store more than a single value. But only a few multi-valued attributes contain references to other objects. In this case these attributes are referred to as **link-value attributes** and contain entries in the Link-Table on every DC. Refer back to the section entitled "How AD Links Objects" for more information.

The Windows Server 2003 version of AD extended the Link-Table with several columns, one of which contains the deletion date of a link. Similar to the tombstones of an object, this now allows the replication of Forward-Links between DCs to include removal of single links in the Link-Table, e.g., when a user has been removed from a group. Just as with tombstones, Object-Links in the Link-Table with a deletion date older than the tombstone lifetime will be cleaned from the database by the garbage collector service.

Nonetheless, as in Windows 2000, when an object containing Back-Links (e.g., a user) gets deleted in an AD domain, the Link-Table will be cleaned instantaneously from the links associated with the object (e.g., the user's group

memberships). The user object's tombstone will cause all other DCs in the same domain and GCs in the forest to do the same.

As Windows 2000 DCs do not know about the extension of the Link-Table, the Link-Value replication mechanism is only available in two Windows Server 2003 forest functional levels:

- Windows Server 2003 Interim (only Windows NT4 DCs and Windows Server 2003 DCs)
- Windows Server 2003 (only Windows Server 2003 DCs)

Specifically, whenever there are still some Windows 2000 DCs remaining in an upgraded Windows Server 2003 AD forest, and you have not made the switch to a higher forest functionality level, the replication mechanism for groups and other multi-value attributes containing linked objects will still be performed the Windows 2000 compatible way, without LVR.

There is one more important detail to understand when upgrading Windows 2000 AD to Windows Server 2003: links that existed in the Link-Table prior to the upgrade will continue to be stored as a binary blob – only the new links will be stored and replicated via LVR. This does not have an immediate downside because changes to any existing group will leverage LVR and should show a decrease of required network bandwidth during the replication of changes for very large groups. However, only links stored as LVR links can benefit from another feature explained below, offered by LVR for the recoverability of Object-Links during an authoritative restore of objects.

Does LVR in Windows Server 2003 AD Solve the Authoritative Restore Problem?

After testing in a "native" Windows Server 2003 AD Forest (switched to *Windows Server 2003 Forest Functional Level*), as well as discussing the problem with the Microsoft product group, we've determined that LVR improves the disaster recovery situation in some ways, but it does not prevent all the issues that occur when objects are deleted in a multi-domain AD forest:

- When recovering objects that have Back-Links to objects of the **same domain** (e.g., users that are members of a Domain Local, Global or Universal Group hosted in the same domain), the appropriate links are **revived** in the Link-Table so that the Forward-Links are automatically re-replicated and thus **fully recovered** to the other objects within the domain. AD achieves this by also increasing the version number of the related LVR Back-Links of the authoritatively restored object in the Link Table.
 - *e.g., the membership of a group is automatically re-replicated to other DCs, where it was previously "cleaned" due to the deletion of the user object.*

-
- Links to objects in other domains (e.g., user accounts in Universal Groups from other domains in Global Catalog) are intentionally **NOT** recovered on the DC/GC where the authoritative restore is performed. The mechanism for reviving the links that refer to objects in the read-only partitions on the DC/GC does not make sense, as a remote domain's DC will only perform outbound, and no inbound, replication to another DC's read-only partition (it naturally does not expect any changes to occur on these partitions of a GC).
 - *This means that links to objects in remote domains, e.g., members of Universal Groups in the GC, will **NOT** be revived during authoritative restore of an object, even though these links are known locally to the GC where the object was physically restored.*
 - The same is true for the **manager/directReports** and **managedBy/managedObjects** link-pairs. These are also recovered within the same domain, but not across domain boundaries.
 - **Domain Local Groups** of remote domains in the same AD forest experience the same issues as in Windows 2000, as their links are not stored in the Link-Tables of DCs/GCs in other domains.

Thus, the process to fully recover Object-Links in remote domains as previously described in this document is valid even for a native Windows Server 2003 AD infrastructure. Within the same domain, no group inconsistencies should exist after an authoritative restore, however, as described before, during mass-recovery of objects, you may still need to perform the authoritative restore of the objects twice to ensure full replication of all changes to all DCs (see [Q280079](#)).

In the meantime, Microsoft has continued to work on the problems surrounding link-recovery when performing a native authoritative restore in AD. The first result was made available in Q1 of 2004 as a little tool called **groupadd**, available to customers via request from MS PSS (see [840001 "How to restore deleted user accounts and their group memberships in Active Directory"](#)).

Groupadd supports the recovery steps previously described, without requiring the customers to write their own scripts to extract and replace the link-information. The tool works for Windows 2000 and Windows Server 2003, but there was more to come with SP1 for Windows Server 2003.

SP1 Improvement: Updated NTDSUTIL with Improved Authoritative Restores

Even though the groupadd program simplifies the recovery process somewhat, the changes in NTDSUTIL in Windows Server 2003 SP1 are the most significant improvements yet made to AD backup/restore.

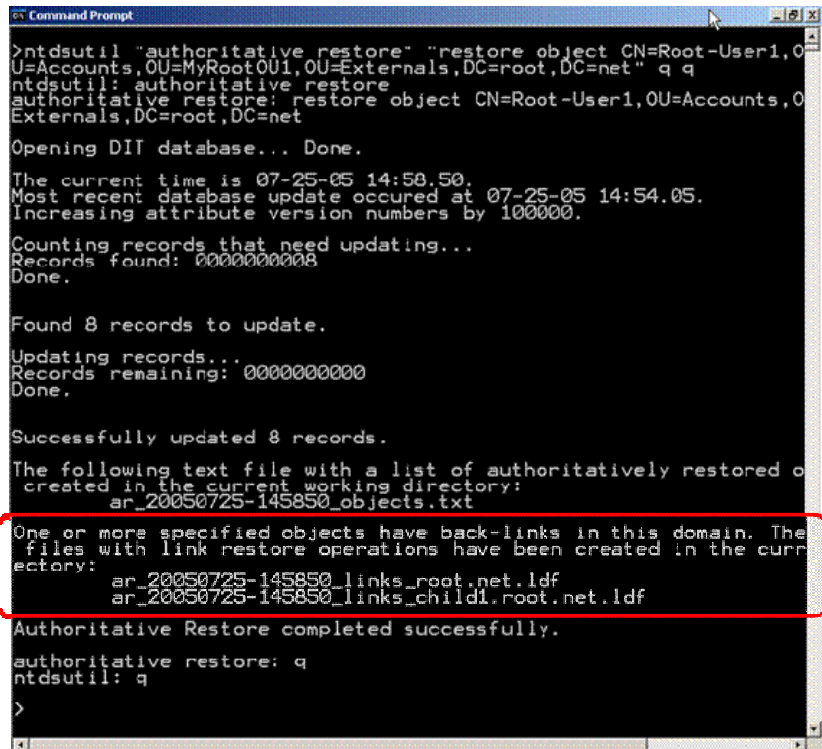
The groupadd program removes the requirement that administrators write scripts to recover user objects from backup, but it still leaves the restoration process difficult and error-prone. For example, the administrator needs to ensure that membership data in the AD database recovered on the DC used to perform the authoritative restore (recovery DC/GC) isn't accidentally overwritten at first reboot, and successful replication with another partner DC takes place. To do so, an administrator must take a DC offline and disable inbound replication on the recovery DC, usually done with the **repadmin** command (`repadmin /options <recovery dc name>+DISABLE_INBOUND_REPL`).

Afterwards, a combination of more-or-less painful LDIF and groupadd commands allow retrieval of the `memberOf` multi-value attribute of the restored user or group objects from the recovery DC. As the `memberOf` attribute on the recovery DC still stores the Back-Links of the object's membership in all groups of its own domain and in Universal Groups of other domains in the forest (if the recovery DC is a GC), the groupadd tool is able to create separate LDIF files per domain in the forest, where the deleted objects need to be re-added to groups. After creating these files and ensuring successful replication of the recovered objects in the forest, the domain-specific groupadd-LDIF files can then be imported on a GC of the respective domain to finally ensure correct recovery of the group-memberships for the deleted objects:

```
Ldifde -i -k -f Groupadd_<fully.qualified.domain.name>.ldf.
```

You can see that using the Groupadd tool is not a simple task, which is why Microsoft intended to build a similar function natively into the authoritative restore routine performed via the NTDSUTIL command. This was finally made available with Service Pack 1 for Windows Server 2003.

Similar to the Groupadd tool, the new NTDSUTIL version now creates an LDIF file during the authoritative restore of objects in AD that can be used to restore group memberships of groups in the forest. It does this by locating all Back-Links of the objects while they are being authoritatively restored, and then writes the appropriate Forward-Links twice to a domain-specific LDIF file - first as a delete and then as an add operation. Initially performing a delete ensures that the link is removed in case it still existed on a DC. Otherwise, adding the link would fail, which in-turn would not replicate the changed link out to other DCs. This feature does not depend on the forest functional level, meaning that it can be used to restore group-memberships that are either stored as legacy links or as LVR links.



```
Command Prompt
>ntdsutil "authoritative restore" "restore object CN=Root-User1,OU=Accounts,OU=MyRootOU1,OU=Externals,DC=root,DC=net" q q
ntdsutil: authoritative restore
authoritative restore: restore object CN=Root-User1,OU=Accounts,OU=Externals,DC=root,DC=net
Opening DIT database... Done.
The current time is 07-25-05 14:58:50.
Most recent database update occurred at 07-25-05 14:54:05.
Increasing attribute version numbers by 100000.
Counting records that need updating...
Records found: 0000000008
Done.

Found 8 records to update.
Updating records... 0000000000
Records remaining: 0000000000
Done.

Successfully updated 8 records.
The following text file with a list of authoritatively restored objects created in the current working directory:
ar_20050725-145850_objects.txt
One or more specified objects have back-links in this domain. The files with link restore operations have been created in the current directory:
ar_20050725-145850_links_root.net.ldf
ar_20050725-145850_links_child1.root.net.ldf
Authoritative Restore completed successfully.
authoritative restore: q
ntdsutil: q
>
```

Figure 4: SP1's Authoritative restore in action with creation of new LDIF files

Note that NTDSUTIL in SP1 doesn't just create LDIF files to recover the group-memberships. While analyzing the authoritatively restored objects, it also checks for other Back-Links, e.g., the directReports Back-Link, and creates appropriate entries into the corresponding LDIF file to re-create these Forward-Links as well – in this example, the manager attribute of users and contact objects.

Below is sample content from an LDIF restore file created by SP1's NTDSUTIL for a child-domain when restoring a user object from a root-domain of a forest (Root-User1). This was a member of a Universal Group as well as the manager for various users in the child-domain:

```
dn: CN=Child1-
UG1,OU=Groups,OU=MyChild1OU1,DC=child1,DC=root,DC=net
changetype: modify
delete: member
member: CN=Root-
User1,OU=Accounts,OU=MyRootOU1,OU=Externals,DC=root,DC=net
-
dn: CN=Child1-
UG1,OU=Groups,OU=MyChild1OU1,DC=child1,DC=root,DC=net
changetype: modify
add: member
member: CN=Root-
User1,OU=Accounts,OU=MyRootOU1,OU=Externals,DC=root,DC=net
```

```

dn: CN=Child1-
User2,OU=Accounts,OU=MyChild1OU1,DC=child1,DC=root,DC=net
changetype: modify
delete: manager
manager: CN=Root-
User1,OU=Accounts,OU=MyRootOU1,OU=Externals,DC=root,DC=net
-
dn: CN=Child1-
User2,OU=Accounts,OU=MyChild1OU1,DC=child1,DC=root,DC=net
changetype: modify
add: manager
manager: CN=Root-
User1,OU=Accounts,OU=MyRootOU1,OU=Externals,DC=root,DC=net

```

NTDSUTIL creates these files as part of the authoritative restore phase, while the DC/GC is booted into Directory Restore Mode. Therefore there is no risk of losing the valuable Back-Link data during the first replication with other DCs in the domain or forest (the potential risk would come from groups that were changed after the objects had been deleted and before they were restored – these could replicate back to the restored DC after reboot and remove the group-membership on this DC). This considerably eases the overall restore process for recovery of objects in AD.

Naturally, after rebooting the recovery DC and ensuring replication of the recovered objects throughout the forest, you will still have to import the respective LDIF files created by NTDSUTIL to a single GC that corresponds with each domain's .ldf file. The syntax to import the data is as follows:

```

Ldifde -i -k -f ar_<date>
<time>_links_<fully.qualified.domain.name>.ldf
e.g., Ldifde -i -k -f ar_20050725-145850_links_child1.root.net.ldf

```

Also note that the Groupadd tool had no means to help with restoring membership of deleted objects in Domain Local Groups of other domains in the AD forest. It solely relied on information stored on the recovery DC/GC, which does not have any information (Back-Links) about the remote **Domain Local Group** (DLG) memberships. DLGs have a scope that limits the infrastructures knowledge of these group objects to ONLY the domain and DCs of the domain in which they are created. This means that although they can hold members from other domains, referencing these groups within any other domain is not possible. In a similar fashion, local groups on a member server are only known to that member server and cannot be used or referenced by other systems in the environment.

This is similar for the new SP1 NTDSUTIL, however, it gives some additional help to support administrators with restoring missing links of the recovered objects in all the other domains of a forest. For this purpose, NTDSUTIL creates the files called:

```

ar_<date>-<time>_links_<objects>.txt
e.g., ar_20050725-145850_objects.txt

```

This file is created in the same directory as the .ldf files, but you use it rather differently. While the new SP1 NTDSUTIL version still has no way of figuring out the Domain Local Group links of recovered objects in a foreign domain, it can retrieve this information on a DC of the respective domain with the help of the aforementioned text files. This file merely lists the recovered objects (it would list all child-objects if a subtree was restored) with GUIDs and is used to create LDIF files containing the links of objects without actually performing an authoritative restore. Here is the sample content for the single user object we restored:

```
3b252a5e-2bb0-49ec-92f3-5190c21faf2f;CN=Root-  
User1,OU=Accounts,OU=MyRootOU1,OU=Externals,DC=root,DC=net
```

So what do you need to do to recover the DLG links in the other domains, e.g., our child1.root.net domain? You need to restore one DC of every domain to a point in time prior to the deletion of the objects in the proper domain. And what is the key? Naturally, you won't need to do a real authoritative restore on these DCs. Instead – while the DC is still booted into DSRM mode - provide the previously created text file to the NTDSUTIL tool running on the child1.root.net DC with the following option to directly create LDIF files. This will create files that contain links to the DLGs of the respective domain:

```
NTDSUTIL "authoritative restore" "Create ldif file(s) from %s"
```

Where: %s is to be replaced with the name of the
ar_<date>-<time>_links_<objects>.txt file, which first needs to be
copied to the DC on which NTDSUTIL is started

```
e.g., NTDSUTIL "authoritative restore" "Create ldif file(s)  
_from ar_20050725-145850_objects.txt"
```

This needs to be done for *every* domain in a multi-domain forest. After reboot of the respective DCs, the LDIF file for that domain can be imported as described above.

So the recovery of memberships with SP1 has improved, and you can even recover memberships in remote Domain Local Groups, but it remains a very cumbersome and error-prone process. Because an easy full recovery of objects in AD remains a special challenge, various third-party recovery tools, such as NetPro's RestoreADmin still provide substantial value over the capabilities of NTDSUTIL in Windows Server 2003 SP1.

Summary

We've discussed how Active Directory stores linked objects, such as users and groups, and the various problems associated with restoring linked objects from backup. We've also looked at the improvements offered by Windows Server 2003 and Windows Server 2003 SP1. The next section of this paper describes the recovery problems associated with another critical component of Active Directory, Group Policy Objects.

Tombstone Reanimation

To briefly review, when AD deletes an object, it does not immediately remove the object from the database. Instead, AD removes most of the attributes of the object, renames the object to <oldname>\0ADEL:<GUID>, and then moves the object to the hidden Deleted Objects system container of the NC. This object is what we call a **tombstone**.

During the deletion process, AD increments the version number of the deleted object so that the tombstone replicates to other DC/GCs and they will then also delete the object. Sometime later, after the tombstone's lifetime has expired, the garbage collection services on each DC finally removes the object from the AD database.

With the introduction of a feature called "**tombstone reanimation**" in Windows Server 2003, Microsoft now supports the online recovery of deleted objects. There is no user interface in the OS that allows exposes the tombstone reanimation feature. Instead Microsoft introduced LDAP functions for third party vendors to leverage with AD backup and recovery tools, such as NetPro's RestoreADmin (<http://www.netpro.com/products/restoreadmin>). The tombstone reanimation process is further described, with sample code, here:

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/ad/ad/restoring_deleted_objects.asp

As previously mentioned, the tombstone of an object does not contain all of the data within the original object – it is merely a "skeleton" of the most important attributes required to replicate the object to other DCs and inform them of the deletion of the object. Attributes that store the actual data that are used for authentication or to identify account information (e.g., any links to other objects such as group memberships) do not exist in the tombstone.

AD does, however, allow administrators to influence the removal of attributes during deletion of an object. If the schema definition of an attribute has bit 3 of their **searchFlags** property set, AD will preserved that attribute in the tombstone object. For more information, see:

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/adschema/adschema/a_searchflags.asp

The following table lists the default attributes in the Windows Server 2003 schema, which are preserved in a tombstone as a result of the **searchFlag** setting:

Attribute is used by...		CN of schema attribute
User	Group	
		CN=Attribute-ID
		CN=Attribute-Syntax
		CN=DN-Reference-Update
		CN=Governs-ID
	x	CN=Group-Type
x	x	CN=Instance-Type
		CN=LDAP-Display-Name
x	x	CN=Legacy-Exchange-DN

Attribute is used by...		CN of schema attribute
User	Group	
		CN=ms-DS-Additional-Sam-Account-Name
		CN=ms-DS-Auxiliary-Classes
		CN=ms-DS-Entry-Time-To-Die
		CN=ms-DS-IntId
		CN=MSMQ-Owner-ID
		CN=NC-Name
x	x	CN=NT-Security-Descriptor
		CN=Obj-Dist-Name
x	x	CN=Object-Class
x	x	CN=Object-Guid
x	x	CN=Object-Sid
		CN=OM-Syntax
		CN=RDN
x	x	CN=Repl-Property-Meta-Data
x	x	CN=SAM-Account-Name
		CN=Sub-Class-Of
x	x	CN=System-Flags
x		CN=uid
x		CN=User-Account-Control
x	x	CN=USN-Changed
x		CN=USN-Created

Table 1: List of attributes kept in tombstone due to searchFlag setting

Comparing this list to the total number of attributes stored with the respective object class allows for the estimation of the remaining challenge for AD backup and restoration after the tombstone reanimation has occurred. Here are samples for the user and group class:

- User Class objects
 - Have 257 attributes in default schema
 - Only 13 flagged with searchFlag Bit 3 to remain in tombstone
 - Approximately 95% of all possible attributes and contained information herein are lost
- Group Class objects
 - Has 132 attributes in default schema
 - Only 11 flagged with searchFlag Bit 3 to remain in tombstone
 - Approximately 90% of all possible attributes and contained information herein are lost

Some applications that extend the AD schema also add other attributes to the list of preserved attributes with the tombstones. Exchange 200X is the best known application to extensively extend the AD schema. It configures the following 10 extra attributes to remain with the tombstone objects (there are no extra tombstone related additions from Exchange Server 2003):

CN of schema attribute

CN=ms-Exch-Home-Server-Name
CN=ms-Exch-Imported-From
CN=ms-Exch-Mailbox-Guid
CN=ms-Exch-Mailbox-Security-Descriptor
CN=ms-Exch-Master-Account-Sid
CN=ms-Exch-Previous-Account-Sid
CN=ms-Exch-User-Account-Control
CN=Proxy-Addresses
CN=Purported-Search
CN=Version-Number

Table 2: Additional attributes kept in tombstone with Exchange 200X

Why Recover Tombstones?

If most of the information of an object is unavailable in a tombstone anyway, why would you want to use the tombstone reanimation feature at all? Why not just re-create the objects from scratch? The answer is actually quite simple.

When you recover objects online, without rebooting the DC into Directory Restore Mode, some critical object data, such as SIDs and GUIDs, are still contained in the tombstone. This data is critical for additional restores, such as assigned group permissions. For example, Access Control Lists (ACLs) use the SID of a security identifier object to store its permissions. A newly created group would always get a new SID and GUID so that permissions assigned to the equally-named old group would not apply to the new group. Similarly, a user's profile would become unusable if a new user with the same name is created, as the GUID and SID are both used to locate the user's profile. Therefore restoring objects is generally preferable to recreating the objects from scratch.

The biggest challenge faced by AD backup and recovery products when performing tombstone reanimation is the repopulation of the lost attribute information on the recovered objects. To do so, most products either leverage a backup of a DC's system state to read information from the NTDS.DIT file restored to an alternate location, or they store the data of the AD objects in a separate database (e.g., [NetPro's RestoreADmin](#)), and are then able to leverage this data to write it back to the previously reanimated tombstone objects.

The Tombstone Recovery Challenge

Some attributes, such as passwords and sIDHistory, cannot be recovered by simply reading them from a source and then writing them back to the AD database during the tombstone reanimation recovery process.

A password cannot be recovered by simply writing it back to an object in AD – this would require that the password was extractable from AD in the first place, and the recovery tool could then write it back to the password attribute. To

forego this issue, the `searchFlag` for the "unicode-PWD" attribute could be configured to preserve it within the tombstone, i.e. set bit 3 to 1 (decimal 8). AD would then recover the attribute automatically when reanimating the tombstone.

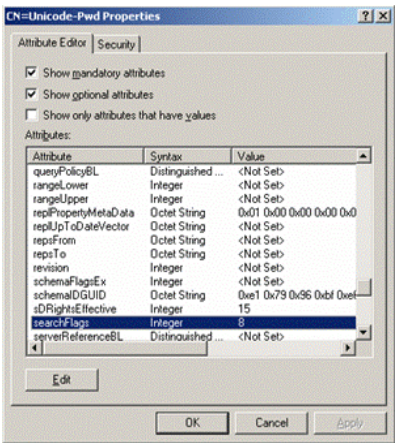


Figure 9: Adding password attribute to tombstone

The `sidHistory` attribute is challenging for several reasons. Although the attribute can be read quite easily, and thus backed-up to another location, it cannot be easily restored due to various restrictive rules that AD enforces during write operations. These are the same restrictions that an administrator faces when migrating objects from one domain to another and wants to save the old SID into the `sidHistory` attribute of the target user in AD. During the write operation, for example, the source domain would have to be connected with administrative credentials and ensured that the object with the source SID actually exists. This could be tricky in itself if the source domain were still reachable, but given the possibility that it no longer even exists, it would be impossible to write the stored SIDs back to the `sidHistory` attribute.

Therefore it certainly makes sense to preserve the `sidHistory` attribute with the tombstones. Microsoft obviously thought the same, adding the `sidHistory` attribute to the set of attributes preserved at deletion within Windows Server 2003 SP1.

However, the installation of SP1 on a DC doesn't change anything in the schema of AD, which would need to be updated if the `searchFlags` for the `sidHistory` attribute were to be used to ensure it's preservation in the tombstone objects. But, this is good news - otherwise the rollout of SP1 would likely be hindered by various schema change tests.

How does it work? In addition to looking at `searchFlags`, there is actually a second mechanism used by AD to decide which attributes to preserve with the tombstones. Microsoft maintains a hard-coded list in the AD code itself so that even something as critical as a schema change, can't cause AD to enter a non-working state.

In SP1, Microsoft added `sidHistory` to the list of always-preserved attributes. But while there is plenty of upside to not requiring a schema change, there is

**Improvement in
Windows Server 2003 SP1**

also a downside. Only Windows Server 2003 DCs with SP1 installed will actually preserve the `sIDHistory` attribute during the deletion of objects in AD. In a mixed AD forest with other Windows 2000 DCs and non-SP1 Windows Server 2003 DCs, this won't occur. Therefore you could encounter an inconsistency between deleted objects – some with and some without the `sIDHistory` attribute after deletion. Once contained in the tombstone, it could be recovered by tombstone reanimation using any Windows Server 2003 DC. Certainly, if you are planning to use the tombstone reanimation features to recover deleted objects in AD, it may still be a good idea to update the `sIDHistory searchFlag` in the schema prior to or even during the rollout of Windows Server 2003 SP1 for your AD DCs. In this case you'd want to ensure that you leave the other bits of the `searchFlag` for `sIDHistory` intact, as bit 0 is already set as an indexed attribute. This means you would need to set the `searchFlag` for the `sIDHistory` attribute to $2^0 + 2^3 = 1 + 8 = 9$ for it to be preserved as a tombstone.

SP1 Improvement: Longer Tombstone Lifetime

As described in the previous section, you can use tombstones to allow online recovery of deleted objects. But more importantly, AD uses tombstones to inform other DCs in the forest about deletions by replicating the tombstone object just like any other object to other DCs.

However, since the usual object deletion in AD should not be accidental nature requiring the recovery of those objects, there must be some means by which objects get fully removed from the AD database. This is performed by the AD garbage collection process. This process runs locally and independently on every DC in the AD forest and periodically queries the DC's database for tombstone objects that are old enough to be erased. By default, this process runs every 12 hours. Its frequency can be configured in the configuration container via the attribute **garbageCollPeriod** on the Directory Service object:

```
garbageCollPeriod ⇨  
CN=Directory Service,CN=Windows NT,CN=Services,CN=Configuration,DC=<MyRootDomain>
```

After removing the stale data from AD, the garbage collection process also invokes an online defragmentation thread which will make the additional free space efficiently available to the database. You can monitor the frequency of the garbage collection process' activity by checking EventID 700 (starting online defragmentation) in the Directory Services event log.

So what happens if a tombstone hasn't been replicated to a specific DC before all other DCs remove it via the garbage collection process, e.g., because this DC was offline for a very long time? Well, that DC has no clue that the object belonging to the tombstone should be deleted and removed from the database, and thus it continues to store it happily in its own database. While that alone could be a security problem, it's even worse when this DC goes back online and a change to the previously removed object is replicated to other DCs, which could re-replicate the object back to all other DCs and GCs in the forest.

Depending on the function of the DC and the naming context that the object "lived" in, this could now introduce a **"lingering object"** into AD.

The time that a tombstone is stored in AD before it is erased from the local database of a DC by the garbage collection process is defined by the tombstone lifetime. This value can be configured in the configuration container via the attribute **tombstoneLifetime** on the Directory Service object:

```
tombstoneLifetime ⇨  
CN=Directory Service,CN=Windows NT,CN=Services,CN=Configurati  
on,DC=<MyRootDomain>
```

Until now, the default value for the tombstone lifetime was 60 days. But Microsoft underestimated how many companies run their AD forests in a more-or-less unmanaged fashion, and have run into issues as described above, where DCs offline for more than 60 days were brought online again and started to replicate stale data back into the AD forest.

As a result, in Windows Server 2003 SP1, Microsoft has increased the default tombstone lifetime from 60 to 180 days for new AD forests that are implemented with SP1, i.e. where the first server being promoted to a DC creating the forest root domain is running Windows Server 2003 SP1. The hope is that longer tombstone lifetimes will decrease the chance that a deleted object remains in the local directory of a disconnected DC beyond the time when the garbage collection service deletes it from all other online DCs in the forest.

Because editing the tombstoneLifetime value for an existing AD forest during the SP1 implementation on DCs would require changes to the configuration container, this is not done automatically when SP1 is rolled out to existing DCs. Nonetheless, companies should evaluate whether they would benefit from increasing this value in their existing AD forests. It's a tradeoff between slightly more disk space required on the volume that hosts the NTDS.DIT file on DCs (as more deleted objects will be stored in the AD database), versus longer recoverability of objects and lower chances of replication issues caused by offline DCs. However, in a well managed and centrally monitored AD this should not be an issue.

Group Policy Objects Recovery

The complex nature of native **Group Policy Object** (GPO) backup and recovery could very well constitute a separate white paper of its own. This complexity is directly related to the fact that GPOs are made up of two parts - the **Group Policy Container** (GPC, an object stored in AD) and the **Group Policy Template** (GPT, a set of folders and files stored in SYSVOL). To compound the problem, these two components are replicated via different methods: normal AD replication and FRS.

- The GPC is an AD object that contains GPO attributes. It includes subcontainers for information about computers and users. Data stored

in GPCs includes version information, status information (whether the GPO is enabled or disabled) and a list of components of the GPO.

- The GPT is a folder hierarchy residing below the SYSVOL Folder on DCs. The GPT contains all administrative templates and scripts that are used by the corresponding GPO. The name of the GPT is the GUID of the corresponding GPO, e.g., `%systemroot%\Sysvol\Sysvol\root.net\Policies\{B2A3F4.....}`
- Through a process known as “**linking**” (don’t confuse this with the Object-Links we discussed earlier), the GPOs are associated with the locations in the AD tree that are to receive the Group Policy. This way, one GPO can be linked to multiple containers in the tree, and a single container can have multiple GPOs assigned to it (e.g., a GPO that enforces a special registry setting can be applied to different OUs without having to recreate the GPO for every OU). If multiple GPOs are assigned to the same container, the order in which they are processed can be defined by the administrator.

Given the difficulty of recovering other linked objects, it’s easy to imagine the challenge to recover an accidentally deleted GPO or one that’s been improperly edited and is now causing problems in an AD domain. Using the normal authoritative restore process, recovering a GPO requires that you restore the appropriate GPC object in AD, along with the correct GPT folder in SYSVOL. The latter tends to cause the most pain, as native tools do not allow an easy and straightforward authoritative recovery of a single folder in the SYSVOL area. In its entirety, this process is both time consuming and error prone. And because broken GPOs can have a serious impact on users and machines in AD, it is a best practice to handle backup and recovery outside the normal System-State backup and recovery routines.

To facilitate a more user-friendly environment for GPO management and recovery, the Microsoft Group Policy Management Console (GPMC) is available for Windows Server 2003. The GPMC is only available on Windows XP or Windows Server 2003 machines, and can also be used to manage a Windows 2000 AD. As exhibited in Figure 5, GPMC includes a feature for GPO backup, providing storage via a combination of XML files and a copy of the content of the GPT directory from SYSVOL, in a directory of your choice. Naturally, any data that is referenced from within a GPO, e.g., a logon script stored in a different folder outside of the GPT, still needs to be backed up separately.

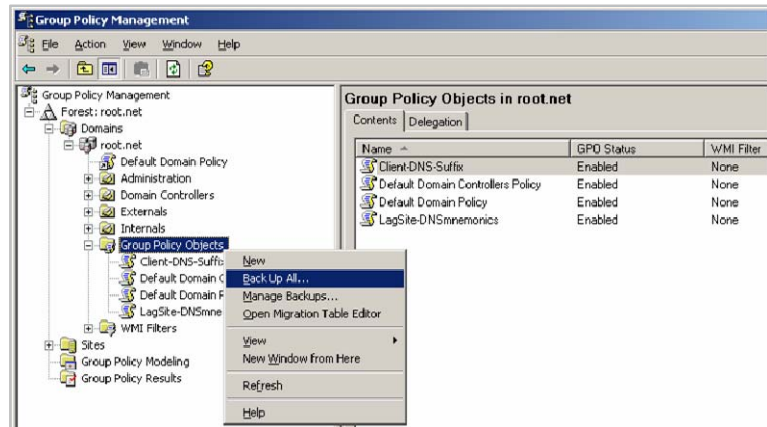


Figure 5: GPO Backup Option in GPMC

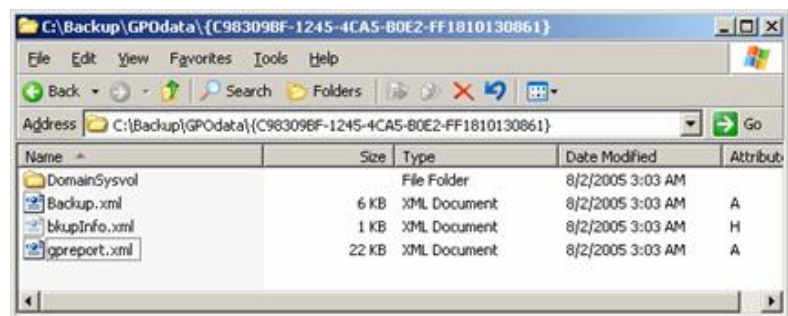


Figure 6: GPO Backup files as created by GPMC

Note, however, that backing up a GPO with GPMC does not resolve all the GPO recovery problems you may face. For example, although the links to the OUs are stored in the `gpreport.xml` file of the backup, the recovery of a policy does NOT restore the Site/Domain/OU links of the GPOs. The GPMC can, nonetheless, create detailed html reports containing all settings of a GPO, including the OUs that a GPO is linked to, in much more legible format. If you use meaningful GPO names, it will make interpreting the reports that much easier.

Links hide			
Location	Enforced	Link Status	Path
MyRootOU1	No	Enabled	root.net/Externals/MyRootOU1
This list only includes links in the domain of the GPO.			

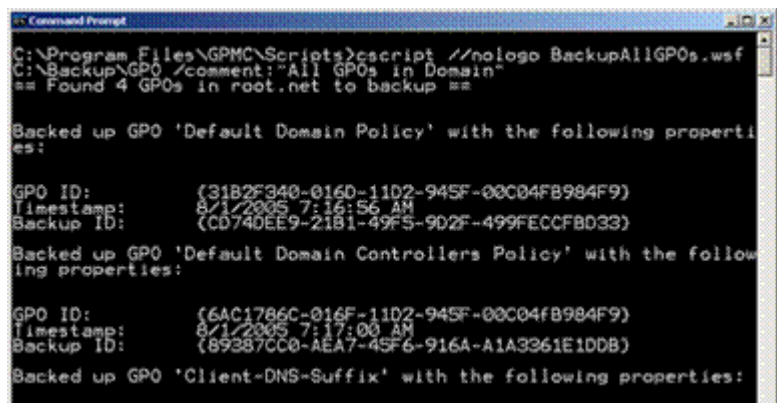
Figure 7: Links of a GPO shown in the HTML report created by GPMC

Automated GPO Backup

You can automate both the backup and reporting of GPOs using scripts included within GPMC's install folder, in a subdirectory called "Scripts". The "Scripts" folder contains various sample scripts to help automate various GPO related processes. Best suited for the backup and recovery tasks are the **BackupAllGPOs.wsf** and **GetReportsForAllGPOs.wsf** scripts for each domain. You can run these with different credentials and command-line parameters via the task-scheduler.

Example:

```
cscript.exe BackupAllGPOs.wsf c:\Backup\GPOdata /domain:root.net  
/comment:"Root GPOs"  
cscript.exe GetReportsForAllGPOs.wsf C:\Backup\GPOreports
```



```
C:\Program Files\GPMC\Scripts>cscript //nologo BackupAllGPOs.wsf  
C:\Backup\GPO /comment:"All GPOs in Domain"  
== Found 4 GPOs in root.net to backup ==  
  
Backed up GPO 'Default Domain Policy' with the following properties:  
GPO ID: {31B2F340-016D-11D2-945F-00C04FB984F9}  
Timestamp: 8/1/2005 7:16:56 AM  
Backup ID: {CD740EE9-21B1-49F5-9D2F-499FECFBD33}  
  
Backed up GPO 'Default Domain Controllers Policy' with the following properties:  
GPO ID: {6AC1786C-016F-11D2-945F-00C04FB984F9}  
Timestamp: 8/1/2005 7:17:00 AM  
Backup ID: {89387CC0-AEA7-49F6-916A-A1A3361E1DDB}  
  
Backed up GPO 'Client-DNS-Suffix' with the following properties:
```

Figure 8: GPO Backup from command line

Using this approach, recovering a broken GPO does not require restoring an AD DC. Instead, you can use the Restore feature of the GPMC UI to directly restore the GPO from the backup files.

GPMC will be included as part of the OS with the Windows Server 2003 R2 release – in the meantime, you can find it at the following link:

<http://www.microsoft.com/downloads/details.aspx?FamilyId=0A6D4C24-8CBD-4B35-9272-DD3CBFC81887&displaylang=en>.

Summary

We've now covered pretty much everything you need to know about restoring deleted objects in Active Directory, either using the authoritative restore function of NTDSUTIL, or by using tombstone reanimation. If you have a situation where you simply recover some deleted data in Active Directory, this section should be enough for you to be successful.

However, if you need to restore an entire domain or forest, the process is considerably more involved. The next section describes how to recover Active Directory as a service.

ACTIVE DIRECTORY SERVICE RECOVERY CHALLENGES

In the previous section we discussed recovering Active Directory data. Restoring data in the directory this way is an appropriate disaster recovery solution for situations where relatively few objects have been lost, say a few users or an OU, and where Active Directory is still functioning properly. In the situation where Active Directory is damaged to the point that it no longer functions properly, you need to perform an Active Directory service recovery.

Losing the availability of general AD services, such as authentication and authorization of users and LDAP directory servers, is actually less likely an issue than accidental deletion of data like user objects in AD. This is due to the fault-tolerant nature of AD, as DCs are basically created equal^[8] and if one fails, clients can typically leverage another DC for authentication and other DC-related actions. If another DC is located in the same site as the client, the outage of a DC goes mostly unnoticed as a client will contact the next closest DC. If a DC across the WAN has to be used, users will likely notice some delay during logon and for other directory queries, but the users can continue to work.

Naturally, the resilience of AD to a failure of a DC does rely on some basic requirements which you have to take care of during the design and deployment of their AD infrastructure:

- Every domain should contain at least two DCs
 - If possible, one DC of each domain should be located in a different geographic location.
 - Clients should have IP-connectivity to at least two DCs – either in the same or in different Active Directory sites.
 - In multi-domain forests, beware of the GC requirements for authentication, i.e. a user does not only require access to a DC at logon, but also to a GC so that the user's universal group memberships in any domain of the forest can be evaluated (could optionally leverage group-caching on Windows Server 2003 DCs to mitigate this requirement).
- DNS needs to be configured in a fault tolerant way as well
 - This is automatically the case if all DCs are also DNS servers and host the required DNS zones (this can be achieved equally well with DNS solutions from other vendors).
 - The required DNS zones, which need to be accessible by clients are:
 - DNS domain of own AD domain (e.g., child1.root.net)

^[8] Of course there are exceptions – there is only one instance of each Flexible Single Master Operations (FSMO) role holder in a forest and domain. Examples of FSMO roles are the Schema master and the PDC emulator. However, none of the FSMO roles are so critical that an outage of a DC that is a FSMO role holder will typically cause any availability issues.

- Service Records sub-zone of own AD domain (e.g., `_msdcs.child1.root.net`)
- Service Records sub-zone of root AD domain (e.g., `_msdcs.root.net`)
- The DNS resolvers for clients and servers have to be configured to allow the client to reach a DNS server in a different site, if the DNS servers in its main site fail.

There are other dependencies of course, such as that the underlying network operates as reliably and as resiliently as possible against outages. But the dependencies on DNS are clearly the ones causing the most issues in DC-outage scenarios.

However, even if AD is fault-tolerant against failures such as the loss of a DC, a broken DC still needs to be fixed. The following chapter focuses on the different options administrators have for restoring DCs, a complete domain or an AD forest.

Domain Controller Recovery

Due to the failure tolerance of AD with multiple DCs, you do not need to back up every DC in a forest. This certainly reduces the overall effort with respect to the backup of AD, especially if you have deployed many DCs in different sites. It further reduces the risk of handling the backup tapes of DC backups in remote sites, which need to be secured physically just as much as the DC itself. Most companies choose to back up only the DCs in their main locations (hub sites), which is sufficient for an AD domain or forest recovery. For safety's sake, you should always ensure that at least two DCs in every domain in a forest are backed up regularly.

And not all DC recovery scenarios require access to backup tapes. If a DC has a hardware problem, such as the failure of a network card, CPU or memory, you can usually replace the broken hardware and boot the DC up again. However, if a whole hard-disk subsystem fails and all the disks are lost on the server, the situation is slightly different as the new disks will not be of much help unless the data is recovered from backup.

In general, you have two basic options to recover a DC:

1. Repair the server hardware and perform a normal restore of the complete operating system along with the data, which will also restore the DC's Active Directory functionality
2. Repair the server hardware or completely replace the server and install the operating system from scratch, then re-promote the server to a DC

Both options have their dependencies and caveats and you need to choose the appropriate option for your particular failure scenario. The following points should help to find the right approach in case the server hardware fails.

-
- **Option 1: Repair server hardware and restore DC from backup:**
 - Repairing the hardware can be time consuming
 - Requires a valid system state backup of the DC
 - Requires identical hardware – although some useful tips on recovering a DC on different hardware can be found in the Internet, Microsoft officially only supports the restore of a DC on the same hardware
 - The restore from backup approach should not be used for a DC holding the RID FSMO role; to protect from duplicate RIDs/SIDs in a domain, it is a best practice never to restore a RID master from backup – as such it should also not be backed up to tape (this will prevent accidental restore from tape)
 - If DC is a FSMO role holder (other than RID master), need to ensure that FSMO role was not previously seized to another DC
 - Recommendation: This approach can be used for any DC (except the RID master) for which a valid backup exists and the server hardware is easily replaced with identical hardware
 - **Option 2: Rebuild from scratch and re-promote DC:**
 - Allows to use different DC hardware – could leverage a "cold standby" server
 - Requires metadata cleanup of DC records in AD prior to re-promoting the DC, so that other DCs no longer attempt to replicate from it via it's old GUID
 - Requires DC to replicate all AD data – can be a time consuming effort
 - New install from media feature in Windows Server 2003 helps to decrease the time required to promote a DC – however, the backup-data to restore from still needs to be copied to the DC prior to promotion
 - Recommendation: This approach can be used for any DC and is required for those where no backup is available. Optimizing the time it takes for DC re-promotion via the promote-from media feature can save a lot of time.

In case of the unlikely event of the corruption of the AD database on a DC, the second option is the correct choice to restore the DC. However, this would not necessarily require rebuilding the DC hardware, as it may still be fully intact. Instead the DC can be demoted forcefully. This is done via the `DCPROMO /forcere removal` command, available for Windows Server 2003 DCs and Windows 2000 DCs with SP4. The preferred method of performing the cleanup in the AD database is using the **NTDSUTIL.EXE** command-line tool with the **metadata cleanup** option, which has been further enhanced in SP1 of Windows Server 2003.

Improvement in Windows Server 2003 SP1

There are actually two changes in Windows Server 2003 SP1 that are not directly related to the process of data being deleted and recovered in AD or within a DC, however, you should be aware of the second recovery option described above, where a DC is "built-from-scratch." This involves the metadata cleanup feature in NTDSUTIL and the install from media option for DCPROMO.

SP1 Improvement: Updated NTDSUTIL for easier server metadata removal

Windows Server 2003 SP1 added some features to NTDSUTIL that make the process for removing stale DC data from the AD database less error prone.

It is still quite a chore to perform all the steps required to setup the metadata cleanup process, i.e. connecting to a server, and selecting the appropriate domain and site. This has not been simplified in SP1 of Windows Server 2003. But once you've done that, you can perform the metadata cleanup operation with a single command:

```
ntdsutil "metadata cleanup" "remove selected server" _  
"<DN of ServerName>"
```

Or

```
ntdsutil "metadata cleanup" "remove selected server" _  
"<DN of ServerName>" on <DNS of ServerName2>
```

Example:

```
ntdsutil "metadata cleanup" "remove selected server" _  
"CN=RootDC3,OU=Domain Controllers,DC=root,DC=net" _  
rootdc2.root.net q q
```

Note: When adding all commands to a single line as shown in this sample (remove the "_"), you must put quotes around the commands that have spaces in them. In addition, you can add quit-commands (q) to the end of the line to exit the NTDSUTIL tool when it has finished processing its command. The number of quit-commands is equal to the command depth of NTDSUTIL.

The main benefit of SP1 is that metadata cleanup now also removes File Replication Service (FRS) connections and attempts to transfer or seize any FSMO roles that the retired DC holds. These additional processes are performed automatically.

Sample Output:

```
Transferring / Seizing FSMO roles off the selected server.  
Removing FRS metadata for the selected server.  
Searching for FRS members under "CN=ROOTDC2,OU=Domain Controlle  
rs,DC=root,DC=net".  
Removing FRS member "CN=ROOTDC2,CN=Domain System Volume (SYSVOL  
share),CN=File Replication Service,CN=System,DC=root,DC=net".  
Deleting subtree under "CN=ROOTDC2,CN=Domain System Volume  
(SYSVOL share),CN=File Replication Service,CN=System,DC=root,DC  
=net".  
Deleting subtree under "CN=ROOTDC2,OU=Domain Controllers,DC=roo  
t,DC=net".  
The attempt to remove the FRS settings on CN=ROOTDC2,CN=Servers  
,CN=Core-LagSite,  
CN=Sites,CN=Configuration,DC=root,DC=net failed because  
"Element not found."  
metadata cleanup is continuing.
```

```
"CN=ROOTDC2,CN=Servers,CN=Core-LagSite,CN=Sites,CN=Configuration,DC=root,DC=net" removed from server "rootdcl.root.net"
metadata cleanup: q
ntdsutil: q
Disconnecting from rootdcl.root.net...
```

Improvement in Windows Server 2003 SP1

SP1 Improvement: Install From Media DCPROMO option retains DNS application partitions

Although the **Install from Media** (IFM) option is targeted more towards classical rollout scenarios, you can use this feature to speed up the process of fixing a broken DC, removing it from AD and re-promoting it again to a DC, potentially on different hardware.

Issues in the past with IFM were often related to the fact that although it could copy all domain naming contexts to the new DC/GC, it couldn't copy application partitions at the time of the promotion. This was especially problematic if the new DC was also a DNS server, as in Windows Server 2003. The resource records in AD-integrated DNS zones are stored by default in the DomainDNSZones and ForestDNSZones application directory partitions.

The SP1 version of IFM now offers a new option to leverage application directory partitions from the backup media that is used to promote the new DC. You can now create a new DC that is also a DNS server and make the DNS server operational right after the promotion, without the requirement for separate replication of the DNS application partitions.

This is particularly useful in recovery scenarios where you can quickly replace broken hardware with new hardware. After performing the metadata cleanup as explained in the previous section, the re-promotion of the DC on the new hardware using the IFM option to add the DNS application partitions allows the new DC to be up and running again quickly.

There are several restrictions to using the new application partition IFM feature:

- The Source DC used to create the backup for IFM and the target DC must be running Windows Server 2003 SP1. *(Similarly, a non-SP1 Windows Server 2003 can only be promoted with IFM from a backup of another non-SP1 Windows Server 2003 DC).*
- The forest functional level has been raised to Windows Server 2003.
- The DC that is used to create the system state backup contains the application directory partitions that are to be included during the DC promotion.
- An answer file is used that contains the distinguished names (or "*" for all names) of the application directory partitions that are to be included.

A sample DCPROMO answer file follows:

```
[DCINSTALL]
UserName=adm.child1
Password=
UserDomain=CHILD1.ROOT.NET
DatabasePath=D:\NTDS
LogPath=E:\NTDS
SYSVOLPath=E:\SYSVOL
SafeModeAdminPassword=
CriticalReplicationOnly=No
SiteName=Core-Site1
ReplicaOrNewDomain=Replica
ReplicaDomainDNSName=CHILD1DC2.CHILD1.ROOT.NET
ReplicationSourceDC=CHILD1DC1.CHILD1.ROOT.NET
ReplicateFromMedia=Yes
ReplicationSourcePath=C:\Temp\AlternateRestorePath
RebootOnSuccess=Yes
ApplicationPartitionsToReplicate=*
```

Start DCPROMO to leverage IFM and install APP partitions (as defined in the answer file):

```
dcpromo /adv /answer:"Path_of_the_Answer_File_Name"
```

For more information, especially about the various options in the answer files, see:

- Q311078 - How to use the IFM feature to promote Win2003-based DCs
- MSTECHNET – Create an answer file for domain controller installation

Domain Recovery

Even though recovering a single domain is a smaller project than recovering the entire forest, it is not particularly easier. If a domain partition has truly been corrupted to the extent that you need to fully recover it, the correct action is to recover only a single DC of that domain and to re-promote all the other DCs of the domain. This is by no means a trivial task and still does not solve the problem of stale data residing in the read-only copies of the respective domain partition stored on GCs in other domains.

It is true that AD does provide the means to authoritatively restore a complete AD database using the NTDSUTIL command in the **Directory Restore Mode** (DRM) of a DC, such as after a system state restore of the DC from a backup tape:

```
ntdsutil "authoritative restore" "restore database"
```

However, this command marks both the domain naming context and the configuration naming context held by the respective DC as authoritative. The command won't affect the schema partition as it cannot be authoritatively restored. But because the configuration naming context is replicated between all DCs in a forest, an authoritative restore of the complete database on a DC

with the goal to recover just the domain partition will have unexpected results in the forest. Therefore it is better to perform a subtree restore, where administrators specify the domain name of the parent container in which all objects are to be restored authoritatively – this could be the complete naming context of the domain to be restored, such as shown in this example:

```
ntdsutil "authoritative restore" "restore subtree  
DC=child1,DC=root,DC=net"
```

This command will authoritatively restore all objects in the domain child1.root.net, including all objects in the domain's system container, such as FRS, DFS and Group Policy Container objects. After reboot, the objects will be overwritten on all other DCs of the domain. Depending on the type of data corruption that occurred in the domain, this could work sufficiently to recover the domain to a known state.

Note that an authoritative restore does NOT overwrite any attributes or objects that did not exist in the domain at the time of the backup of that DC. So if the actual problem in a domain was a result of falsely executed scripts that added attributes to all kinds of objects in the domain (e.g., an email address to non-mail enabled users) or a denial of service attack that added thousands or millions of objects to the domain, an authoritative restore of the domain would NOT remove this extra data. Instead, the data would be merged back onto the recovered DC by means of replication from another DC of the domain that was still running. To fully restore a domain to a previous state, you may have to perform a forest-level restore, which is outlined in the section following the description of yet another AD backup and recovery related change in Windows Server 2003 SP1.

In summary, you can recover a domain, however it is always better to minimize the authoritative restore to the subtree of those objects that are actually causing problems in a domain. The time required to locate the faulty objects will be well spent in comparison to the time required to troubleshoot issues resulting from authoritatively restoring a very large scope of objects (e.g., complete database of the DC). New objects will not be deleted by performing an authoritative restore.

To best prepare for a recovery of a domain or any subtrees within the domain, you should ensure the backup of at least two DCs per domain in the AD forest. The following change in Windows Server 2003 SP1 will help you determine that you have a valid backup of each domain in their forest.

**Improvement in
Windows Server 2003 SP1**

SP1 Improvement: Report if a directory partition has not been backed up recently

Windows Server 2003 SP1 introduces a new event log message with the event ID **2089**, which should be added to the list of monitored events on any tools used to monitor an AD forest. Event ID 2089 warns about the backup status of each directory partition that a DC stores, including application directory partitions and **Active Directory Application Mode (ADAM)** partitions.

If a partition has not been backed up for more than the value set for the backup latency interval, Active Directory logs this event in the Directory Service event log and will continue to log it daily until the partition is backed up.

By default, the backup latency interval is set to half the forest's tombstone lifetime (i.e. 30 days for existing forests and 90 days for new forest with SP1), but you can configured the setting per DC because it is stored in the registry as a REG_DWORD value in the **Backup Latency Threshold (days)** entry in `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS\Parameters`. This value does not exist by default, so you would have to create it to set it to a non-default value.

Note that this does not mean that the DCs of a domain shouldn't be backed up before half the tombstone lifetime has expired. This is merely the longest acceptable timeframe during which a backup should at least occur once. Otherwise the lifetime for the last valid backup data would be getting uncomfortably close to the tombstone lifetime, after which it can no longer be used to recover AD. This also includes backups on remote branch office DCs which are often only written to a local disk on the DC in order to re-promote the DC from media in the case of a hardware or AD failure.

Forest Recovery

AD forest recovery is something that every company using AD as a core security infrastructure element needs to be prepared for. However, the likelihood of a full AD forest recovery is very small. Nevertheless, if you ensure a central backup of at least one DC of every domain and run one virtual DC of each domain inside a Lag-Site, you will be well prepared to cope with such a disaster. This chapter will outline the requirements for AD Forest recovery, however, it will not attempt to replace the official white paper by Microsoft on this critical task:

<http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=3EDA5A79-C99B-4DF9-823C-933FEBA08CFE>

The steps in the paper describe how to restore the entire AD forest to a point in time before a critical malfunction and ensure that each of the recovered DCs does not replicate from a DC with potentially dangerous data.

In the white paper, Microsoft assumes that an AD administrator has worked with Microsoft Product Support (PSS) to determine the cause of the failure, evaluated any possible remedies, and reached the conclusion that restoring the whole forest to a point in time before the failure occurred is the best way to recover from the failure. In many cases, total AD forest restoration should be the last option. The paper does not suggest a cause of such a failure or recommend any procedures to prevent the failure.

The roadmap of an AD Forest recovery is as follows:

Step	Notes
1. Determine Forest Structure and available backups	All domains should be well known to the person responsible for a company's AD. The backups of at least the Lag-Site DCs (if available) and one other DC in every domain must be closely monitored.
2. Identify single DC for each domain with valid backup	A "valid" backup can only be determined by performing periodic test-restores – these should be scheduled monthly for special DCs (e.g., the Lag-Site DCs).
3. Shutdown all DCs in the forest	<p>Even though this may be unrealistic, it is the suggested approach. In a distributed environment such as a global AD infrastructure, this step is one of the hardest to achieve and must be well thought through.</p> <p>The goal of this step is to ensure that no DC will be online and capable of replicating the "bad" changes back to a restored root-domain DC, as it will always be successful in overwriting the schema container (which can't be restored authoritatively).</p> <p>If all DCs are centrally managed and contain an out-of-band management board (e.g., HP Proliant ILO board), it would be possible to manage the Restore-Passwords of all DCs, disable the production NIC instead of completely shutting down the machines.</p>
4. First recover Forest Root Domain <ul style="list-style-type: none">will ensure recovery of trust hierarchy and critical DNS resource records	Can be done with a production DC or one within a Lag-Site. Metadata cleanup for all other DCs required.
5. Recover one DC of each child domain <ul style="list-style-type: none">ensure recovery of parent domains prior to their child-domains to maintain trust hierarchy	These could again be Lag-Site DCs to speed up the overall forest recovery process, as it also allows to quickly build complete Global Catalog Servers (as all are in one site). Metadata cleanup for all other DCs required.
6. Cleanup and Re-Promote all other DCs in the forest	Most time-consuming task - if the out-of-band management boards are available for all DCs in the forest, this task can be fully performed remotely. Otherwise support of the local admins might be required. Consider leveraging IFM (Install from Media) options along with unattended DCPROMO routines.

REPLICATION LAG-SITES

Recovering deleted AD objects using normal tape backups can be a lengthy process that involves more than one support group, particularly in medium to large companies. Coordinating efforts and finding backup tapes can lead to lengthy restore times. In the event that a user account or other directory object is deleted or false data has been injected into the objects, recovering the object quickly is critical to keeping a company's business running smoothly. A **Lag Replication Recovery Site** (Lag-Site) can help you quickly recover these objects.

The basic concept is simple. You create a set of DCs in special Active Directory sites that only replicate at delayed intervals with the rest of the forest.

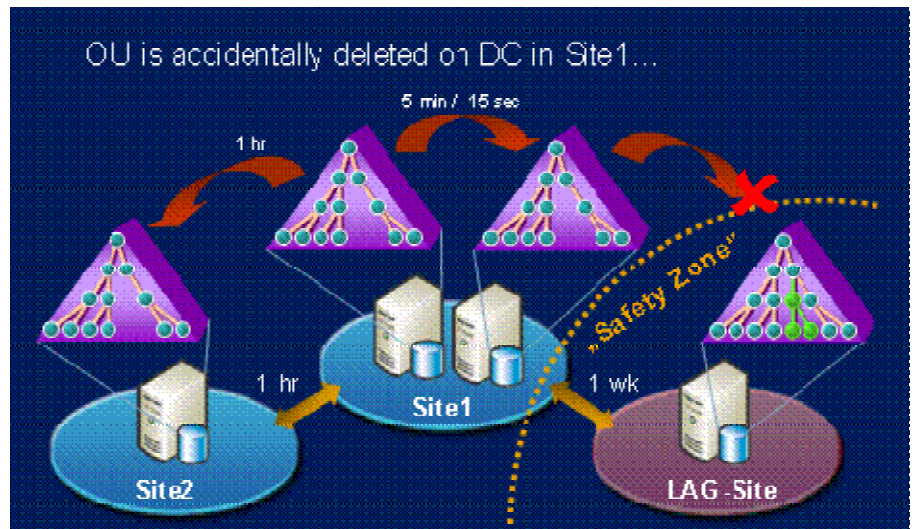


Figure 10: AD Lag-Site not replicating deleted information

How could two Lag-Sites each with a DC of every domain replicating at a staggered schedule be beneficial? For example, one lag DC replicates on Tuesday at 6:00pm and the other on Fridays at 11:00pm. The idea is that this staggered replication schedule ensures there is always a minimum of 3 ½ days to be able to recover an item after it is deleted. If only one lag DC per domain exists, there is a chance the timing of a delete could be right before a replication causing a loss of opportunity to recover the object because the lag DC would have replicated in the delete.

Lets say for example that an object gets deleted on Tuesday at 5:00pm but is not noticed until Wednesday morning. If there is only one lag DC replicating at 6:00pm on Tuesday, it would have replicated with the rest of the DCs in the domain. You would have lost the opportunity to recover the object from the lag site DC. With a second DC replicating on a staggered schedule you could still recover the object from that second DC.

Although a Lag-Site reduces the number of personnel necessary to recover an object, there are costs associated with implementing a Lag-Site and having extra hardware sitting around waiting to be used for the rare authoritative recovery.

To help mitigate this cost, you should consider using Virtual Servers. Assuming enough memory and processing power, several DCs can reside on one physical server. See the following section on "Better protection for false restores of DCs as Virtual Servers with SP1" for more information on operating DCs as virtual servers.

Sample Lag-Site Setup

This section describes the details of a sample Lag-Site configuration including special parameters to be set in the registry of the Lag-Site DCs to prevent registration of unwanted DNS records. These missing DNS records will ensure that the Lag-Site DCs are not used for normal authentication by users or for GC queries by Exchange Servers or other applications that perform LDAP lookup against AD DCs.

The general requirements for a Lag-Site are:

- Lag-Site itself should be configured in AD with a separate sub-net
- Hardware or Virtual Machine for one DC of every domain in the forest
- Need sufficient disk-space to run as Global Catalogs

Replication Schedule

The key goal of the Lag-Site will be to ensure that DCs in this site only replicate during specific intervals with other DCs of every domain in the respective production sites. The configured schedule will directly influence the data quality and the window of opportunity to recover accidentally deleted objects from the Lag-Site DCs.

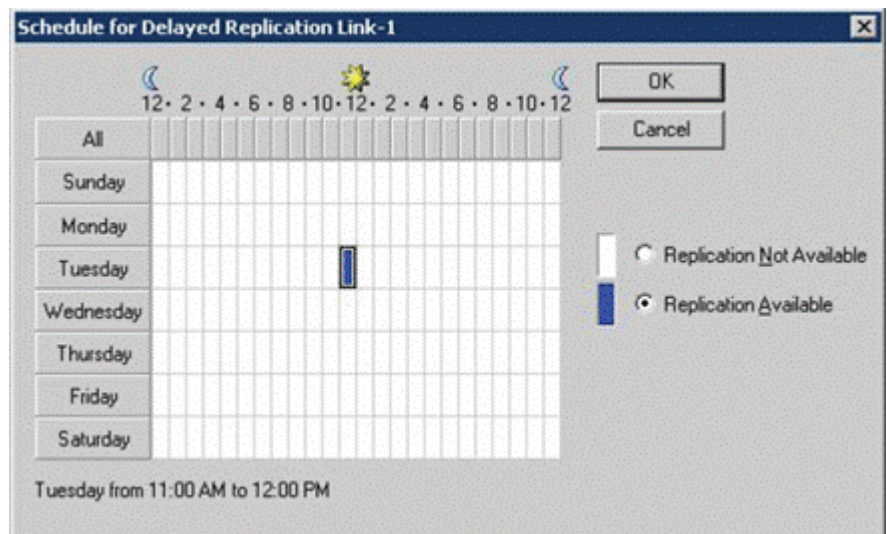


Figure 11: Replication Schedule for Lag-Site

If the Lag-Site DCs are configured to replicate more often (e.g., once a day at midnight), then the data on the Lag-Site DCs will be very up-to-date, but the maximum window of opportunity is 24 hours, assuming an object was deleted just after the last replication to the Lag-Site DCs has happened. However, if the deletion of the object happened at 11pm, then the window of opportunity to recover the object via the Lag-Site is only 1 hour. If replication into the Lag-Site only happens weekends, e.g. Saturday at midnight, then under most circumstances there is a longer window of opportunity, but the data will in worst case be seven days old.

If you plan to use only a single Lag-Site, set the replication to occur once a week on weekends, which under most circumstances will ensure quick recovery of any accidentally deleted objects. In case newer data is deleted, it can still be recovered from the daily tape-backups of another DC – however, this will take considerably longer to perform.

The Lag-Site DCs

The Lag-Site DCs are very similar to normal DCs, however they are hosted in the specially configured Lag-Site so that these DCs only replicate with other DCs at special intervals. Because these DCs are not always up-to-date with the latest changes that occurred on objects in a domain (e.g. a user's password was reset), it is important to ensure that users and applications do not use the Lag-Site DCs as authentication or LDAP servers.

Because replication to the Lag-Site will lag behind by several days, you should consider the data on the Lag-Site DCs as stale. To prevent user authentication and directory lookups, Windows Server 2003 allows you to apply a special Group Policy setting to the AD Lag-Site that hosts the Lag-Site DCs. This Group Policy setting essentially hides the DC from the rest of the environment and allows for replication only with partner DCs. The site-based **DC Locator DNS Records** not registered by the DCs GPO setting prevents the Lag-Site DCs from registering SRV and other DNS records. You can find this setting in the **Group Policy Editor** (GPE) under the `\administrative templates\system\netlogon\ DC Locator DNS Records`. For Windows 2000 DCs, you need to manually enter the settings in the registry of each Lag-Site DC, while Windows Server 2003 offers new GPO settings to configure the DNS locator records^[9].

[9] For more information also see Microsoft article ["How to Optimize the Location of a Domain Controller or Global Catalog That Resides Outside of a Client's Site"](#). You can also read Gil's article on [Active Directory Replication Topology](#).

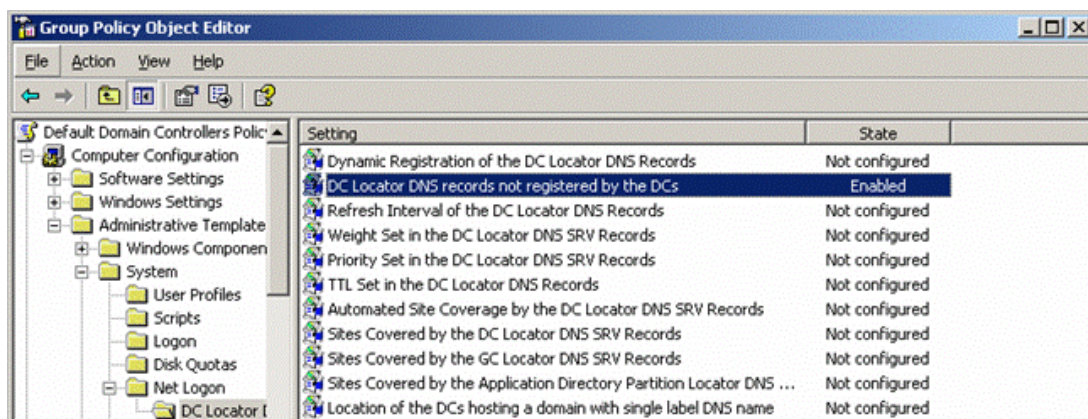


Figure 12: Configuring DNS Locator records via GPO for Win2003 DCs

The goal is to permit the registration of only the **GUID Cname** record in DNS, along with the A record for the DC **nodename** (because the GUID Cname points to this A record). It's important that **NetLogon** is not allowed to register any other DNS records, including the domain A record, any SRV records, and the **Global Catalog** (GC) A record. Each record classification is represented by a mnemonic to help make policy application easier. For each record type that should not register in DNS, that specific mnemonic must be entered into a space delimited list in the group policy. **DsaCname** is the only mnemonic that should be missing from the list of space delimited mnemonics that need to be entered into the policy.

Here is a space delimited list of DNS Mnemonics to prevent from registration as it would be entered for Windows 2000 DCs in the registry or the respective GPO for Windows Server 2003 DCs:

```
LdapIpAddress Ldap LdapAtSite Pdc Gc GcAtSite DcByGuid
GcIpAddress Kdc KdcAtSite Dc DcAtSite Rfc1510Kdc
Rfc1510KdcAtSite GenericGc GenericGcAtSite Rfc1510UdpKdc
Rfc1510Kpwd Rfc1510UdpKpwd
```

The complete list of mnemonics is shown in the table below:

Mnemonic	DNS Record Type	Description
LdapIpAddress	A	<DnsDomainName>
Ldap	SRV	_ldap._tcp.<DnsDomainName>
LdapAtSite	SRV	_ldap._tcp.<SiteName>._sites.<DnsDomainName>
Pdc	SRV	_ldap._tcp.pdc._msdcs.<DnsDomainName>
Gc	SRV	_ldap._tcp.gc._msdcs.<DnsForestName>
GcAtSite	SRV	_ldap._tcp.<SiteName>._sites.gc._msdcs.<DnsForestName>
DcByGuid	SRV	_ldap._tcp.<DomainGuid>.domains._msdcs.<DnsForestName>
GcIpAddress	A	_gc._msdcs.<DnsForestName>
DsaCName	CNAME	<DsaGuid>._msdcs.<DnsForestName>
Kdc	SRV	_kerberos._tcp.dc._msdcs.<DnsDomainName>
KdcAtSite	SRV	_kerberos._tcp.dc._msdcs.<SiteName>._sites.<DnsDomName>
Dc	SRV	_ldap._tcp.dc._msdcs.<DnsDomainName>
DcAtSite	SRV	_ldap._tcp.<SiteName>._sites.dc._msdcs.<DnsDomainName>
Rfc1510Kdc	SRV	_kerberos._tcp.<DnsDomainName>
Rfc1510KdcAtSite	SRV	_kerberos._tcp.<SiteName>._sites.<DnsDomainName>
GenericGc	SRV	_gc._tcp.<DnsForestName>
GenericGcAtSite	SRV	_gc._tcp.<SiteName>._sites.<DnsForestName>
Rfc1510UdpKdc	SRV	_kerberos._udp.<DnsDomainName>
Rfc1510Kpwd	SRV	_kpasswd._tcp.<DnsDomainName>
Rfc1510UdpKpwd	SRV	_kpasswd._udp.<DnsDomainName>

It is also important to prevent the Lag-Site DCs from registering in **WINS** where down-level clients might be attempting to resolve the 1C record to find a suitable DC in the domain. Each DC in the domain registers a 1C record in WINS. This record maps a domain name to an IP address, allowing client systems to find an appropriate DC based on the domain name. To prevent the registration of the 1C record, do not configure WINS resolvers in the IP configuration of the Lag-Site DCs.

Once up and running, you can use the Lag-Site DCs for normal authoritative restore operations when recovering data in AD. The key difference is that the authoritative restore commands with NTDSUTIL can commence right after booting the Lag-Site-DC into DSRM mode – you do not have to first recover the **systemstate** from tape, which will save a considerable amount of time. Other recovery steps are identical as previously described in this document.

Also important to note: because the Lag-Site-DCs do not publish locator records in DNS, the, no applications or clients will use the Lag-Site DCs for normal authentication, so rebooting the DCs into DSRM mode will not have any additional impact on the rest of the infrastructure.

Challenges with Lag-Sites

Although the use of Lag-Sites has become fairly popular to add protection to large AD infrastructures, you need to be aware that there are some caveats with Lag-Sites. Although we are able to configure the replication schedule to hinder normal replication of changes to DCs in the Lag-Sites, there is no way to fully disable replication of changes coming through to Lag-Sites when using *forced* replication.

Forced replication can be achieved by the following **repadmin** command:

```
repadmin /replicate DC1 DC=mydom,DC=com /force
```

But it is also achieved quite simply by using **ReplMon** from the support tools and *pushing out* changes from a DC. As there is no feature in AD to push changes, this command basically forces the respective replication partners to perform a *pull-replication* of the changes. Only a DC that is offline can be completely prevented from replicating.

Also take into account that a single DC per domain in a Lag-Site may not be enough. It was previously mentioned, that an unwanted change (e.g., deletion of objects) could overwrite the data on the DCs in the Lag-Site by occurring just prior to the planned replication schedule. So you should use at least 2 DCs per domain with alternating replication schedules.

Last but not least, even when using DCs in Lag-Sites to restore data via the native Authoritative Restore, you still need to handle the recovery of linked objects. Naturally, if all Lag-Site DCs are running Windows Server 2003 with SP1, the process has improved a bit but is still cumbersome for multi-domain infrastructures so that the challenges for restoring Back-Links (e.g., group-memberships for users) remain.

**Improvement in
Windows Server 2003 SP1**

SP1 Improvement: Better protection for false restores of DCs as Virtual Servers

In Q4 2004, Microsoft released an important white paper describing how to [Run Domain Controllers in Virtual Server 2005](#). Besides discussing potential usage scenarios (e.g., a branch office DCs running as a separate VM on a server also hosting other services for branch), the white paper also details requirements that must be met before Microsoft can support DCs that are running on Virtual Servers. These are also important to understand when hosting DCs in a Lag-Site as Virtual Servers.

One such criteria is a special hot fix ([875495](#)) that needs to be applied to the **Virtual Domain Controllers**. This provides protection against directory corruption that can result from improper backup and restoration of DC images.

What is the issue with these Virtual Server images? There is no general issue in taking an image of a Virtual Server or VMware for backup and fast recovery of virtual machines – even if they are running as DCs. This is a powerful feature of all the virtualization solutions, and it allows quicker recovery to a known state than many tape-based backup solutions do. Generally, they will outperform an OS level backup.

These images allow us to efficiently restore a virtual machine to a previous state, i.e. to put it "back in time". And this is where the problem lies, as a DC typically replicates its data with other DCs in the domain or forests. The replication process between DCs is strongly controlled so that every DC will only request changes that have happened since the last time it contacted its replication partner. It does this by remembering the last value of the **Update**

Sequence Number (USN) for each replication partner, e.g., USN of DC1 = 750 and USN of DC2 = 200. This number is increased for any change that happens on a particular DC so that a replication partner will always expect the next USN value of its replication partners to be equal or larger to the one it already knows.

If one DC is now taken "back in time" by restoring a previous image of the virtual server and is then brought online without any changes, the DC will try to replicate with its replication partners just like it did before. For example, a restore would bring DC1 back to a USN of 700 even though DC2 still thinks it has all the changes of DC1 up to USN 750. In this case, DC2 would not try to replicate any changes from DC1 until another 51 changes are made on DC1. It would then request the change that corresponds to USN 751. The changes made between USN 701 and USN 750 are of no interest to DC2, as it believes it already knows of these changes (but it doesn't since there are 50 new changes that occurred after the restore of DC1). This issue is referred to as *USN rollback*: DC1 was rolled back from USN 750 to USN 700 and thus the replication process to its replication partners was broken.

Why doesn't a USN rollback cause issues with normal restores of DCs? Even a DC that was restored normally from a system state backup is rolled back to a previous USN number. The reason that this doesn't cause the same issues as when restoring from a virtual server image is due to a special feature that allows this DC to *inform* its replication partners of its change in status - the **Invocation ID**. This ID is basically a version-id for the AD database on the DC, and it gets changed every time a DC's system state is restored. If a DC notices a change in the invocation ID of a replication partner, it knows it must bring the respective DC up to date and replicate the latest changes that occurred since its backup.

When restoring a DC with an image of a virtual server, the invocation ID does not change, and thus the USN rollback is not expected by its replication partners, causing the issues previously described. To limit the damage that can happen with an improper DC restoration, Microsoft requires that you apply hotfix 875495 on DCs running as Virtual Servers (see [Q875495 - How to detect and recover from a USN rollback in Windows Server 2003](#)).

This fix ensures that when a source DC sends a previously acknowledged USN number to a destination DC without a corresponding change in the invocation ID that replication on the source DC is halted and the system stops advertising as a DC.

The USN rollback is logged as event ID 2095 (NTDS Replication) and 2103 (NTDS General) and triggers three other actions:

1. Stops inbound replication
2. Stops outbound replication
3. Pauses Net Logon service

When the Net Logon service is paused, user and computer accounts cannot change the password on a DC that will not outbound-replicate such changes. Similarly, AD administration tools will favor a healthy DC when they make updates to objects in AD.

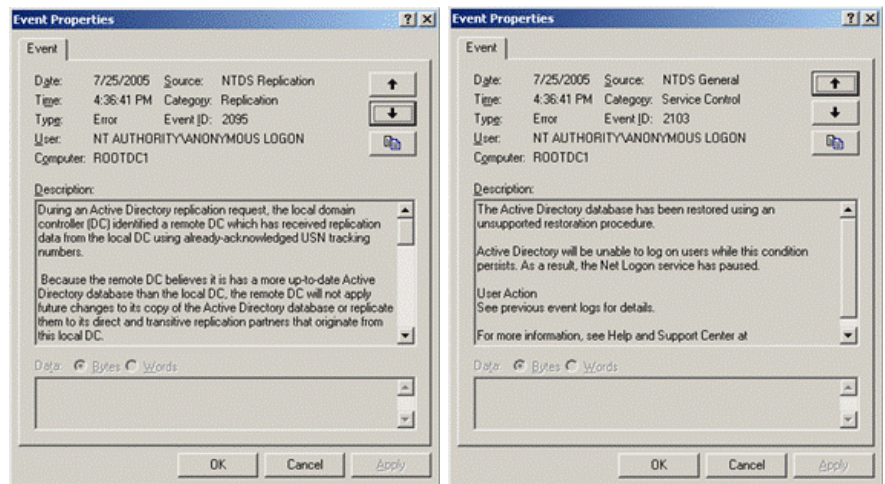


Figure 13: Error messages related to USN Rollback

This protection of hot-fix 875495 has now also been integrated into Windows Server 2003 SP1. Note that USN rollback can also occur by leveraging other means to create images of existing hardware DCs (e.g., Norton Ghost) and restoring that image so that the SP1 rollout will not only protect DCs running as Virtual Servers, but any DCs that could potentially be restored in an unsupported manner.

When a DC has reached the "USN rollback" status and is blocked from replication and authentication it can be repaired in two ways. When another system-state restore exists for this machine this can be used for a non-authoritative restore on the server – this will change the invocation ID of the database and thus everything is back to normal. If there is no existing backup, it's still not too late to make one (providing no other changes have been made that are expected to replicate out). This should immediately be followed by a non-authoritative restore, which will again change the invocation ID of the database, thus returning everything back to normal.

Summary

Using Lag-Sites is a very effective way of establishing a "hot backup" of Active Directory. Lag-Sites leverage Active Directory's scheduled inter-site replication to maintain one or more DCs in each domain as hot backup domain controllers that have relatively recent copies of all the domain data, without requiring an explicit backup process. Lag-Sites will save you a lot of time when you have to restore AD objects, either using the authoritative restore process, or the by reanimating the tombstones of deleted objects. Using Microsoft Virtual Server or other virtualization products can reduce the cost and complexity of creating a Lag-Site.

THIRD-PARTY BACKUP AND RESTORE SOLUTIONS

There are many Windows-compatible backup and restore solutions available from third-party software vendors. Some of the top-selling products include Veritas Backup Exec, Ultrabac for Windows, and CA ARCserve. Generally, these products replace the native Windows backup and restore tools, and provide additional ease-of-use, management, scalability, and performance features.

Like the Windows backup and restore tools, these solutions can backup and restore the entire DC, including the AD DIT. But because these products can only restore the entire AD DIT, not individual AD objects, they are useful for restoring DCs, but are not particularly useful for restoring specific objects or groups of objects that may have been deleted or changed inappropriately in AD.

NetPro's RestoreADmin

One of the more common “failure” scenarios in AD is the accidental deletion of one or more objects. It is not difficult for an administrator in a hurry to “fat-finger” the deletion of a user or computer, or even an entire OU. Recovery from these types of failures using the native tools is tedious at best.

NetPro's RestoreADmin provides the quickest and easiest-to-use solution for restoring deleted AD objects. RestoreADmin snaps in to the Microsoft Active Directory Users and Computers MMC, providing a new “Recycle Bin” container for each AD domain naming context, displaying the tombstone objects representing the deleted objects.

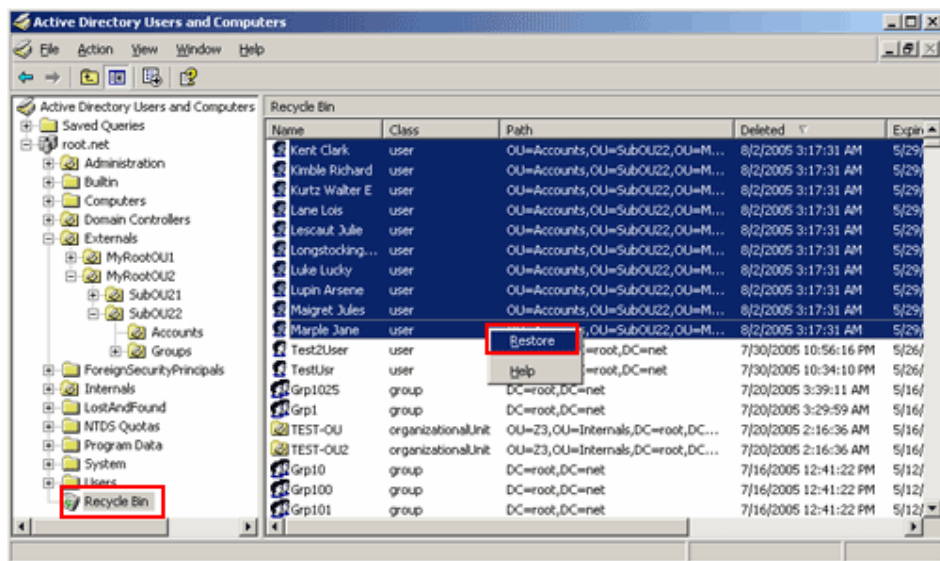


Figure 14: RestoreADmin adds a Recycle Bin to ADUC. Restoring objects and their critical attributes is as simple as selecting them and clicking “Restore.”

Restoring a deleted object to its previous state couldn't be simpler. Simply click on the deleted object in the Recycle Bin, and select the **Restore** option from the context menu. RestoreADmin automatically reanimates the tombstone object, repopulating its critical attributes, and placing the object back in its original container. You can quickly restore single objects, groups of objects, or entire OUs with one mouse click.

RestoreADmin allows you to schedule automatic backups of preconfigured sets of AD objects to a SQL Server database. This ensures that your backups are timely, scalable, and use the minimal amount of system resources.

Features

- Recover deleted objects while the DC is still online, including Users, Computers, Groups, Contacts, OUs, Associated Attributes and Group Memberships
- Recover security descriptors, sIDHistory, and passwords^[10]
- Create and schedule centralized backups of supported objects
- Rollback existing supported objects to the previous backup state
- Snaps into Microsoft's Users and Computers interface

To learn more about NetPro's RestoreADmin, please visit <http://www.netpro.com/products/restoreadmin>.

^[10] Recovering sIDHistory and passwords requires modification of the searchFlags of these attributes as described in the Tombstone Reanimation section of this paper.

CONCLUSION

Active Directory is the most critical part of the security infrastructure of your Windows network, and it requires special consideration as you develop your Business Continuity and Disaster Recover Plans.

Recovering deleted data in Active Directory is not a trivial exercise. The way Active Directory manages linked objects such as groups and the users that are members of those groups, adds significant complexity to the data recovery process.

Recovering a failed DC is a fairly straightforward process, but still requires care on your part, even with the improvements in NTDSUTIL metadata cleanup function and the ability to build a new domain controller from media using the new IFM features.

Recovering a failed domain or a failed forest to a previous state is not a project you should approach lightly. It requires a lot of planning and organization on your part, and is generally the plan of last resort.

Finally, using a tool like NetPro's RestoreADmin can significantly reduce the workload associated with recovering specific objects and OUs in Active Directory.

NETPRO CONTACT INFORMATION

NetPro has been ensuring the health and performance of network directories since 1991. Now the leading provider of directory operations management software, NetPro offers a full suite of solutions to manage Active Directory operations.

For the latest information on RestoreADmin and the other solutions that make up this suite, visit our web site at: <http://www.netpro.com/products.cfm>.

Corporate Office:

4747 N 22nd Street, Suite 400

Phoenix, AZ 85016-4774 USA

Telephone: 602-346-3600

Fax: 602-346-3610

Email: info@netpro.com

Internet: <http://www.netpro.com>

European Office:

Telephone: +31.36.540.5959

Sales:

USA: 800-998-5090

Canada: 888.858.9220

International: +1 602 346 3630

Worldwide Technical Support:

Telephone: 602-346-3670

Monday - Friday 06:00 - 18:00 MST (-7GMT)

Email: support@netpro.com