

Chapter 3: Preparing Your Windows NT Network

In this chapter, I'll talk about the work you need to do to prepare your Windows NT 4.0 domain environment to migrate to Windows 2000 (Win2K). Depending on which approach you take—in-place upgrade, security identifier (SID) history migration, and so on—you may have to spend a fair amount of time “cleaning up” your NT 4.0 infrastructure before you upgrade. In fact, many companies look at migrating to Win2K as a chance to “clean house” and exorcise the demons that have settled into their existing infrastructure.

No matter what approach you take, the first step in migrating to Win2K is planning your Active Directory (AD) namespace—how you'll structure your AD domains and Organizational Units (OUs)—and how your current NT 4.0 domain environment maps to it.

Planning the AD Namespace

As the previous chapters of this book have shown, a lot of work will go into planning an AD (and Win2K) implementation, not the least of which is planning your AD namespace.

What is an AD namespace plan? Basically, this is the process of deciding how many AD domains, domain trees, and forests your organization will have and, in each domain, how your OUs will be organized. As you can imagine, the design of an AD namespace has a direct relationship on your migration process. That is, you need to decide how the domains, users, groups, and computers in your NT 4.0 environment will map to the AD domains and OUs in your AD namespace.

In fact, the design of your AD namespace will likely be largely influenced by your current NT 4.0 infrastructure. For example, if you have an NT 4.0 infrastructure composed of 20 master account domains and 200 resource domains—not an uncommon scenario—you may find yourself unable to migrate to a single AD domain, and thereby unable to meet the business goals of your Windows infrastructure, simply because the process is too complex and time-consuming. However, if your NT 4.0 infrastructure is composed of one or two domains, developing an AD design that calls for a single domain is completely viable.

Designing the Namespace

So how do you begin to design an AD namespace that lets you move NT 4.0 into the world of AD? Here are a few facts to help you:

- AD domains are more scalable than NT 4.0 domains, which are typically limited to the size of a 40-MB Security Account Manager (SAM). This means that it's perfectly reasonable, and probably desirable, to reduce the number of domains you have when you migrate from NT 4.0 to Win2K.
- Resource domains in NT 4.0 are built to allow delegation of administration for resources like servers and workstations. In Win2K and AD, OUs serve this delegation role. Thus, mapping multiple NT 4.0 resource domains to OUs in a single AD domain is a common approach.

- The domain is a security boundary in both NT 4.0 and Win2K. This means that in both versions of the operating system (OS), security policy such as password length and intruder lockout is set at the domain level. If you need different account policies for two groups of users, they need to be in separate domains.
- The AD forest is a boundary that can contain multiple domain trees and domains. A single forest shares a common Global Catalog (GC) and schema. Multiple forests have no relationship to each other. In Win2K, you can't manage multiple forests easily, and there is no security relationship among them. Domains, resources, and users in one forest must establish explicit, non-transitive trusts with domains in other forests.

Keeping these principles in mind, I'll describe some different approaches you can take to plan your AD namespace. Remember that there are an infinite number of ways to arrange an AD implementation. The ones I show here are just a few common approaches. You may find any number of better ways to design your namespace, and as long as the result is manageable and flexible over time, that is just fine.

The most important criterion I'd place on any AD namespace design is that you do whatever you can to minimize the impact of the inevitable reorganizations, mergers, and acquisitions that your company is likely to go through. To do this, avoid building domains along departmental lines. If you need to organize by department, OUs are more flexible and more easily torn down or renamed if your company's structure changes.

In the following examples, I'll show how you can migrate some common types of NT 4.0 namespace designs to AD.

Migrating a Single NT 4.0 Domain to a Single AD Domain

The first example, shown in Figure 3.1, is the simplest case—migrating a single NT 4.0 domain to a single AD domain.

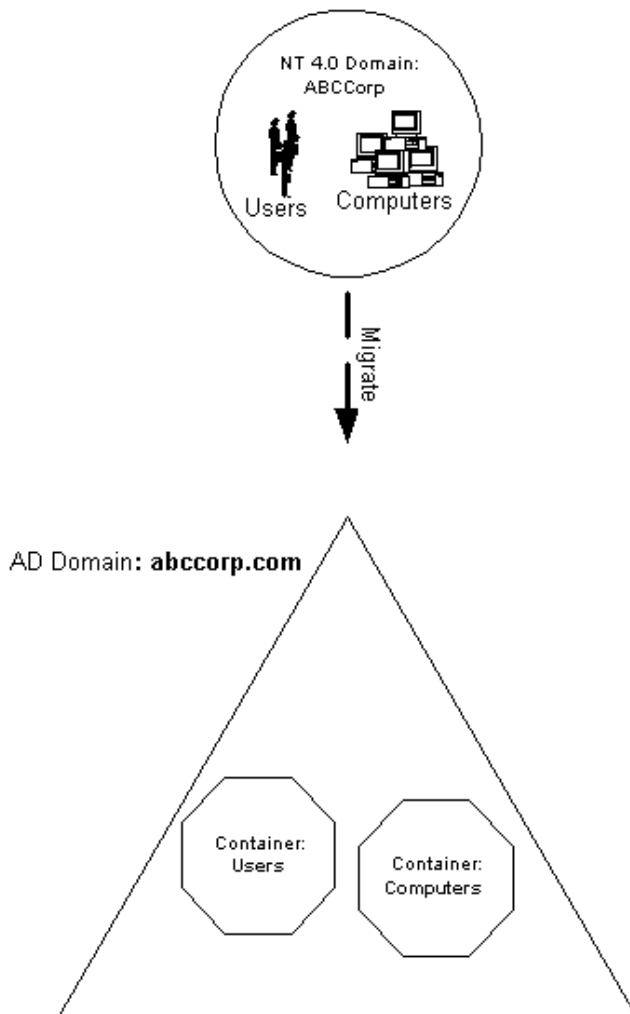


Figure 3.1: Migrating a single NT 4.0 domain to a single AD domain.

The figure above shows an NT 4.0 domain called ABCCorp containing users and computers. When you upgrade ABCCorp to Win2K, the result is an AD domain called abccorp.com. In this new AD domain, the user accounts are placed in a container called Users, and the computer accounts are placed in a similar container called Computers. These containers aren't true OUs, but rather holding containers that serve as a temporary storage for the domain objects until you're ready to move them into OUs that you create.

Migrating a Single-Master NT 4.0 Domain to a Single AD Domain

The next example starts with a fairly common NT 4.0 domain model—the single-master model. In the single-master model, a single master account domain called ABCMaster contains users and groups plus any number of resource domains that contain machine accounts (such as servers and workstations). The resource domains have a one-way trust relationship to the master account domain to facilitate access to resources. Figure 3.2 shows the layout of such an NT 4.0 domain model.

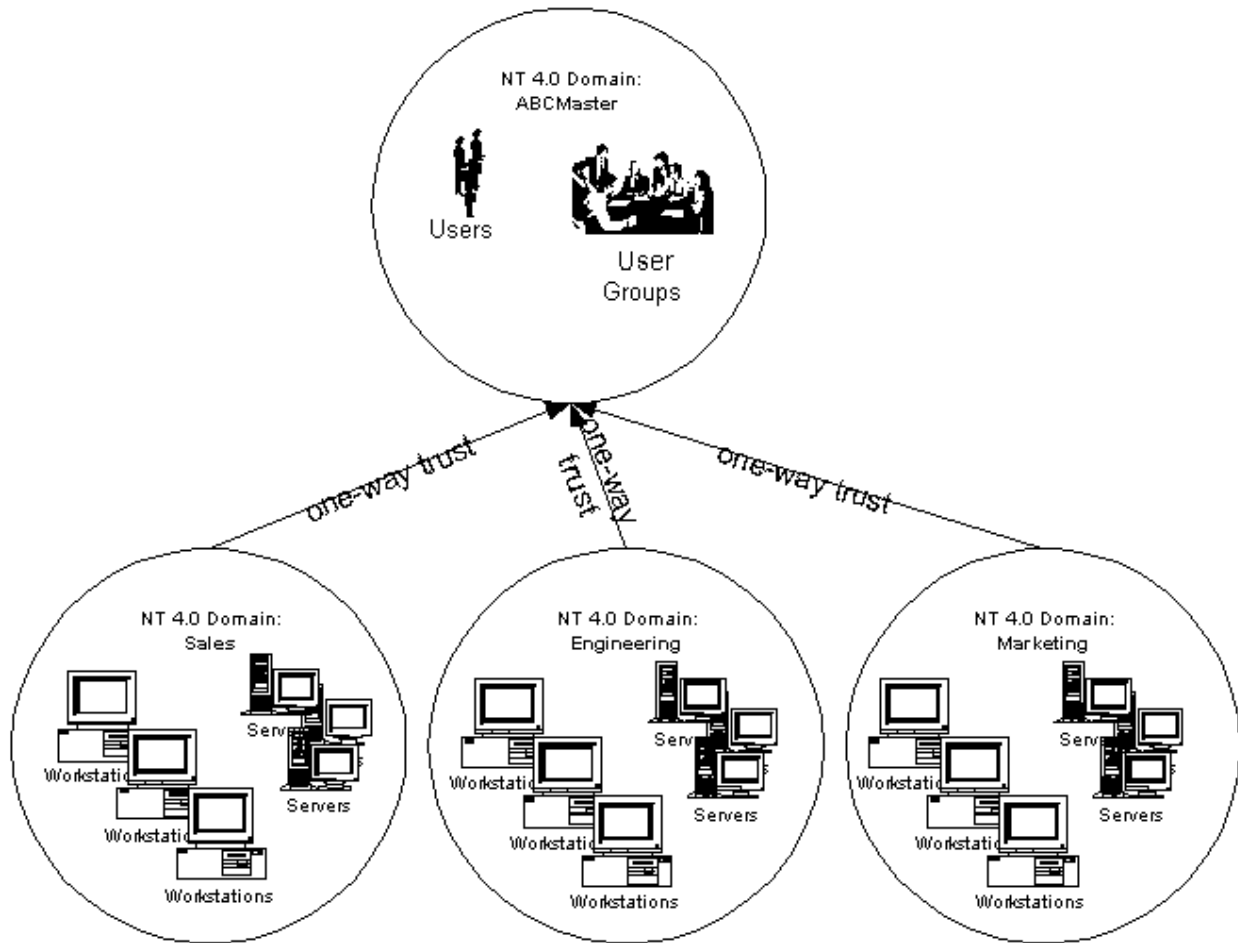


Figure 3.2: A single-master NT 4.0 domain model.

If you then design an AD namespace to accommodate the domain shown in Figure 3.2, it might look something like Figure 3.3 below.

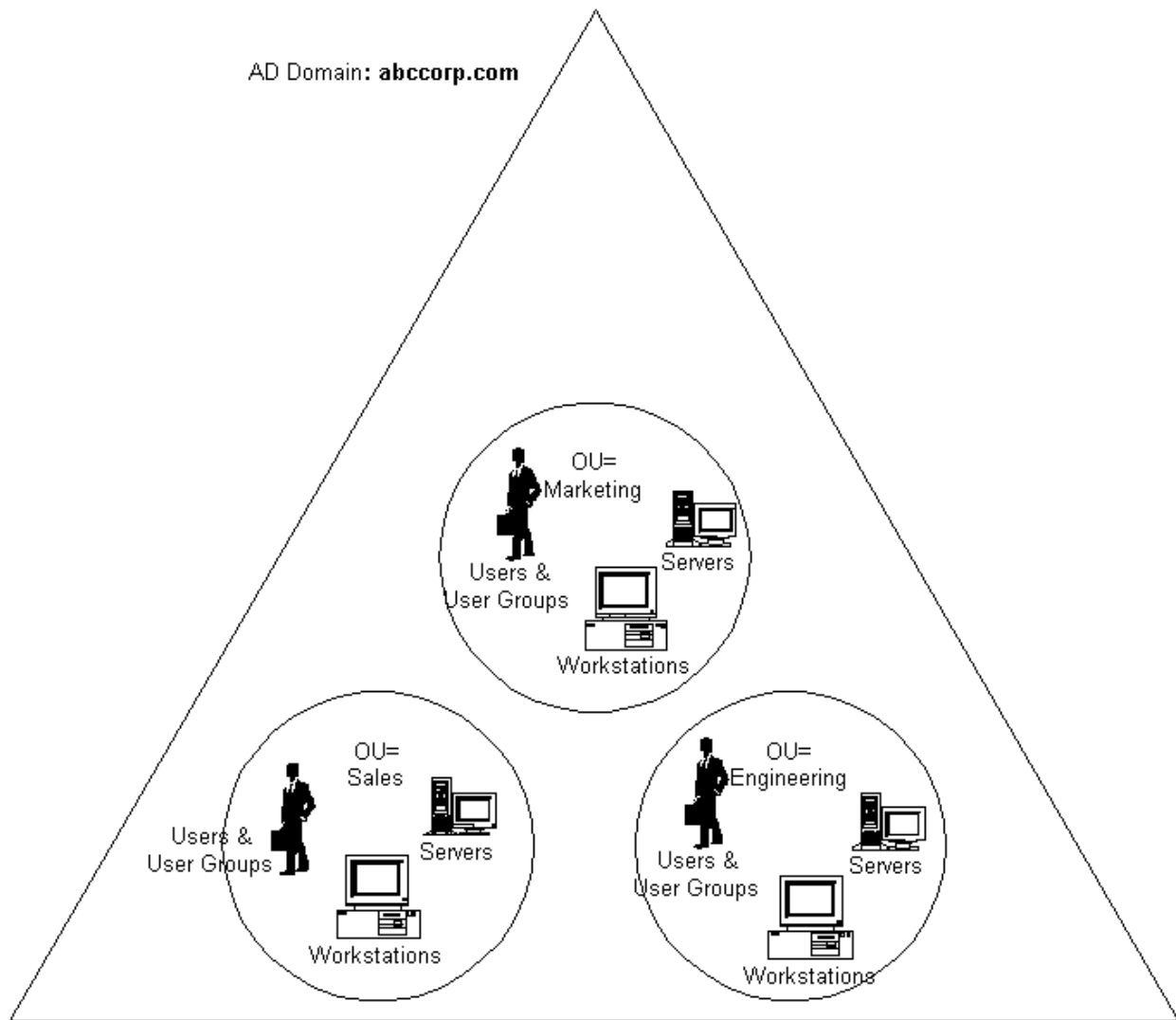


Figure 3.3: An AD domain after migration from a single-master NT 4.0 domain.

As you can see in Figure 3.3, the four NT 4.0 domains in Figure 3.2 have been subsumed by a single AD domain called **abccorp.com**. In this AD domain, three OUs replace the previous NT 4.0 resource domains, and each OU has the same name as its predecessor. The user accounts and user groups that existed in the NT 4.0 master account domain called **ABCMaster** are also folded into the three OUs. Because the AD namespace takes four NT 4.0 domains into a single AD domain, you can use OUs instead of resource domains as a way of delegating control of servers and workstations (as well as users and groups). In fact, this is a very common approach. OUs are the “unit of delegation” in Win2K and AD, allowing the need for resource domains in AD to go away completely.

Migrating a Multimaster NT 4.0 Domain to an AD Namespace

The final example starts with a fairly complicated NT 4.0 domain model—namely, a multimaster domain model. This model contains multiple master account domains, each of which is trusted by a number of resource domains. Such an approach is typically used in very large NT 4.0 infrastructures, either because a single master account domain is not sufficiently scalable to meet the needs of the enterprise or the organization is structured in such a way that it needs multiple master account domains to support it. Figure 3.4 shows an example of such a domain.

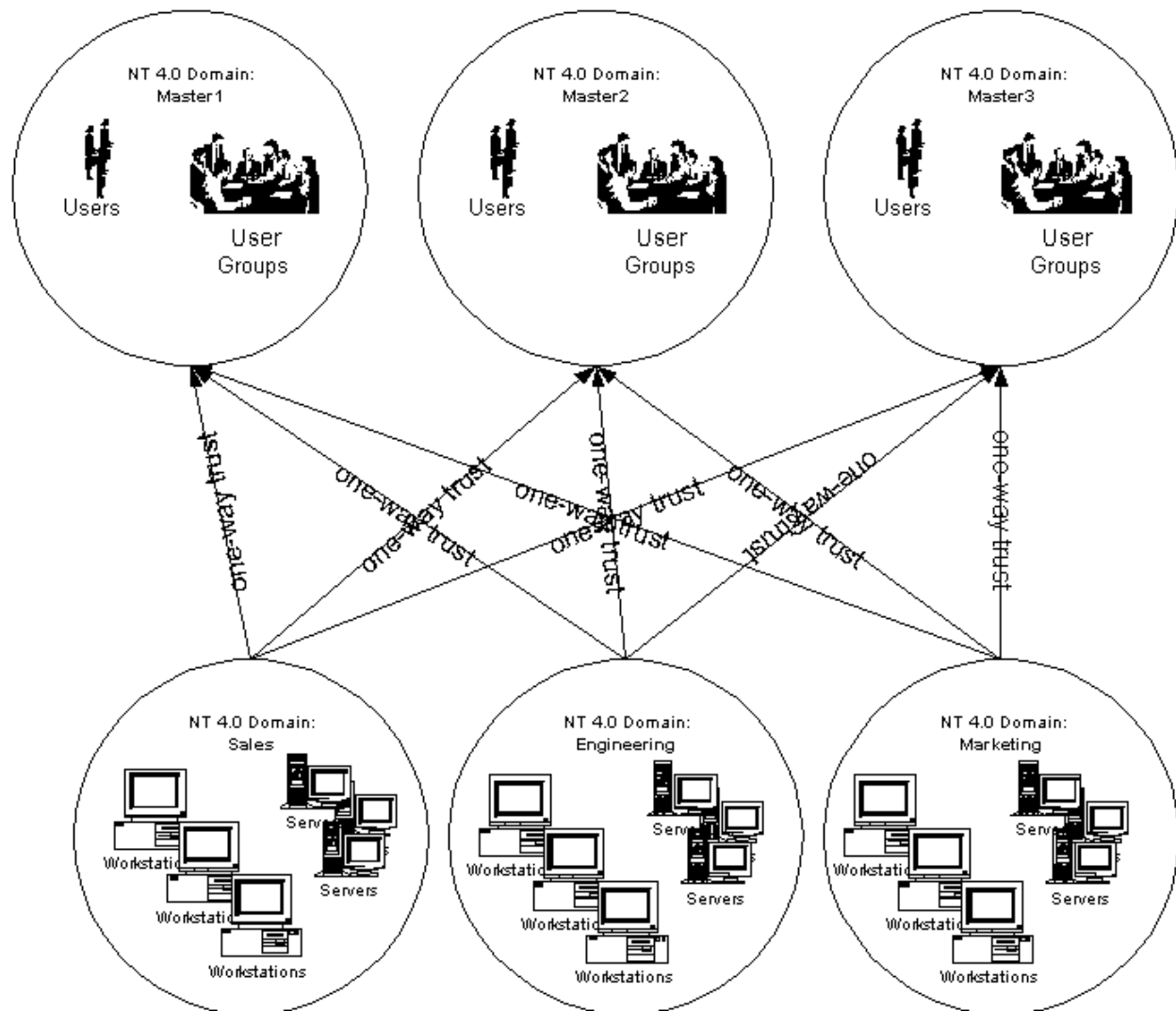


Figure 3.4: An NT 4.0 multimaster domain model.

As you can see in the figure above, the non-transitive nature of NT 4.0 trust relationships means that each resource domain—Sales, Engineering, and Marketing—must have a one-way trust relationship with each master account domain. The resulting spider web of trusts is one of the big challenges and limitations of NT 4.0 domains. Each resource domain contains workstation and server machine accounts that are usually administered by a local business unit's administrators.

However, user accounts reside in one of the master account domains, which may be organized regionally or by business division.

Using trust relationships, users in any of the three master account domains shown in Figure 3.4 above can have access to any of the workstations or servers in the three resource domains (provided permissions to those resources have been granted). I've also thrown into the mix the fact that a different organization or business unit may administer each master account domain, which is different again from the administrator of each resource domain. The challenge, then, is coming up with an AD namespace design that preserves the strong points of this multimaster domain approach, while at the same time taking advantage of the best that AD has to offer.

The Organizational Challenges of Migrating to AD

The changes that AD brings might suggest or even dictate a change in the way your Information Technology (IT) support organization is structured. For example, NT 4.0 domains lend themselves to a very decentralized support structure, where each domain is an island that can be administered independently from the others. However, AD's more scalable nature and more integrated approach can lead you to combine your many existing NT 4.0 domains into fewer (or even a single) AD domain. This is a good thing, but it may also put a strain on your existing decentralized support structure. This has been a common challenge in Win2K migrations that I've seen, and you should be prepared to deal with such a likelihood in your own organization. AD requires a higher level of cooperation and communication among independent administrative groups. Allow time in your AD planning process to discuss with your IT management how this new type of organization might best be handled.

Now let's look at one approach for migrating the domain design shown in Figure 3.4 above to AD. Let's assume that of the three master account domains, two of them—Master1 and Master2—are administered by a central IT organization in the United States. The third—Master3—is the European division of the company, and it's supported by a different administrative group. Of the three resource domains, each contains computers that are physically located in both the US and Europe. That is, some workstations in the Sales domain are physically in both the US and Europe, and the same is true of the Engineering and Marketing domains. Figure 3.5 below shows one approach to designing an AD namespace that meets the needs of this organization.

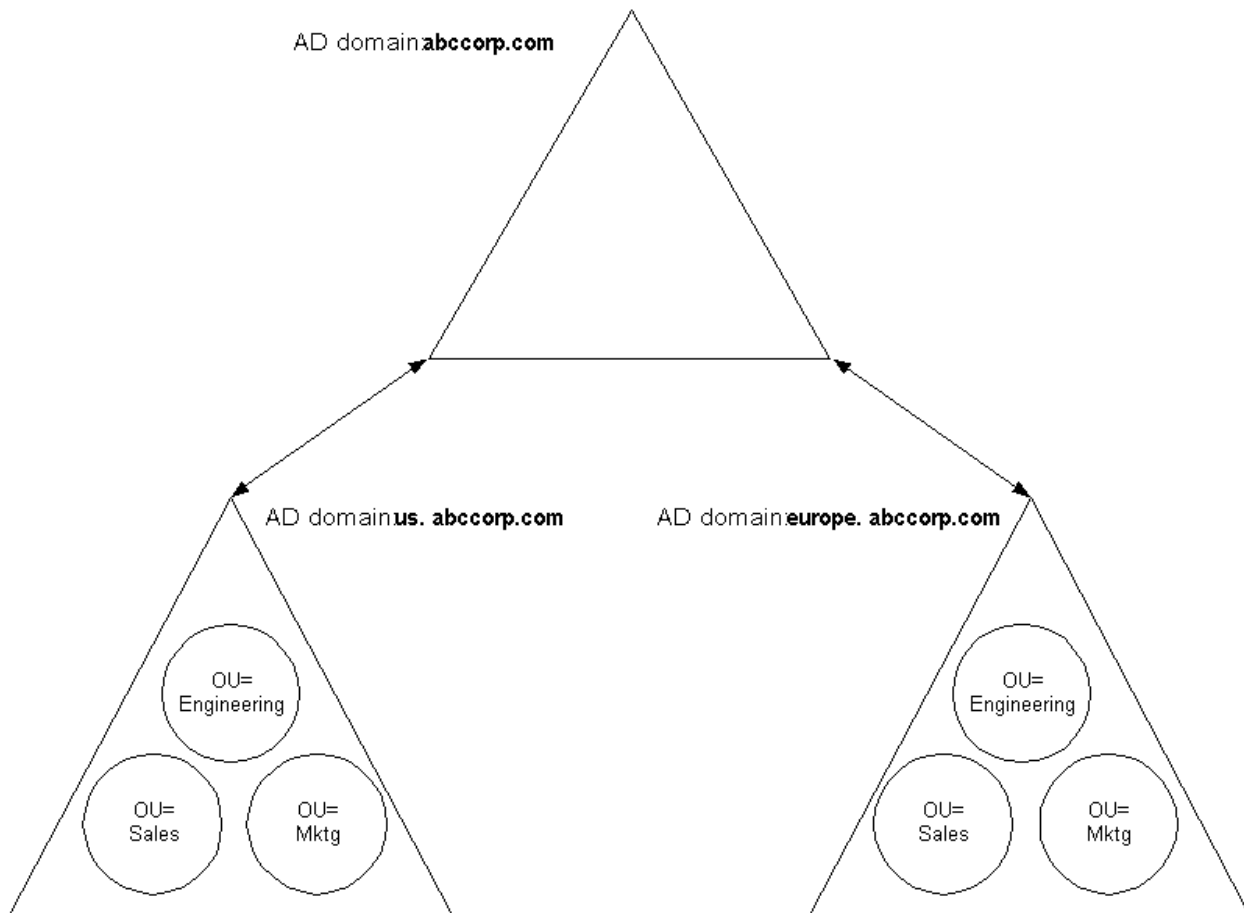


Figure 3.5: An AD namespace that has been migrated from an NT 4.0 multimaster domain model.

Figure 3.5 shows three AD domains—`abccorp.com`, `us.abccorp.com`, and `europe.abccorp.com`. The first, `abccorp.com`, is the forest root domain because it's the first domain in the forest. Because this domain has a special role in the forest, you often see AD namespace designs that leave this domain empty, or unused—as a sort of infrastructure domain. This is the approach shown in this figure. In addition, two child domains have been created under `abccorp.com` called `us` and `europe`. These two child domains take the place of the master account domains in the previous NT 4.0 model. Master1 and Master2 user accounts have been combined into the single `us.abccorp.com` domain and Master3 into `europe.abccorp.com`.

You can also see that each new AD domain has identical OUs, which correspond to the three resource domains in the previous NT 4.0 infrastructure. Each OU in each domain contains user and group accounts that were previously in the NT 4.0 master account domains, as well as workstation and server accounts that were previously in one of the three resource domains. Table 3.1 gives a brief summary of how these NT 4.0 objects are mapped to the new AD design.

This NT 4.0 Object	Is Mapped to This AD Object
Master1 and Master2 users and groups	One of the three OUs in us.abccorp.com, depending on whether they're Sales, Engineering, or Marketing users.
Master3 users and groups	One of the three OUs in europe.abccorp.com, depending on whether they're Sales, Engineering, or Marketing users.
Workstations and servers in the Sales domain	The Sales OUs in either us.abccorp.com or europe.abccorp.com, depending on whether the workstations and servers are physically located in the US or Europe.
Workstations and servers in the Engineering domain	The Engineering OUs in either us.abccorp.com or europe.abccorp.com, depending on whether the workstations and servers are physically located in the US or Europe.
Workstations and servers in the Marketing domain	The Mktg OUs in either us.abccorp.com or europe.abccorp.com, depending on whether the workstations and servers are physically located in the US or Europe.

Table 3.1: How objects in the NT 4.0 domain in Figure 3.4 are mapped to the AD namespace in Figure 3.5.

As you can see from this table, workstations and servers that were previously in three separate resource domains based on business function are now stored in geographic domains and, in those domains, in business function–based OUs (for example, Sales, Engineering, and Marketing). This approach provides, as I mentioned earlier, a fairly static domain structure (that is, the geography of Europe and the US rarely changes!) but a flexible OU structure, which can be changed as the business changes.

In addition, administrators who previously had to manage users in both Master1 and Master2 can now manage users in a single domain—us.abccorp.com—albeit across three OUs. In each OU, furthermore, administrators can delegate administration of workstations and servers to the earlier resource domain administrators so that they retain control of those resources.

Generally speaking, when you design your AD namespace, and unless you have an overriding reason to maintain domain boundaries, always seek to reduce the number of domains in, and thus the complexity of, your Windows infrastructure. Of course, you'll have to do more work up front to migrate from your existing NT 4.0 infrastructure, but it can be worth the effort. As many NT 4.0 environments have grown, they've become disorganized and complex, and they'd greatly benefit from the “starting over” approach that a significantly new AD namespace provides.

Cleaning Up the Network

The amount of “cleanup” work you need to do before you migrate from NT 4.0 to Win2K and AD depends to a large degree on what method you use to migrate. Chapter 4 talks extensively about your options, but suffice it to say that if you choose the in-place upgrade method, the amount of cleanup you'll positively *need* to do is pretty small. Still, always consider such a migration an opportunity to “clean out the cobwebs”.

An *in-place upgrade*, as the name implies, lets you upgrade your existing NT 4.0 domain controllers to Win2K, thereby installing AD in place of your NT 4.0 domain(s). Everything remains the same in AD as it was in NT 4.0—same users, same computer accounts, same groups, and so on. The challenge comes when you decide that, as part of your AD migration, you’ll collapse (that is, reduce) the number of NT 4.0 domains into fewer AD domains. You’ll also face challenges if you decide to start from scratch and build new AD domains to which you then migrate your NT 4.0 users and resources.

And what are these challenges? The principal issue that you have to deal with as you migrate users and resources to Win2K and AD is access to resources. Most companies don’t have the luxury of allowing their users to go home on Friday night, then come in on Monday morning to a fully deployed AD infrastructure. Migrations take time, planning, and ultimately coexistence between the servers, workstations, and users in both NT 4.0 and Win2K environments. While it’s not as problematic as it may at first sound, such a scenario does require a lot of planning.

The crux of the issue is really how you’ve granted permissions on resources such as printers, files, and file shares and how you plan to migrate these resources. As you know, you can grant permissions on resources so that specific users and groups have access to use, manage, and modify them. If the users or groups to whom you’ve granted access are domain users and groups, your goal is to continue to provide seamless access to these resources—which may still be running on NT 4.0 servers—while you migrate your NT 4.0 domain users and groups to AD running on Win2K domain controllers.

Table 3.2 lists the most common types of NT 4.0 resources to which you need to maintain access as you migrate users to AD.

These Resources	Have These Permissions Considerations
Files and file folders	On NT file system (NTFS) volumes, if files reside on a member server, you can grant permissions by domain local group, domain global group, or local group. If files reside on a member server, you can grant permissions by domain user or local user.
File shares	You can grant permissions using domain or local users and groups.
Printers	You can grant permissions by domain local group, domain global group, or domain users as well as by local groups and local users.
Services	Some may run using domain accounts for security. This information is kept in the Registry on the server or workstation where the services are installed. If accounts change as part of the migration, service permissions need to change as well.
User profiles	The user account that owns a profile typically has the rights to use it, but you can also grant permissions on profiles using user group.
Mailboxes	If you’re running Microsoft Exchange as your corporate e-mail software, you typically grant permissions on your NT 4.0 user accounts to a given mailbox or distribution list.

Table 3.2: The common NT 4.0 resources to which users and user groups have access.

Table 3.2 is by no means complete, but it covers most of the resources that you need to be concerned with during a migration. You may also have to contend with Remote Access Service (RAS), Internet Information Server (IIS) resources, and other Microsoft products that use NT’s

security model. However, the real problem you'll have as you migrate resources and users from your NT 4.0 domains to AD is the fact that each user and user group is identified by a unique *Security Identifier (SID)*.

The SID is what is really assigned to a file or folder when you allow a user or group to access it. If the SID for that user or group goes away—because the user or group is retired or re-created in a new AD domain (thus receiving a new SID)—the user who used to have access to that resource before it was migrated to AD won't have access afterward. This is such a key point that I'll spend the next few sections of this chapter discussing it—how to plan for and work around the loss of access to resources as users and groups are moved or retired.

Handling Permissions Issues

Not all resources will cause you problems as you migrate to Win2K. In this section, I'll describe in detail the areas you need to be most concerned about.

Let's suppose you're migrating from a multimaster NT 4.0 domain model to a fewer number of Win2K AD domains. (This is discussed in "Migrating a Multimaster NT 4.0 Domain to an AD Namespace" earlier in this chapter and illustrated in figures 3.4 and 3.5.) I'll call this Scenario A. In this scenario, you need to collapse some of your NT 4.0 domains, most notably the resource domains, into fewer AD domains. You also need to move user accounts from three NT 4.0 master account domains into two AD domains. The method of migration you choose will affect how many permissions you need to clean up before and after migration.

For the Three Migration Approaches

Table 3.3 summarizes the permissions issues to consider in each of the three major migration approaches. (Chapter 4 will discuss each approach in detail.)

This Migration Approach	Raises These Resource Permissions Issues
In-place upgrade	Few if any because you maintain all user and group SIDs as they were in NT 4.0. However, if you upgrade an NT 4.0 domain in place, then collapse it into another AD domain, some issues may arise.
Re-create new users in a new AD infrastructure	All SID information from the NT 4.0 domains is lost. To continue to give these new users and groups access to resources, you need to grant permissions again to all resources with the new user or group accounts.
Clone NT 4.0 users to a new AD infrastructure (using SID history)	This Win2K-specific approach preserves the NT 4.0 SIDs for these users and groups, while at the same time giving them new "primary SIDs" in the AD. These new users will still have access to their resources by virtue of their cloned SIDs, and they'll also have access to resources granted to their new AD accounts or groups.

Table 3.3: Major AD migration approaches and their effect on resource permissions.

This table shows that an in-place upgrade will likely create the least amount of work to ensure that migrated users can still access their resources. The two other approaches are commonly referred to as the "pay now or pay later" approaches.

If you re-create user accounts and groups in a new AD domain, you need to grant permissions again to all of the resources that were accessed by those users and groups as soon as you've migrated those users; otherwise, the new accounts won't have access to the resources.

Using the cloning method and SID history, however, you can be assured that users should still have access to all of their NT 4.0 resources. At some point, as you retire your old NT 4.0 domains, you'll need to clean up those resource permissions and apply access to the new AD user accounts and groups so that you can stop using the SID history feature.

What Is SID History?

As the name implies, SID history, first provided in Win2K, allows a user or group to contain more than one SID. In the context of our migration discussion, it means that when you create a new user or group account in AD, it can also hold, as one of its security attributes, a list of other SIDs that should be associated with it. When Win2K or NT determines whether a user or group has access to a resource, it checks not only that user's or group's primary SID but also its SID history. If one of the SIDs in SID history is allowed to access the resource, permission is granted.

For Scenario A

Now, when it comes to ease of migration, not all access permissions are created equal. Let's go back to Scenario A and decide how to migrate users, groups, and resources to the new AD environment. Assume that you'll be re-creating users and groups from scratch in your new AD domains, then migrating resources (workstations and servers) from NT 4.0 to Win2K. Figure 3.6 lays out this approach.

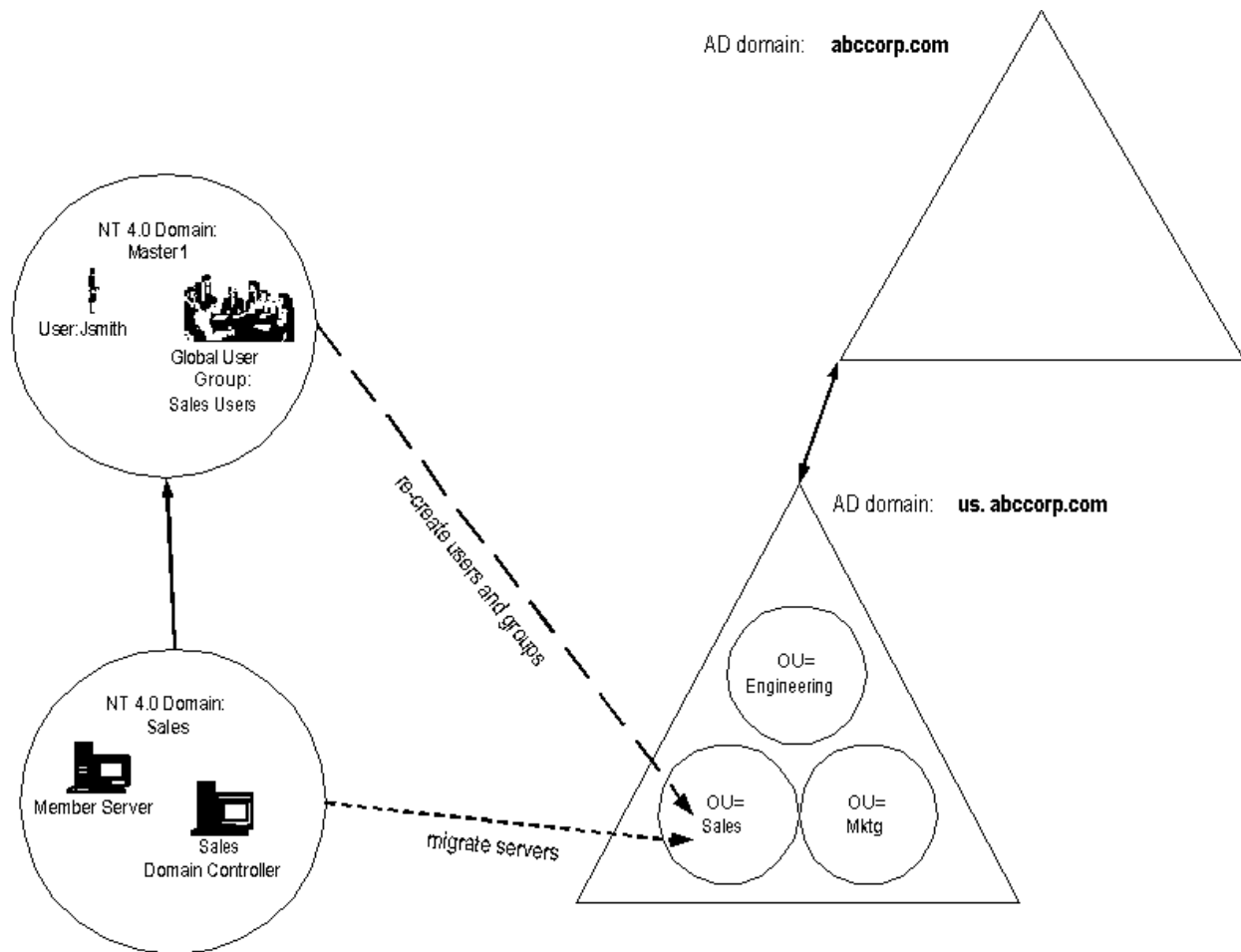
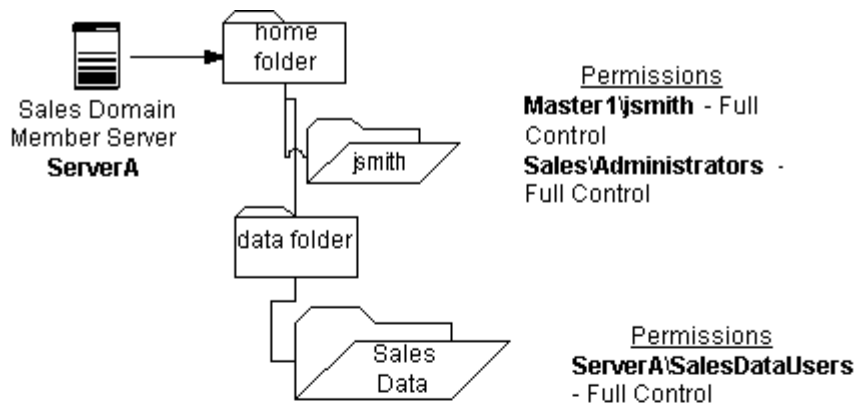


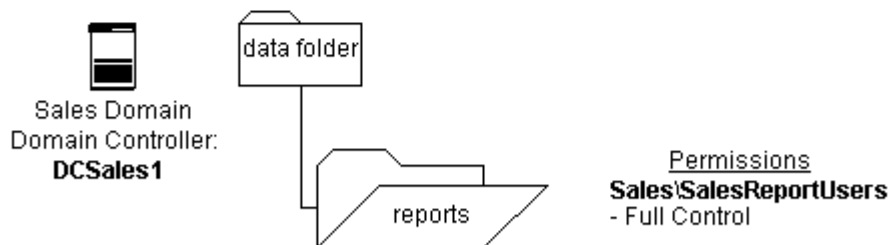
Figure 3.6: One approach to migrating from NT 4.0 to AD for Scenario A.

The figure above shows a part of the NT 4.0 environment introduced in Scenario A earlier in this chapter (see “Handling Permissions Issues”)—namely, a master account domain called Master1 and a resource domain called Sales. In Master1, I’ve highlighted a user account called Jsmith and a user group called Sales Users. The resource domain contains two servers—a regular member server and a domain controller for the Sales resource domain. There are file resources on each server, and you want to move the two servers and their resources into the OU in us.abccorp.com called Sales. As I mentioned earlier, you also want to re-create the Jsmith user account and the Sales Users group in the Sales OU. When the user and group accounts are migrated to AD, they’ll have new SIDs.

To identify where the challenges are with respect to resource permissions, Figure 3.7 drills into the Sales domain and shows the resources that have permissions granted to them on the two servers being migrated.



Note that ServerA\SalesDataUsers is a local group on ServerA that contains the global group Master1\Sales Users (and other groups)



Note that Sales\SalesReportUsers is a domain local group in the Sales domain that contains the global group Master1\Sales Users (and other groups)

Figure 3.7: The resources on the two servers in the Sales NT 4.0 domain.

I've simplified the resources shown on each server in this figure to more easily describe the migration process, but the same concepts apply to servers with hundreds of files and folders. As you can see in the first part of this figure, there are two folders of interest on the member server called ServerA. The first folder is a Home directory for users in the domain. Under the Home folder is a jsmith folder, to which the user Master1\jsmith has Full Control NTFS permissions. There is also a data folder that contains a subfolder called Sales Data. The Sales Data folder has permissions granted on it to allow a local group, defined on ServerA, Full Control permissions.

In the second part of Figure 3.7, a domain controller in the Sales resource domain also contains some user data. Namely, it contains a data folder, which then contains a reports subfolder. The permissions on the reports folder grant Full Control to a domain local group, defined in the Sales

resource domain called SalesReportUsers. This group contains a number of global groups defined in the various master account domains in this NT 4.0 environment.

As you probably know, Microsoft has always recommended granting permissions to folders on NTFS volumes using local groups, which can contain multiple global groups. This simplifies granting permissions and prevents the Access Control List (ACL) on a given resource from growing too big and unwieldy. The worst-case scenario is assigning permissions to individual user accounts on a folder because every new user who needs access to the folder would have an Access Control Entry (ACE) defined in the ACL. So instead you can use local groups. This is all well and good except when you use *domain* local groups to define permissions.

When it comes time to migrate these resources to Win2K and AD, you have a problem. You want to move resources that currently exist in the Sales domain to a new AD domain called us.abccorp.com, but these resources exist on an NT 4.0 domain controller. You're not upgrading the Sales domain in place to Win2K, but you want to move its resources into the Sales OU. Unfortunately, those resources (in particular, the reports folder) have granted permissions to a domain local group in Sales that won't be available when those resources move into the new AD domain. And you can't migrate that domain local group because it belongs to the Sales domain, and the Sales domain won't exist in the new AD forest.

You can't upgrade an NT 4.0 domain controller to Win2K without either upgrading the whole NT 4.0 domain (starting with the primary domain controller, or PDC) or just upgrading it and not running the DCPromo utility, thus making it a member server. If it's a member server, it knows nothing about the Sales\SalesReportUsers local group, and thus users who had access to that folder by virtue of their membership in that group no longer do. So because you've done everything correctly, and according to past Microsoft recommendations, you've created a problem for yourself! What's the solution?

Unfortunately, there's no easy solution. The easiest thing to do is to grant permissions again to all resources that use domain local groups before you begin your migration; this allows access to a group that will exist in your new AD structure. I call this the easiest thing, but if this situation exists on hundreds of NT servers scattered around the world, you won't think it so easy.

The alternative is to use SID history to keep the NT 4.0 users and groups that you migrate to Win2K and AD with the same security context that they had before migration. You then need to keep the domain controllers housing the resources in the Sales NT 4.0 resource domain for some period of time until you're ready to move and grant permissions on those resources again. Again, it's a pay-now-or-pay-later trade-off. Figure 3.8 illustrates the problem and how SID history helps mitigate it.

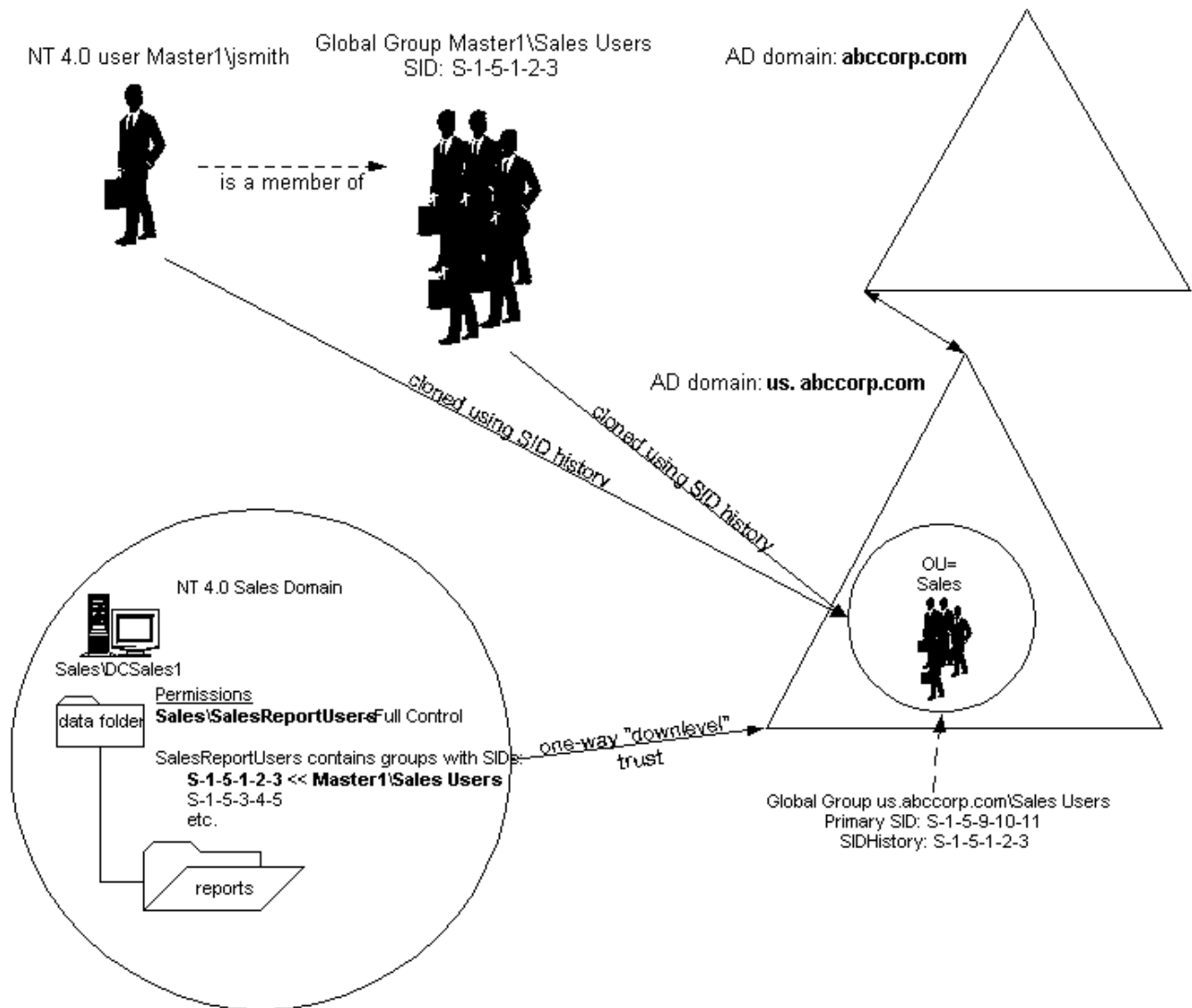


Figure 3.8: Using SID history migration to work around domain local group permissions on resources.

The figure above replicates a piece of Scenario A. A global group in the Master1 domain is called Master1\Sales Users and has a SID of S-1-5-1-2-3. This group is also a member of the domain local group in the Sales resource domain called Sales\SalesReportUsers. When you migrate user jsmith and the Master1\Sales Users group to the new AD domain us.abccorp.com, their SIDs will migrate with them. Thus, when jsmith logs on to the new AD domain, his membership in a new us.abccorp.com\Sales Users group will be maintained. Also, because this new group has a SID history containing its original Master1 domain SID, jsmith will still have access to the reports folder in the NT 4.0 Sales domain.

One last point that I haven't discussed so far is key to making this work: Sales must have a trusting relationship to us.abccorp.com. That is, to allow jsmith to access resources in Sales, you need to build an explicit "down-level" trust relationship between Sales and us.abccorp.com. Once you do this, everything will work fine.

The second part of Figure 3.7, in which a member server is being migrated from the Sales domain to the new Sales OU in us.abccorp.com, is a much easier case to handle. The resources for which permissions are granted on this server use either individual master account domain user accounts (for example, the home folder has permissions granted to Master1\jsmith) or member server local groups. Because a member server can simply be re-joined to a new domain, without losing its groups or SIDs, the ServerA\SalesDataUsers group remains in place after the server has been made a member of the AD domain; thus, the members of the group maintain all access to this resource.

In the case of the user's home folder, if you re-create a new user account in the new AD domain, you need to reassign permissions up front; if you use SID history, you need to reassign permissions after you're ready to retire the Master1 domain. Keep in mind that, while my examples apply to file resources, the same principles hold true for other types of resources, such as printers, user profiles, and so on.

In general, resource domain local groups represent the biggest problem for migrations. If you've granted permissions on these resources using these groups and you plan to retire the resource domain after you migrate to Win2K, you have to deal with permissions sooner or later. In the next sections, I'll present some techniques for examining these resources before you migrate to Win2K and AD. The key to ensuring that your users have access to all of their resources as you migrate them to AD is knowing what permissions are granted on all of your resources. For small shops, not knowing this before you migrate may not be a big deal. But if you're an NT 4.0 shop with hundreds of resource domains containing hundreds of servers, this can really be a nightmare.

Auditing Users and Groups

The best way to manage permissions on resources is to document all of the affected users and groups in your environment before you begin your migration planning. A number of third-party tools can help you audit your existing domain users and groups and, more important, any *group nesting* (that is, which global groups are part of which local groups). Some good NT Resource Kit utilities can also help with this auditing process. If you need to document a large environment, I recommend looking at the commercial products available for this purpose. The leading vendors in the NT migration and management space all have products that meet this need, including:

- NetIQ (www.netiq.com)
- BindView (www.bindview.com)
- FastLane Technologies (www.fastlane.com)

☞ Microsoft also provides a free utility called the Active Directory Migration Tool (ADMT). It's a stripped-down version of the commercial migration utility provided by NetIQ, and it lets you perform most basic migration tasks. You can find this tool at <http://www.microsoft.com/windows2000/downloads/deployment/admt/default.asp>.

Table 3.4 describes some of the more useful Resource Kit tools available to help document information about your current user and group environment.

This Tool	Does This
Findgrp.exe	<p>Contained in the NT 4.0 and Win2K Server Resource Kits and lets you ferret out a particular domain user's group membership from the command line. This tool returns a user's global and local group membership, even if the user is a member of a local group only by virtue of belonging to a nested global group. For example, if user jsmith is a member of the Sales Users global group and Sales Users is a member of the Sales local group, findgrp will list Sales under the local group. The syntax for this command is:</p> <pre>Findgrp <domain> <domain>\user</pre> <p>where <domain> is the domain you want to search and <domain>\user is the user account you want to search for. The Win2K version of this tool lets you search member servers in addition to domains. You can use findgrp.exe in conjunction with other tools that enumerate users in a domain to create reports of all of your user's group memberships. Then you can use this information to determine who'll be affected on which servers as you begin to migrate.</p>
Addusers.exe	<p>A command-line tool that lets you dump a list of users and groups on a given domain. It can be useful for generating a master list to feed tools like findgrp.exe above. The syntax for this command is:</p> <pre>Addusers \\server /d dumpfile.txt</pre> <p>where \\server is a domain controller or member server containing user and group accounts, /d is the option to dump information, and dumpfile.txt is the name of the dump file to create.</p>
Global.exe	<p>A command-line utility that lists all of the members of a global group on a domain or member server. It provides the opposite information to findgrp.exe in that it tells you who is in a given group, rather than what group a given user is in. The syntax for this command is:</p> <pre>Global "Domain Users" DomainA [or \\servera]</pre> <p>where "Domain Users" is the name of the global group you're searching on and DomainA (or \\servera) is the name of the domain or member server, respectively, that you're searching in.</p>
Local.exe	<p>Similar to global.exe, returns membership information for all local groups in a domain or member server. The syntax for the command is:</p> <pre>Local "Administrators" DomainA [or \\servera]</pre> <p>where "Administrators" is the local group name and DomainA (or \\servera) is the name of the domain or member server, respectively, that you're searching in.</p>

Table 3.4: Useful Resource Kit utilities for reporting on user and group information in your domains.

Each of the tools listed in Table 3.4 can help you document the user and group information on your NT network before migrating. However, the real challenge will come when you document the resources in your environment. Corporate networks typically contain far more and less consistent files, printers, and other resources than users or groups. Some networks may have hundreds or thousands of servers with thousands or tens of thousands of folders and files. If you didn't establish a regular process for granting permissions on these resources from the beginning,

you may have a lot of cleanup to do to ensure that users don't lose access to critical resources when you migrate them to Win2K.

Auditing File, Printer, and User Profile Permissions

Auditing and reporting file, printer, and other resources is the most critical task you need to carry out before you migrate to Win2K. The third-party companies I listed above are good resources for commercial products to help with this. Another tool that I've found useful in this area is Hyena, available at <http://www.systemtools.com/hyena/>. This utility, shown in Figure 3.9 below, is chiefly an administrative tool, but it also contains some pretty good reporting capabilities.

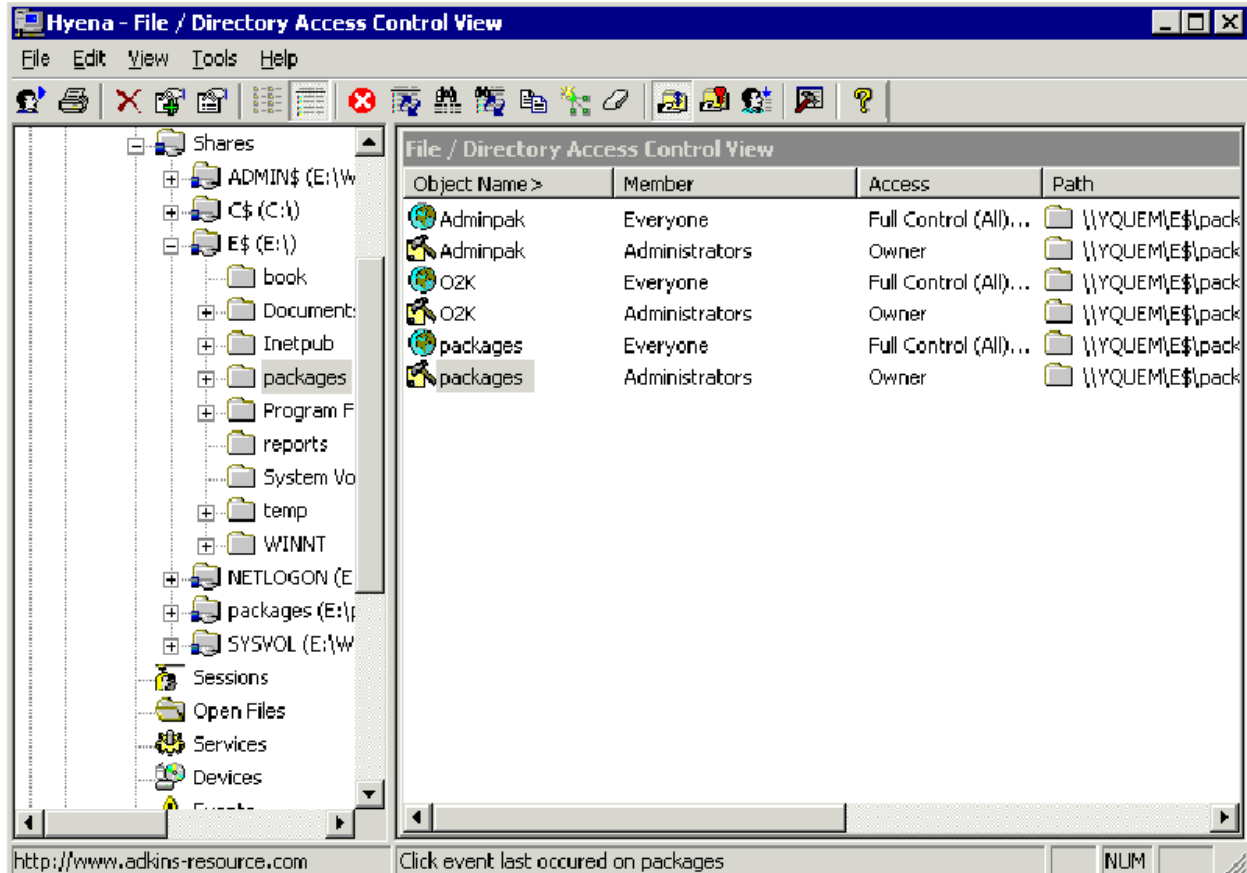


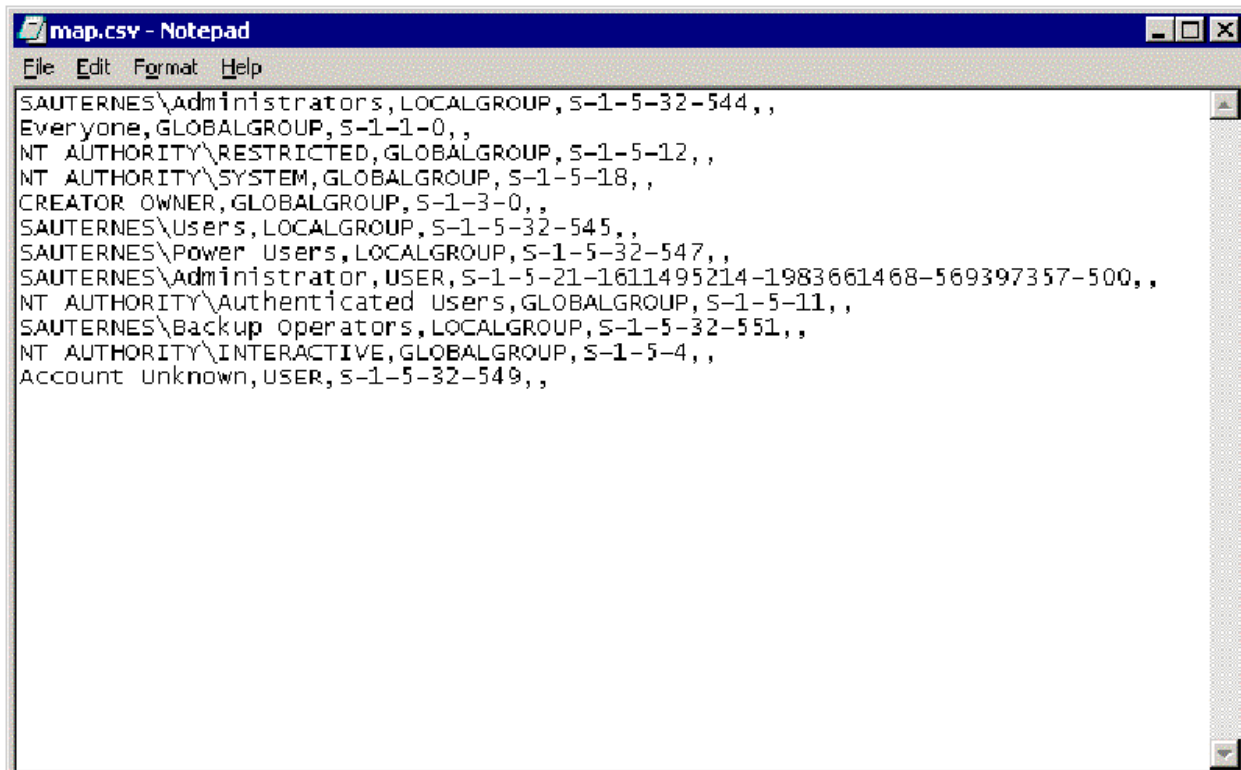
Figure 3.9: Using the Hyena administrative utility to examine file and folder security in your domain.

One of the best features about Hyena is that, using an Explorer-style interface, it provides a multi-domain view of your NT 4.0 network—something that the tools built into NT 4.0 (such as User Manager and Server Manager) can't.

A number of Resource Kit utilities can also come in handy when you're auditing resource permissions. The most useful ones that I've found are listed in Table 3.5 below.

This Tool	Does This
Showacls.exe	<p>A command-line utility available in the NT 4.0 and Win2K Resource Kits for reporting on NTFS file permissions on folders and files. You can also use it to search for ACEs for a particular user (but not group). The syntax for this command is:</p> <pre>Showacls /s <domain>\<user> <path></pre> <p>where the /s parameter searches all subdirectories under the path you provide, <domain>\<user> tells the tool to search for ACEs for a particular user account, and <path> is the file or folder path you want to search.</p>
Xcacls.exe	<p>An NT 4.0 and Win2K Resource Kit utility that lets you list and set NTFS file permissions on files and folders (although the NT 4.0 version doesn't seem to work on Win2K). It's similar to the built-in cacls.exe tool but provides more capabilities for setting permissions. This tool provides pretty basic functionality for reporting and editing ACLs, and it requires a lot of testing to get it right. The basic syntax for this command is:</p> <pre>Xcacls <path></pre> <p>This tool also has a number of other command-line options for editing permissions.</p>
Showacccs.exe	<p>A Win2K Support Tools utility that takes what showacls.exe does to the next level. It's a powerful tool for reporting on file, printer, and Registry security. The tool also works hand-in-hand with another utility, sidewalk.exe, to let you change resource permissions for migrating to Win2K. The syntax for this command is:</p> <pre>Showacccs <output file> /f <path to search> /m <map file name></pre> <p>where <output file> is the name of the file that contains the information about the security of the resource you're querying, /f <path to search> indicates that you want to search an NTFS file path, and /m <map file name> generates a mapping file that the sidewalk.exe utility can use as input to modify permissions on a resource at a later time. (Figure 3.10 below shows a mapping file.)</p>

Table 3.5: Utilities for reporting on resource permissions.



```
SAUTERNES\Administrators,LOCALGROUP,S-1-5-32-544,,
Everyone,GLOBALGROUP,S-1-1-0,,
NT AUTHORITY\RESTRICTED,GLOBALGROUP,S-1-5-12,,
NT AUTHORITY\SYSTEM,GLOBALGROUP,S-1-5-18,,
CREATOR OWNER,GLOBALGROUP,S-1-3-0,,
SAUTERNES\Users,LOCALGROUP,S-1-5-32-545,,
SAUTERNES\Power Users,LOCALGROUP,S-1-5-32-547,,
SAUTERNES\Administrator,USER,S-1-5-21-1611495214-1983661468-569397357-500,,
NT AUTHORITY\Authenticated Users,GLOBALGROUP,S-1-5-11,,
SAUTERNES\Backup Operators,LOCALGROUP,S-1-5-32-551,,
NT AUTHORITY\INTERACTIVE,GLOBALGROUP,S-1-5-4,,
Account unknown,USER,S-1-5-32-549,,
```

Figure 3.10: A mapping file generated by the *showaccs.exe* utility.

Of course, once you've reported on permissions for all the resources on your network, you need to modify resource permissions to support your migration. Again, the approach you take depends on your migration method (for example, in-place, create and re-ACL up front, or create and use SID history), but the issues are all the same.

Modifying Resource Permissions

The typical approach to managing migration of file, folder, and printer resources is that once you've audited their permissions, you need to change the permissions to support your migration. Another approach is to clean up the NT 4.0 permissions before you begin the migration. For example, as I indicated in "Handling Permissions Issues" earlier in this chapter, resource domain local groups can cause problems if you assign them to resources in domains that won't be migrated to your new AD infrastructure. You might decide that you want to handle permissions up front by changing them on these resources while you're still in your NT 4.0 environment.

Similarly, if you haven't taken a consistent approach to granting permissions on resources up until now—for example, you've granted permissions on some resources using local groups, some using global groups, and perhaps some to individual user accounts—you might decide that making your permission scheme consistent before even considering migration is the way to go.

In all of these cases, you need a way to change large numbers of resource ACLs on potentially many different servers in many different domains. Again, if you have a large NT environment, your best bet is probably to purchase one of the commercial migration packages I've mentioned

(see “Auditing Users and Groups” earlier in this chapter). Most of these tools are robust utilities for changing resource permissions across many servers without severely affecting server and network performance.

If you only have to consider a small number of servers, some inexpensive utilities are available that may fit the bill. The one that comes to mind first is SIDWalker (part of the Win2K Support Tools), which I referred to in Table 3.5. This tool actually has two parts. The first, `sidwalk.exe`, is a command-line utility that lets you script a re-ACLing operation. The second is Security Administration Tools, available from the Support Tools menu. It’s a Microsoft Management Console (MMC) snap-in that lets you create mapping files similar to those generated using the `showaccs.exe` utility. Sidwalk.exe then uses these mapping files to drive the re-ACLing operation. Figure 3.11 shows the Security Administration Tools MMC snap-in.

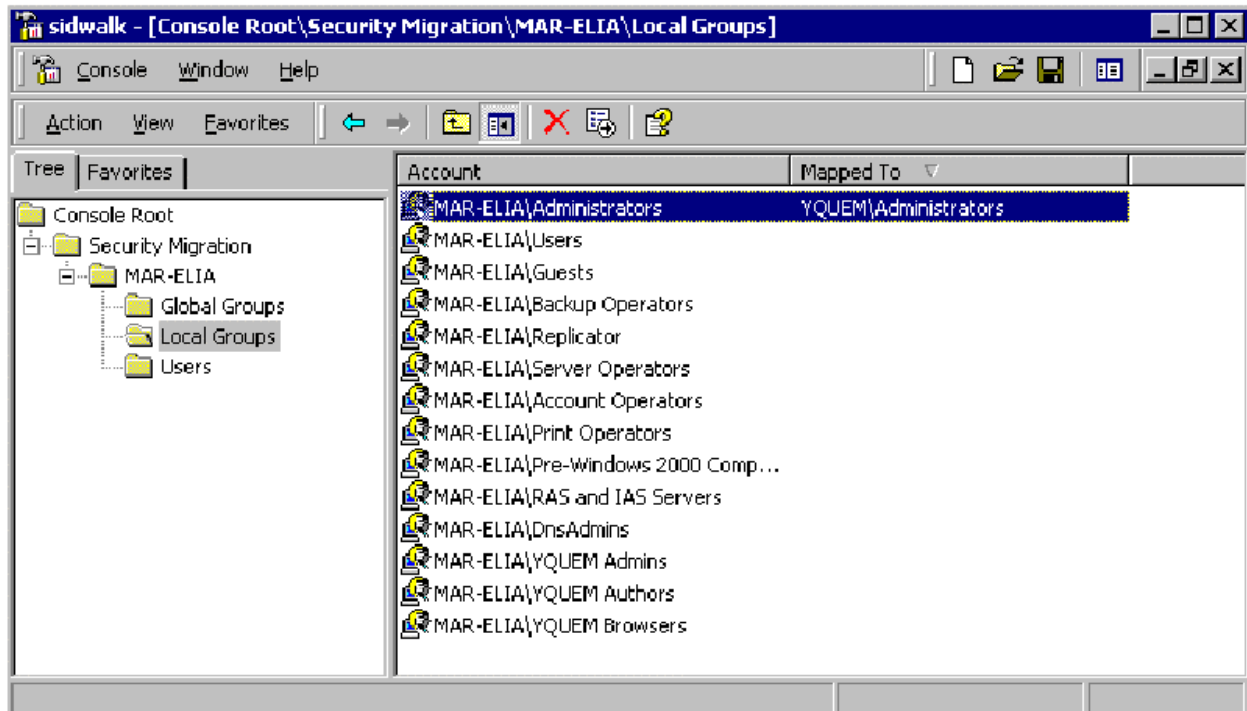


Figure 3.11: The Security Administration Tools MMC snap-in.


As you can see from this figure, SIDWalker lets you map global groups, local groups, and users from one domain to another. Once you’ve done this, you right-click the domain name and select Save As from the shortcut menu. This creates a comma-separated values (.csv) file that `sidwalk.exe` uses to modify permissions on resources, changing permissions from the source user or group to the target user or group.

Unfortunately, `sidwalk.exe` doesn’t edit the ACL; it changes it. (Most of the third-party tools that perform this kind of operation give you the choice of adding new ACLs rather than editing existing ones. This has obviously less impact and allows you to back out.) So you need to be sure that the change you’re making is the one you want! Sidwalk.exe does give you a chance to take a dry run, using the `/t` option, before making the actual change, and I suggest you take advantage of this feature before running the tool for real.

Another point to keep in mind is that running `sidwalk.exe` across your network on multiple servers with hundreds or thousands of files will have a big impact on both server and network performance. I suggest doing any re-ACLing operation after hours and running the tool locally on each server (rather than across the network using mapped drives) to minimize system disruption.

Finally, a word about changing ACLs on user profiles: You grant permissions on a user profile by setting security on the Registry hive that makes up the `HKEY_CURRENT_USER` portion of the profile. This hive file is called `ntuser.dat` in NT 4.0 and Win2K. When a user is logged in, this hive file is loaded into the Registry and is available for permission changes. However, this implies that the user is logged on when you're changing profile permissions, and this may not always be the case.


Alternatively, you can manually load the `ntuser.dat` hive file into the Registry using a tool like `reg.exe` from the Resource Kit, then perform security changes on it, but this requires a lot of coordination. I recommend that before you migrate to Win2K, you edit security on a user profile as part of a logon script. For example, in the days leading up to the migration, insert a command into your regular logon script that adds a new ACE for the user profile; this ACE gives access to the new user account in the new AD domain. A tool like `regini.exe` from the Resource Kit allows such a permission change.

 Profile migration tools will be discussed in Chapter 4. See the section "Deployment Tools and Techniques"—"User Profile Migration."

Updating the NT Operating System

Once you've prepared your resources for migration, the next step is to take a look at the OS revision level that runs across your NT 4.0 infrastructure. It goes without saying that, as you migrate to Win2K, having all servers at the same revision level minimizes any surprises and ensures that you have a level playing field for troubleshooting problems that may arise. This advice is, of course, most relevant if you plan to upgrade servers in place rather than create a new AD infrastructure from scratch.

I suspect that very few organizations have the resources to build an entirely new infrastructure beside their NT 4.0 one, so it behooves you to keep things as consistent as possible. I recommend that all servers, and especially domain controllers, be running a minimum of NT 4.0 Service Pack 5. Allow enough time during your migration to upgrade any servers that aren't at this minimum level, and make sure you test before upgrading to ensure that newer service packs don't break applications that you've installed on these servers.

 While Microsoft does support upgrading servers from NT 3.51 Service Pack 5 to Win2K, consider avoiding doing this if you still have 3.51 servers. It might be a good time to retire those boxes or upgrade them to NT 4.0 before migrating.

You can use a tool like Hyena (described earlier in “Auditing File, Printer, and User Profile Permissions”) to examine service-pack levels on your servers and make sure that they’re the most current. Whether you bring all of your NT 4.0 workstations to the same level depends on whether you plan to upgrade them to Win2K Professional or simply “wipe and load” them. In the latter case, it’s probably not that important what revision they’re currently running.

Another, optional, suggestion is, before you upgrade to Win2K, to install the Security Configuration Manager tools, also known as Secedit, that were first shipped as part of Service Pack 4 for NT 4.0. Secedit changes the ACL editor tool that NT 4.0 provides to the Win2K-style ACL editor (shown in Figure 3.12 below).

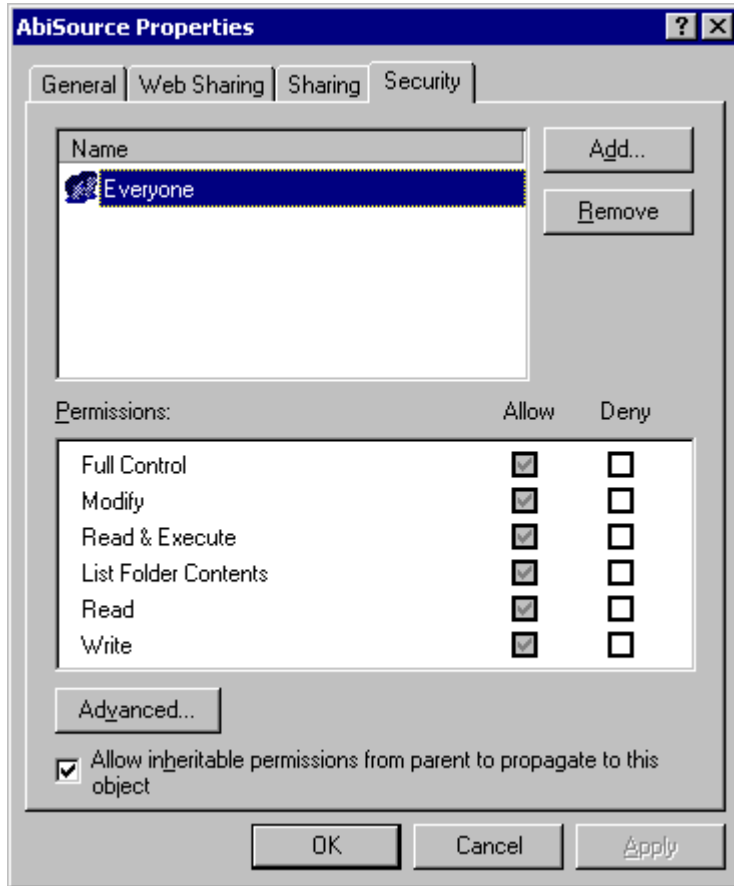



Figure 3.12: The Win2K-style ACL editor installed as part of the Security Configuration Manager tool set.

After you get used to this new-style ACL editor, it can really help you troubleshoot Win2K permissions once you’ve begun your migration.

 There are some known incompatibilities between the new-style ACL editor installed by the Security Configuration Manager and the one provided by NT 4.0. For more information on this, check out <http://support.microsoft.com/support/kb/articles/Q195/5/09.ASP>.

Restructuring Domains

Another issue that you may need to consider is whether to restructure domains—either your existing NT 4.0 domain infrastructure before migrating to Win2K or the resulting AD domain once you’ve migrated.

NT Environments before Migration

In a few circumstances, it may be prudent to restructure your NT 4.0 domain infrastructure before migrating to Win2K. In general, however, I recommend avoiding it. Migrating to Win2K requires a lot of work, and unless your NT 4.0 environment is in such a mess that you can’t possibly hope to be successful, you’re better off putting your energies into the migration.

Here are some of the reasons you might consider restructuring your existing NT 4.0 environment before migrating:

- You plan to perform an in-place upgrade to Win2K, and you have a number of NT 4.0 domains that you don’t want to upgrade. In this case, you might combine several NT 4.0 domains into fewer than exist today.
- You can’t afford to build a new, pristine AD environment to move your existing NT 4.0 infrastructure into, and if you reorganize your NT 4.0 environment, you might be able to recover hardware for the migration.
- Your NT 4.0 environment has had no consistency or standards in place, resulting in many domains, inconsistent permissions on resources, and no clear owners for the domains. You might want to clean things up before beginning a migration to ensure that the migration is successful.
- Your company has experienced a reorganization, merger, or acquisition, and you know that your current NT 4.0 infrastructure isn’t going to represent what you want to deploy in Win2K. In this case, you might want to start with an NT 4.0 environment that more easily maps to your planned Win2K environment.
- You plan to have your NT 4.0 and AD domains co-exist for quite some time, and to make life easier, you don’t want to have an NT 4.0 mess to manage while moving to Win2K.

All of these are valid reasons to restructure your NT 4.0 environment before migrating to Win2K. Some of them are mitigated by the tools that are available for migration—such as SID history cloning—which allow you to move your NT 4.0 environment into a better Win2K structure while leaving the NT 4.0 baggage behind. However, even if you decide not to reorganize your NT 4.0 environment before migrating, there may be good reasons to do it afterward.

AD Domains after Migration

You most often restructure AD domains when you upgrade your NT 4.0 infrastructure in place and, after migrating your NT 4.0 domains to Win2K, you move users and computers into fewer AD domains. Figures 3.13 and 3.14 show such a scenario.

Migration Step 1

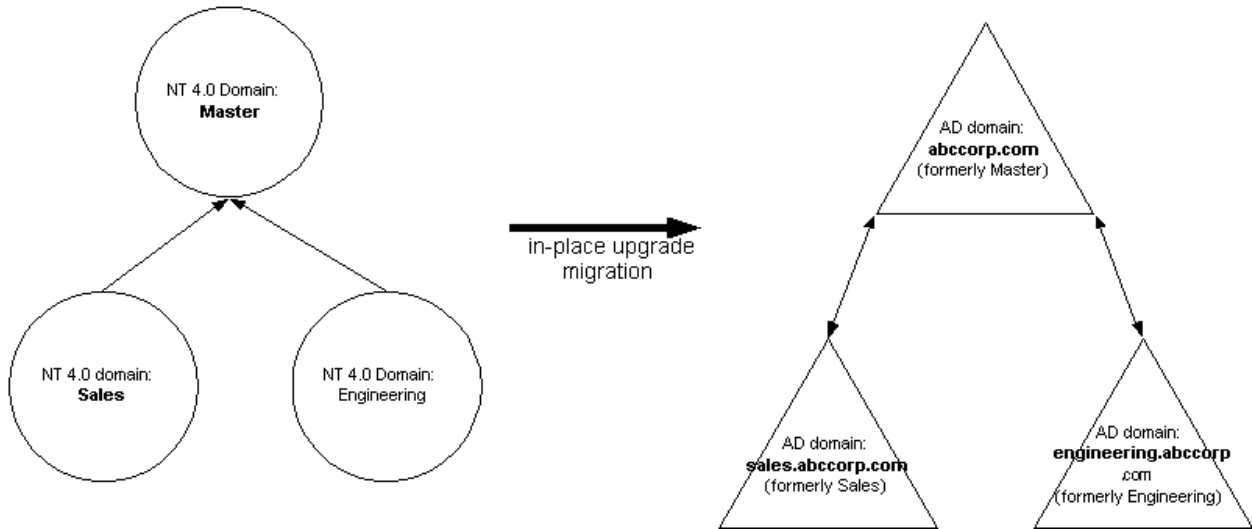


Figure 3.13: Step 1 of an in-place upgrade of an NT 4.0 infrastructure to Win2K. Domains have been upgraded in place to a multi-domain AD forest.

Migration Step 2

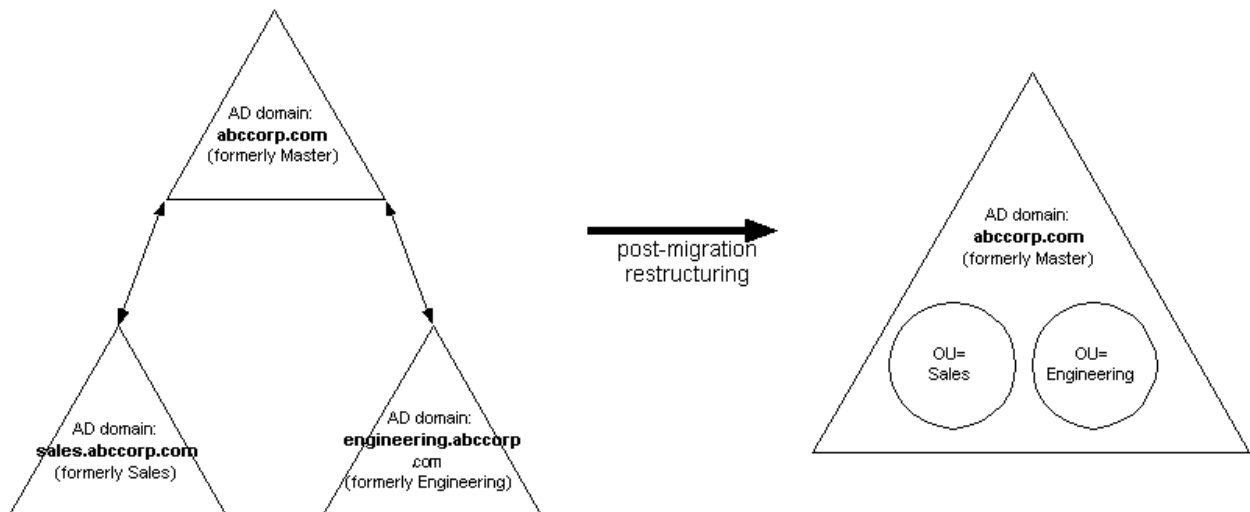


Figure 3.14: Step 2 of an in-place upgrade. Once in the AD domain, old NT 4.0 resource domains are collapsed into a single AD domain, using OUs in place of resource domains.

In step 1, shown in Figure 3.13, the three NT 4.0 domains are upgraded in place. This creates three AD domains in the new AD forest. The domain previously known as Master becomes the new forest root domain called abccorp.com. The old resource domains are child domains to abccorp.com called sales.abccorp.com and engineering.abccorp.com. All of the user accounts

that were in Master are now in abccorp.com. All of the resources—servers and workstations—that were in Sales and Engineering are part of the child domains.

In step 2, shown in Figure 3.14, all of the resources have been moved into the new AD forest, and now it's time to restructure to achieve the final AD namespace. Namely, the resources that are in sales.abccorp.com and engineering.abccorp.com are moved into new OUs in abccorp.com. The two child domains can then be decommissioned, and the result is a single AD domain for all of the resources.



See the discussion in Chapter 4 on the idea of using a protected forest root domain before deciding whether to create a single AD domain.

Keep in mind that when you upgrade the Master domain in place, all of the user and group accounts are moved into an AD container called Users. As part of step 2 restructuring, you'll probably move these user accounts into one of the two OUs that you create, depending on which division the user is a member of.

Two Choices, Similar Issues

Whether you restructure your NT 4.0 domain environment before migrating to Win2K or restructure AD domains afterward, the issues are similar. That is, you need to ensure that if you're creating new security principals (such as users and user groups), they can access their resources after restructuring. For example, if you're combining two NT 4.0 master account domains into one, you need to re-create the users and groups from one of the master account domains in the remaining one. Because the new users and groups will have new SIDs, you need to reapply permissions on all of their resources.

The issues are also similar when you move objects from one AD domain to another. The SID for the user will change, but because Win2K supports SID history, you don't necessarily have to re-ACL the resource when you move a user or group.

Domain-Restructuring Tools

A number of tools are available to help you restructure domains—both before and after migrating to Win2K. Some are from Microsoft, others are from third parties. Keep in mind that the larger and more complex your environment is, the more likely you'll need to purchase a commercial migration tool that contains many sophisticated features. However, a few tools from Microsoft are either free or part of the Resource Kit.

From Microsoft

Microsoft provides a number of tools in the Resource Kit, in the Win2K Support Tools, and on its Web site that help with domain restructuring.

Active Directory Migration Tool (ADMT)

Microsoft's primary domain-restructuring tool is the Active Directory Migration Tool (ADMT), which I introduced earlier in this chapter (see "Auditing Users and Groups").

 For a full discussion of ADMT, see Chapter 4 of this book.

ADMT is supplied as an MMC snap-in, and it provides reporting and migration capabilities for a variety of migration scenarios. Figure 3.15 shows the features of the tool.

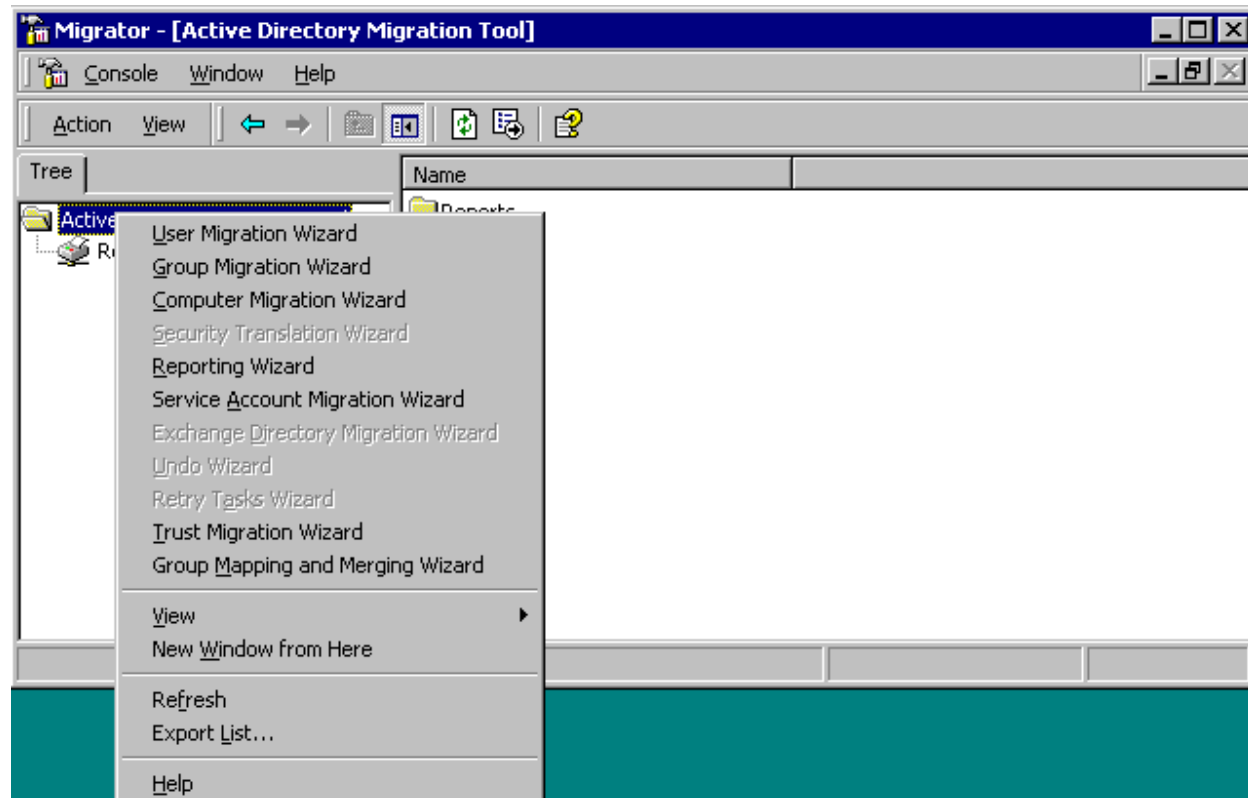


Figure 3.15: The features of ADMT.

As you can see from the figure above, ADMT provides migration capabilities for a variety of tasks, including user and group migration, computer account migration, service accounts, and even trust relationships. ADMT also provides a back-out facility in case you need to back out of or undo a migration after you start it.

Netdom

Another tool to help with your restructuring tasks is the command-line Netdom.exe utility in the Win2K Support Tools. You can use this tool to move workstation and member server machine accounts from one domain to another (either NT 4.0 or Win2K). Netdom basically re-creates a machine's account in the new domain without requiring you to manually leave one domain and join the other in the Network Control Panel applet. The syntax for netdom.exe is:

```
Netdom move <machinename> /domain:<dest domain> /userd:<dest user> /password: <dest password> /reboot: <sec>
```


where:

- <machinename> is the name of the member server or workstation that you're moving
- <dest domain> is the name of the domain that you're moving the machine to
- <dest user> is an account in the destination domain that has privileges to create machine accounts
- <dest password> is the password for dest user
- <sec> specifies how many seconds to wait before the machine is restarted (and for the domain change to take effect).

In addition, to place the machine account in a specific OU, you can specify a destination OU. The following is an example of using netdom to move a member server called AppServer1:

```
Netdom move appserver1 /domain:abccorp.com /OU:
OU=finance,DC=abccorp,DC=com /userd:administrator /password:*
/reboot:10
```

In this example, I'm moving the Appserver1 server to the Finance OU (specified by the distinguished name OU=finance,DC=abccorp,DC=com) in the abccorp.com domain. The administrator account is for the abccorp.com domain, and the asterisk after the *password* option tells netdom to prompt me for a password when the command runs. I'm also waiting 10 seconds before restarting the server.

 Netdom *doesn't* work on domain controllers, so don't even try!

MoveTree

Another useful tool is movetree.exe. MoveTree is another Win2K Support Tools utility that lets you move whole OUs or individual users and groups among domains in a Win2K AD forest. Thus, movetree.exe assists in post-migration restructuring. MoveTree doesn't support moving machine accounts or domain global or local groups, but you can use netdom.exe for the former if needed and other migration tools like ADMT for the latter. Figure 3.16 shows the syntax for movetree.exe.

```

Command Shell
D:\Program Files\Support Tools>movetree
THE SYNTAX OF THIS COMMAND IS:
MoveTree [/start | /continue | /check] [/s SrcDSA] [/d DstDSA]
        [/sdn SrcDN] [/ddn DstDN] [/u Domain\Username] [/p Password] [/verbose]

/start      : Start a move tree operation with /check option by default.
            : Instead, you could be able to use /startnocheck to start a mov
e
/continue   : tree operation without any check.
/continue   : Continue a failed move tree operation.
/check      : Check the whole tree before actually move any object.
/s <SrcDSA> : Source server's fully qualified primary DNS name. Required
/d <DstDSA> : Destination server's fully qualified primary DNS name. Require
d
/sdn <SrcDN> : Source sub-tree's root DN.
            : Required in Start and Check case. Optional in Continue case
/ddn <DstDN> : Destination sub-tree's root DN. RDN plus Destinaton Parent DN.
Required
/u <Domain\UserName> : Domain Name and User Account Name. Optional
/p <Password> : Password. Optional
/verbose    : Verbose Mode. Pipe anything onto screen. Optional

```

Figure 3.16: The syntax for the movetree.exe utility.

The figure above shows that while the syntax for movetree.exe is fairly complex, the command is powerful. Let's go back to Figure 3.14 and suppose that you want to move users and groups in the sales.abccorp.com domain to an OU in the abccorp.com called Sales. This was part of your post-migration restructuring operation. Suppose too that in sales.abccorp.com, after upgrading from NT 4.0, you moved all of the users that you planned to restructure into an OU called transfer. The following shows the format of a movetree command to complete this operation:

```

Movetree /start /s salesDC1.sales.abccorp.com /d
ABCDC1.abccorp.com /sdn OU=transfer,DC=sales,DC=abccorp,DC=com
/ddn OU=sales,DC=abccorp,DC=com /u abccorp.com\administrator /p
adminpassword /verbose

```

In this command:


- start begins the move process and also checks the validity of the move by default before actually moving anything
- /s specifies a source domain controller
- /d specifies a destination domain controller
- /sdn lists the distinguished name of the OU that you're transferring from
- /ddn lists the distinguished name of the OU that you're transferring to
- /u is the optional (assuming that you're not already authenticated to the domain) credential that you want to pass to the destination domain to create the new OU
- /p is the password for that account
- /verbose sends verbose output about the status of the command to the screen.

If the `movetree.exe` utility encounters objects that can't be moved in the source container, it simply ignores them. If you want, you can use `MoveTree` to move individual users and groups rather than whole OUs.

`MoveTree` is a fairly powerful utility for restructuring AD domains and collapsing Win2K domains, but it's a command-line utility, and it's up to you to deal with logging and backing out. That's where the commercial packages come in handy.

From Third Parties

There is a big market for Win2K- and AD-related domain-migration tools. Most of them provide features to help you clean up and restructure your NT 4.0 environment in preparing to migrate. As I mentioned earlier in this chapter (see "Auditing Users and Groups"), there are several big names in this space, with quite a few smaller companies trying to provide solutions as well. Rather than give a detailed review of every product, I'll suggest features that you can look for. Hopefully, the same tool or suite of tools that you use to prepare your NT 4.0 network will also meet your AD-migration needs.

 Chapter 4 describes the migration process in more detail and discusses the challenges you're likely to face.

When you shop for a domain-restructuring tool, look for one that offers the following features:

- The ability to create migration "projects" or scenarios that can be repeated or replayed as needed.
- A reporting and inventory capability that lets you take stock of your NT 4.0 environment and use that information to help you migrate to Win2K.
- Some kind of rollback capability, where you can undo a migration scenario if it's not what you intended.
- Re-ACLing all major resources, including files, shares, printers, service accounts, user profiles, and Microsoft Exchange mailboxes.
- Distributing the re-ACLing process to the server where the operation is happening to minimize the network impact of re-ACLing and provide a more scalable solution.
- Migration and restructuring of NT 4.0 domains as well as AD mixed-mode and native-mode domains. This means support for migrations from NT 4.0 to AD as well as restructuring within or across AD forests and in or across NT 4.0 domains.
- Comprehensive reporting on the status of a migration, including errors or failures.

These are features that you'll likely need for any medium- to large-scale NT 4.0 environment. Most of the vendors I've mentioned in this chapter provide some, if not all, of these features. Some are stronger in the restructuring area than migration and vice-versa. Depending on your migration approach (for example, in-place upgrade, re-create and re-ACL, or SID history), you'll likely find a tool that fits your needs.

Summary

In this chapter, I reviewed the work involved in migrating from WinNT 4.0 to Win2K and the initial clean-up required to ensure that you're successful. I also introduced another important aspect of your migration: AD design. In the next chapter, I'll look at the actual migration process, paying particular attention to the tools to expedite the process, such as ADMT from Microsoft, and alternatives from third-party vendors that, depending on your situation, can make the process more manageable.

Copyright statement

This site contains materials created, developed, or commissioned by Realtimerepublishers.com, Inc. and is protected by international copyright and trademark laws. No material (including but not limited to the text, images, audio, and/or video) may be copied, reproduced, republished, uploaded, posted, transmitted, or distributed in any way, except that one copy may be downloaded for your personal, non-commercial use on a single computer. In connection with such use, you may not modify or obscure any copyright or other proprietary notice.

If you have any questions about these terms, or if you would like information about licensing materials from Realtimerepublishers.com, please contact us via e-mail at info@realtimerepublishers.com