

# Active Directory

## How Does It Measure Up?

Gary Hein

After years of delay, Microsoft's Active Directory has finally arrived as an inseparable part of Windows 2000. You've undoubtedly read the hype from Microsoft, the trade press, and industry experts. Given your busy schedule and limited budget, however, you may not have had time to actually install Active Directory in a test lab and try it out. You may still be wondering exactly what Active Directory is and how it will affect your job, your company, and your customers. To help you become more familiar with Active Directory, this article outlines the following:

- The history of Active Directory, which, in turn, explains its architecture and many of its features
- The major pitfalls and "gotchas" of deploying and managing Active Directory
- Solutions for integrating and managing a mixed Active Directory and NDS eDirectory environment

Because you are an NDS administrator, this article uses NDS eDirectory as a reference point, comparing how Active Directory works with how NDS eDirectory works.

### A HISTORY OF THE DOMAIN

Over the years, Microsoft has continued to improve the Windows NT platform, enhancing its stability (Windows NT 3.51) and adding the Windows 95 GUI (Windows NT 4.0). Despite these enhancements, however, Windows NT has suffered from limitations in the user management system. After all, the Windows NT domain was built on the legacy LAN Manager architecture, circa 1980s. As a result, Windows NT 4.0 and 3.51 have limitations in the following areas:

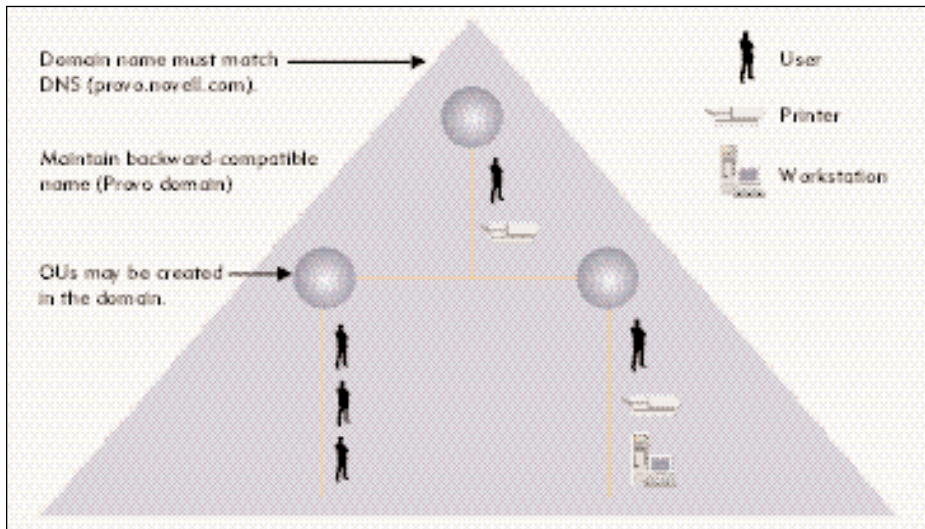
- **Scalability.** In Windows NT 4.0, the domain is stored in the Windows registry. Although these domains are theoretically limited to 40,000 objects (users, groups, and workstations), most practical deployments cannot exceed 10,000 objects before becoming extremely cumbersome and essentially unsupported. Simple management tasks, such as changing a user's password, may require sorting through thousands of objects. Further complicating matters, the registry is always cached in the server's memory, so large domains require a lot of server RAM.
- **Replication and Fault Tolerance.** Windows NT 4.0 domain replication is built on a master-slave replication model: Domain changes (such as adding a user, creating a group, and changing a



password) occur at the Primary Domain Controller (PDC), which then sends out updates to all Backup Domain Controllers (BDCs). Unfortunately, if the PDC is down or unreachable due to a LAN or WAN problem, the domain is unmanageable. The PDC is not only a single point of failure in terms of management but is also the sole source of domain information, creating inefficient replication in large companies that may include hundreds of BDCs.

- **Limited Object Types/No Extensibility.** Conceived before cellular phones, pagers, and the Internet, the Windows NT 4.0 domain structure can't represent objects found in today's IT infrastructure. Windows NT 4.0 domains are limited to users, groups, and computers, with no capability to extend the domain for new types of objects (such as web servers and routers) or new attributes (such as pagers, cellular phones, and web site home pages).
- **No Hierarchy.** Users, groups, and computers are equals within the domain, even if they exist in separate business divisions or departments. Simply stated, due to its flat nature, the Windows NT 4.0 domain can't represent a company's hierarchy or organizational chart.
- **Trust Relationships.** Connecting multiple Windows NT 4.0 domains together requires trust relationships, which grow exponentially as domains are added. Trust relationships also increase management overhead and complicate domain-planning issues.
- **Delegation of Authority.** Administrative rights within the Windows NT 4.0 domain system are all-or-nothing. It is impossible to delegate administrative rights to a subset of objects, such as giving a department manager rights to manage users within their department.

Of course, these problems are not unique to Windows NT 4.0 domains. NetWare 3 and 2, which are based on a flat bindery, have many of these same limitations. For large enterprise networks and Internet solutions, neither the domain nor the bindery provide the necessary scalability. All companies, large and small, also suffer from the inability to extend the domain or bindery to include emerging technologies. Due to these limitations, Microsoft realized (as Novell had years before) that a flat domain structure couldn't meet the needs of future networking technologies.



**Figure 1.** You can use Active Directory OUs to represent your business hierarchy in the domain structure, but you cannot use these OUs to set up domain security.

With a well-known list of Windows NT 4.0 domain limitations and problems, Microsoft set out to remedy these issues with Active Directory, much as Novell had done when designing NDS to solve NetWare 3 and 2 bindery limitations. However, Microsoft and Novell took different approaches to solve their respective problems.

### Novell's Approach

When Novell set out to solve bindery limitations, it initially tried to fix the bindery by increasing its scalability, adding replication, simplifying management, and so forth. Novell's "super bindery," known as NetWare Name Services (NNS), was a flop because it was based on legacy bindery technology. Novell quickly realized that fixing limitations wasn't enough—the underlying bindery architecture was flawed. As a result, Novell abandoned the bindery and used existing standards to design a scalable, extensible directory—NDS.

### Microsoft's Approach

Rather than starting from scratch, Microsoft chose to enhance its existing domain technology and market the enhanced domain service as Active Directory. In fact, Active Directory is a bit of a misnomer: It's really just Windows NT 4.0 domains with "fixes" to address previous domain limitations. Active Directory includes many improvements to Windows NT 4.0 domains, such as the following:

- **Scalability.** The Active Directory database is no longer stored in the registry

but in a database similar to the database used by Microsoft Exchange. Microsoft claims that this new database supports up to 10 million objects per domain.

- **Replication and Fault Tolerance.** Active Directory does not have a PDC/BDC hierarchy and, therefore, has no single point of failure within the domain management system. Active Directory also supports multimaster replication, whereby domain changes may occur on any domain controller. These changes are then automatically replicated to all other domain controllers.
- **Extensible Schema.** The Active Directory schema supports many new types of objects and attributes and allows developers to define their own application- or product-specific domain objects.
- **Hierarchy.** Domain Organizational Units (OUs) allow companies to represent their business hierarchy within the domain structure.
- **Improved Trust Relationships.** Trust relationships are now both automatic and transitive in Active Directory, decreasing management overhead and domain-planning issues.
- **Delegation of Authority.** Active Directory enables you to delegate administrative rights. For example, you can give a department manager administrative rights over users within his or her department.

Active Directory is obviously an improvement on the previous Windows NT 4.0 domain. However, because Active Directory is based on legacy domains, it

continues to suffer from domain limitations—including limitations in security, delegation of rights, partitioning and replication, and deployment.

### LEARNING THE TERMS

To understand Active Directory's strengths and weaknesses, you need to know some basic terms and concepts. Because Active Directory is based on Windows NT 4.0 domains, you can draw on a knowledge of Windows NT 4.0 to understand these terms and concepts. For example, the domain is the basic unit of Active Directory. (See Figure 1.) The domain is the security, administrative, and replication boundary. Security policies, such as minimum password length or password change frequency, are set on a per-domain basis and apply to all objects in the domain.

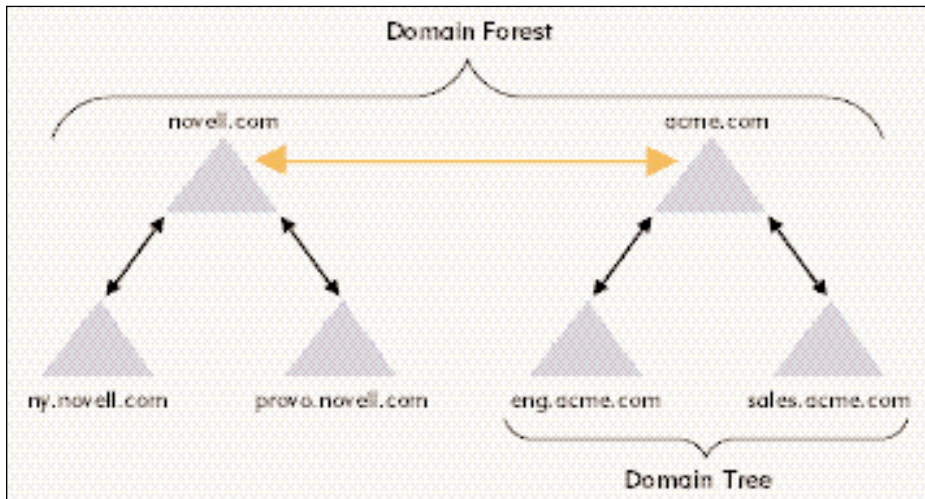
Because the domain is also the administrative boundary, all administrative tasks or delegated administrative rights are valid only for objects within the domain. As the replication boundary, the domain is the smallest unit of replication, which means that all objects within the domain are replicated to all other domain controllers. Unlike NDS, Active Directory cannot replicate just a portion of a domain's objects, such as an OU that represents a small field sales office. As a result, either all domain objects are replicated to domain controllers in every site, or a multiple-domain design is required.

Hierarchy is established within the domain by using OUs. (See Figure 1.) OUs simplify management by allowing you to group domain objects, such as users, groups, and computers, into a hierarchical structure.

You can set up trust relationships to link domains together to form a domain tree. (See Figure 2 on p. 24.) Within a domain tree, all domains must be part of the same Domain Naming System (DNS) hierarchy.

You can also use trust relationships to link domain trees together to form a domain forest. (See Figure 2 on p. 24.) A domain forest links separate DNS domain trees together.

By using OUs, domains, domain trees, and domain forests, you can represent the logical structure of a company within Active Directory. In addition, Active Directory provides domain sites, which are a group of IP subnets, to represent the physical location of objects and services.



**Figure 2.** You can use trust relationships to link domain trees to form a domain forest.

(See Figure 3 on p. 28.) You can also use domain sites to establish domain replication policies.

Because Active Directory can be distributed across multiple domains and servers, it is unlikely that any single Active Directory server will contain all of the objects for an entire domain forest. Although distributing Active Directory across multiple servers improves scalability, it limits the performance of directory-wide searches. A single search may require contacting multiple directory servers, possibly across slow WAN links.

To address this problem, Active Directory uses the global catalog, which is a prebuilt index of directory information stored on selected servers. To speed up directory searches, the global catalog does not include all directory information. Instead, the global catalog contains selected directory attributes, such as e-mail addresses, phone numbers, and names.

Global catalog servers are an integral part of Active Directory. The global catalog is required for user authentication, access control calculations, and directory

management. As such, planning global catalog servers is an important aspect of any Active Directory design.

#### DIRECTORY MANAGEMENT 101

The heart of any directory is the ability to manage rights and relationships between directory objects. What's the point of putting all of your company's information in a directory if you can't leverage it to simplify network management?

In general, directories use security principals and access control lists (ACLs) to manage relationships and to grant or deny access to services. A security principal is any directory object that can be given management rights or access rights to other directory objects.

For example, NDS security principals include users, groups, workstations, OUs, organizational roles, printers—actually, everything! Any NDS object may be granted rights to any other NDS object or corporate resource, even a developer-defined object.

Active Directory, like Windows NT 4.0 domains, has a limited set of security

principals—users, groups, and workstations. Other Active Directory objects, such as OUs or developer-defined objects cannot be security principals. As a result, all management rights within Active Directory must be granted or denied to users, groups, or workstations.

If you are like most NDS administrators, you are using the NDS OU to simplify your life by granting directory and resource rights to OUs. For example, you could grant the OU=Engineers rights to access the cellular phone numbers of other engineers. Or, you could grant the OU=Engineers rights to access engineering resources, such as servers, printers, and web sites (all examples of resource rights).

In other words, rather than granting rights to individual users or groups, you can associate rights within the NDS hierarchy. Then when a user is added to or deleted from the OU=Engineers, that user automatically gains or loses these rights.

Because Active Directory security principals are limited to users, groups, and workstations, Active Directory cannot leverage the domain OU hierarchy to reduce administrative overhead. All rights must be granted directly either to users or to groups of users.

An interesting story lies behind this limitation in Active Directory. Microsoft claims that using NDS OUs as security principals is an undesirable feature that somehow increases NDS administration and complexity when you move users. (See [www.microsoft.com/windows2000](http://www.microsoft.com/windows2000).) Microsoft believes that rights should be granted only through groups, not through the directory's OU structure. Perhaps Microsoft is confused—more than 90 percent of NDS deployments are currently using this “undesirable feature” to reduce their management tasks and to simplify their users' computing experience.

#### SETTING UP SECURITY

Not surprisingly, Novell and Microsoft took different approaches to security. To understand how rights are inherited in both NDS eDirectory and Active Directory, you must first understand dynamic inheritance and static inheritance.

Dynamic inheritance uses directory or hierarchy information to determine access rights. For example, if access rights are granted at O=Novell, subordinate objects automatically inherit these rights, even though the actual subordinate objects aren't directly updated. Thus, if a tree has

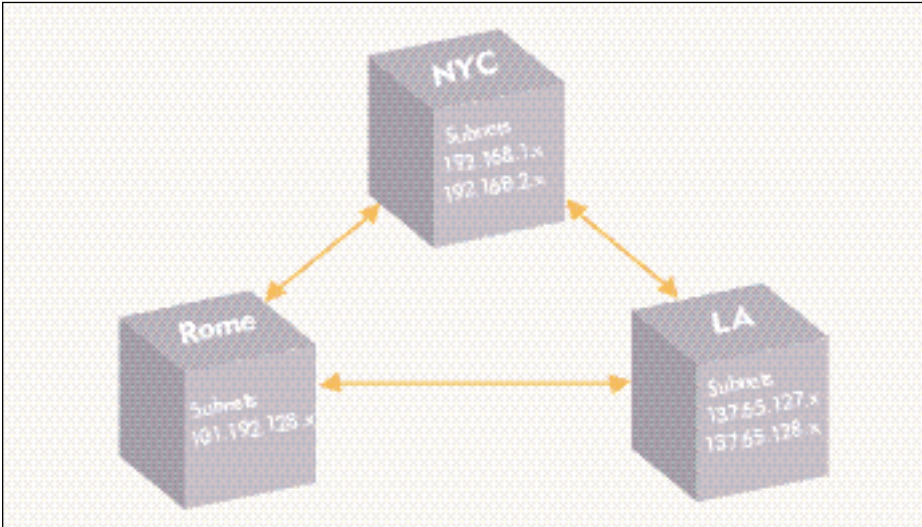
### Comparing Dynamic and Static Inheritance

#### Dynamic

Uses directory hierarchy to calculate rights  
 Less information is written into the directory  
 May need to resolve to [Root] object to calculate rights  
 Calculating access control may be a multistep process ●

#### Static

Each object holds all access control information  
 Access control calculation is a single-step process  
 Directory database grows very large with simple access rights



**Figure 3.** With Active Directory, you can use a domain site, which is a group of subnets, to represent the physical location of objects and services.

one million users, only one change must be written at the O=Novell level in the NDS tree to affect all one million users.

Static inheritance doesn't use directory or hierarchy information to determine access rights. Instead, static inheritance "inherits" the rights by applying the change to all child objects. Thus, if a tree has one million users, all one million objects must be updated to inherit the access right.

Both static and dynamic inheritance have advantages and disadvantages: For example, dynamic inheritance is more scalable since less information must be written to the directory. Static inheritance generally doesn't scale as well since every object must contain all access rights implicitly. (For more information about advantages and disadvantages of dynamic and static inheritance,

see "Comparing Dynamic and Static Inheritance" on p. 24.)

NDS eDirectory is extremely intelligent about how rights are dynamically inherited down the NDS tree from parent objects to child objects. In a pure dynamic inheritance model, NDS eDirectory would have to resolve all the way up to the [Root] object for every access request. Because this model isn't efficient in a partitioned environment, NDS eDirectory uses a hybrid static-dynamic inheritance model: Rights granted to parent objects are statically stamped to child partition root boundaries (but not to all child objects). NDS eDirectory can then calculate rights without contacting another NDS server (such as an NDS server higher in the hierarchy). (See Figure 4 on p. 30.)

Active Directory is based more on a static-stamped model. Rights granted in the Active Directory domain "flow" down by being written to all child objects. (See Figure 5 on p. 30.) As a result, granting rights within Active Directory greatly increases the size of the domain database and impacts server utilization during operation.

Unfortunately, every Active Directory system has a nasty little problem brewing—unchecked growth of the Active Directory database. Simple day-to-day operations can and will cause the Active Directory database to grow to hundreds of megabytes—perhaps gigabytes—on all domain controllers. Contrast this with NDS eDirectory; even the largest multinational NDS deployments rarely exceed 10 MB. (For more information about the difference between the size of NDS databases and Active Directory databases, see "A Real-World Experiment.")

Active Directory has other limitations in the way rights are inherited. Because the domain is the security and administrative boundary of Active Directory, static inheritance works only within the domain. This inheritance does not cross trust relationships between domains.

In NDS terms, a permanent, hidden inherited right filter exists between all Active Directory domains, so that any administrative rights assigned in parent domains never cross into child domains. Although multiple domains are linked together into an Active Directory domain tree, they are managed as separate domains. In other words, you cannot grant administrative rights or rights to resources at the top of a domain tree and expect these rights to "flow" to lower domains through trust relationships. (See Figure 6 on p. 32.)

### A Real-World Experiment

If you have access to Windows 2000, you can try the following real-world experiment to determine how NDS eDirectory and Active Directory track access rights: Install both NDS eDirectory and Active Directory, delegate the following administrative rights, and check the results. Your results will resemble the following:

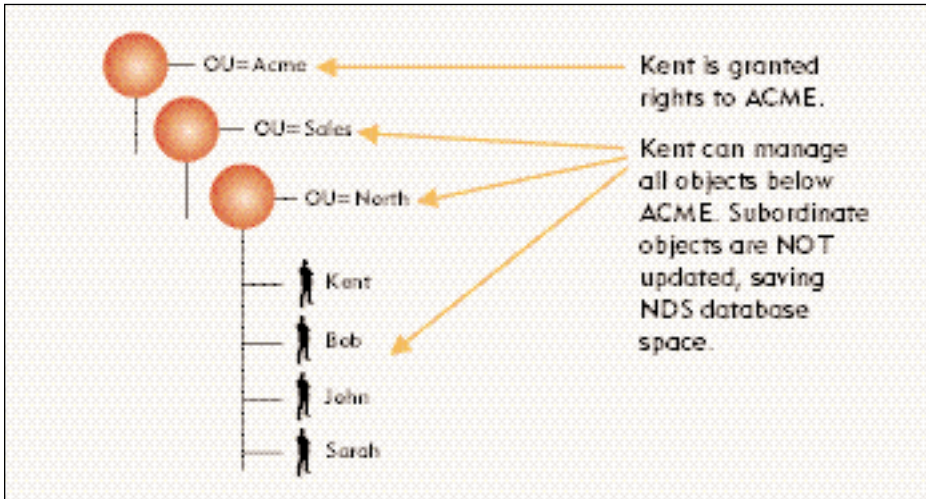
	NDS	ACTIVE DIRECTORY
Base directory installation	.7 MB	47 MB
Add 50,000 users	74.5 MB	280 MB
Create a password administrator at the top of the tree	74.5 MB	380 MB
Number of directory updates	1 (dynamic inheritance)	50,000+ (static inheritance)
Server utilization static during stamping (20 minutes)	0 percent	40-70 percent ●

### ACTIVE DIRECTORY AND DNS—THE NITTY GRITTY DETAILS

As you know, DNS is used primarily as a name service, which translates human-friendly host names (such as www.novell.com) into computer-friendly IP addresses (such as 137.65.72.1). The DNS NS record associates human-friendly web names with computer-friendly IP addresses. For example, the following is a typical DNS NS record:

```
www.novell.com ns 137.65.72.1
(DNS name) (record type) (IP address)
```

Of the DNS records used on the Internet, more than 98 percent of the DNS



**Figure 4.** To work efficiently in a partitioned environment, NDS eDirectory uses a hybrid static-dynamic inheritance model.

entries are NS records used for name services—mapping a human-friendly host name to a computer-friendly IP address. Less popular (but very important) are DNS service records. Service records allow an enterprise to advertise well-known services to other Internet users and devices.

For example, one of the most popular service records is the DNS mail-exchange (MX) record, which advertises the name of the computer servicing Internet mail. If a computer needed to send e-mail to sam@acme.com, the sending e-mail system would query DNS to find the MX record for acme.com. This MX record would identify the mail server at acme.com that is responsible for processing Internet mail.

In other words, the DNS MX record advertises a service (Internet mail). Less

than 1 percent of all Internet DNS entries advertise service information.

Although DNS is a wonderful name service—translating millions of host names to IP addresses—DNS isn't nearly as adept at advertising and locating network services, especially in a dynamic environment that includes WAN links. DNS may advertise the existence of a service, but DNS doesn't advertise the state of the service (available or unavailable). For example, a DNS MX record will advertise which server is responsible for processing Internet e-mail; however, the DNS MX record will not advertise whether or not the mail server is actually available and online.

Today's enterprise networks commonly use several dynamic service advertisement

systems. These systems advertise network services (such as file, print, and authentication) and the current state of the service. Some of the more popular dynamic service advertisement systems are listed below.

- IPX Service Advertising Protocol (SAP) advertises services in IPX environments.
- Microsoft Windows NT Browser advertises services in Windows NT 4.0 domain systems.
- IP Service Location Protocol (SLP) advertises services in IP environments. (For more information about SLP, download Request for Comments [RFC] 2165 from [www.rfc-editor.org](http://www.rfc-editor.org).)

Each of these systems tracks the current state of the services by "listening" for service advertisements. If the service is not heard within a certain period of time, the dynamic service advertisement system removes the service from its list. The following is a sample service record:

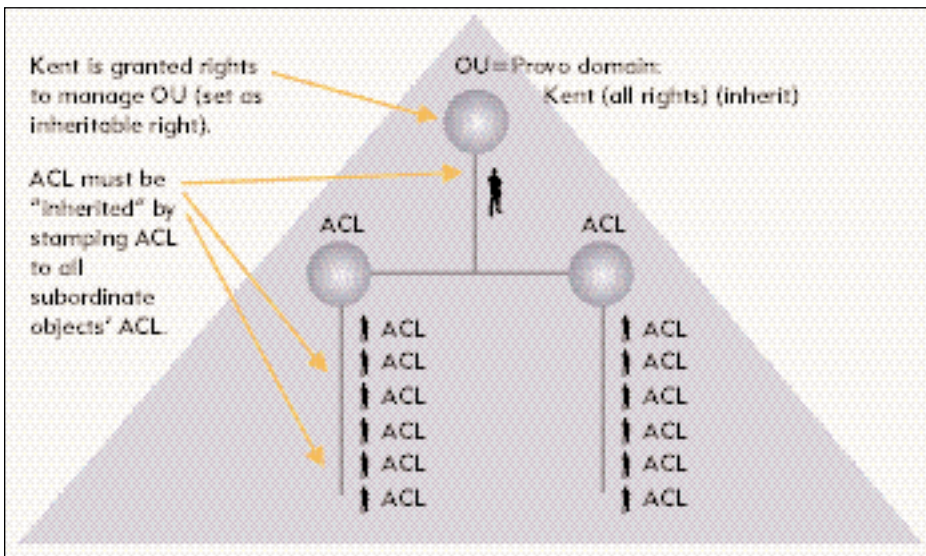
```
FS1 file server 137.65.1.90 12:01pm
(name) (service type) (address) (last registration time)
```

By tracking the registration time, these service advertisement systems can "age out" or delete service entries that are not heard within a given period of time. Service entries may not be heard because the service is no longer available or because the network path to that service is unavailable (if a WAN link is broken, for example).

Network clients contact these service advertisement systems to find interesting network services. By tracking the last registration time of network services, these service advertisement systems can respond to client requests with only those network services that are available.

#### ADVERTISING AND LOCATING ACTIVE DIRECTORY SERVICES

As mentioned earlier, Windows NT 4.0 includes a dynamic service advertisement system known as the Microsoft Windows NT Browser (hereafter referred to as the NT Browser). The NT Browser tracks all services for the domain, including servers, printers, authentication servers, and even workstations. As Windows NT servers are started or stopped, the NT Browser dynamically updates its database of service information. Workstations browsing Network



**Figure 5.** Active Directory uses a static-stamped inheritance model.

Neighborhood see only servers that are currently available. If a server shuts down unexpectedly, the NT Browser deletes the service record so that workstations will not try to connect to an unavailable server.

When Microsoft introduced Active Directory, it tied Active Directory to DNS. Unfortunately, because DNS is a static name service, it is less dependable than the NT Browser (although the NT Browser is actually not that scalable).

Comparing a static DNS service record with a dynamic NT Browser record highlights the differences between the systems:

#### Static DNS

```
FS1.Acme.com file server 137.65.1.90  
(service name) (service type) (IP address)
```

#### Dynamic NT Browser

```
FS1 file server 137.65.1.90 12:01pm  
(service name) (service type) (IP address)  
(registration time)
```

With the Windows NT browser, a registration time is associated with all service records. This registration time allows workstations to discover only currently available services.

With Active Directory, on the other hand, workstations cannot determine whether or not the service is available. Each workstation must contact the service and wait (and wait and wait) until the server doesn't respond. The result? Increased traffic while workstations attempt to contact servers that are unavailable and increased user frustration while users wait for servers that will ultimately not respond.

The following analogy illustrates the importance of a dynamic service advertisement system: Suppose you were searching the yellow pages for a plumber. If the phone directory were static, it would list every plumber that has ever existed, whether or not the plumber was still in business and available for service calls. When browsing the yellow pages, you would not know whether or not a plumber were still in business (in other words, the state of the plumber's store). You would then have to contact all of the plumbers until one answers the phone.

Fortunately, the phone directory is not exactly static; records are updated periodically. If a plumber is no longer in business, his or her ad is removed from the yellow pages. Updating information prevents increased traffic (since customers aren't trying to contact out-of-business plumbers)

and customer frustration (since only available plumbers are listed in the directory).

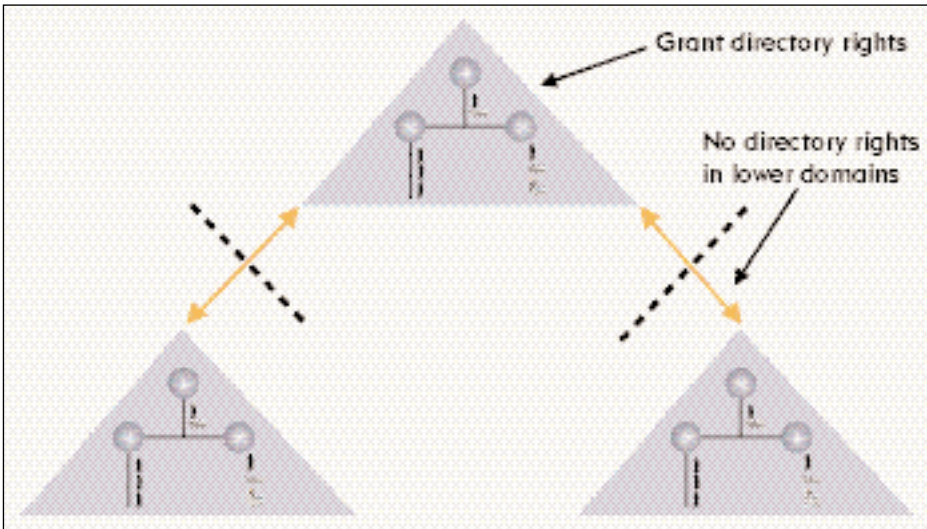
Microsoft realized that a static name service could not adequately represent a dynamic Active Directory environment. To solve this problem, Microsoft decided to change the behavior of DNS by adding proprietary Microsoft DNS extensions. Microsoft modified the Windows 2000 DNS server to include a registration time for DNS records. As a result, Windows

2000 DNS servers can track the state of Active Directory services, much like SLP tracks NDS services or the NT Browser tracks Windows NT 4.0 services.

Although modifying DNS solves Microsoft's problem of tracking dynamic services, this solution is alarming for several significant reasons:

- Microsoft has made a proprietary extension to DNS that is not supported by

**Visit our advertiser,  
LearnKey, at  
[www.learnkey.com](http://www.learnkey.com).**



**Figure 6.** If you grant rights to a domain in your company's domain forest, these rights do not "flow" down to other domains in that domain forest.

other DNS server vendors. This proprietary extension may impact interoperability between standards-based DNS servers and Windows 2000 DNS servers.

- Microsoft has not submitted its DNS extension to the DNS Extensions working group, which is part of the Internet Engineering Task Force (IETF).
- The default age-out period for the DNS extension is seven days. In other words, if a server disappears from the network, DNS continues to advertise the server for one week. Compare this age-out period to other systems: The age-out period for IPX SAP is five minutes, the age-out period for the NT Browser is 40 minutes, and the age-out period for SLP is 30 minutes. Microsoft's choice of a default age-out period is essentially useless. (I can see you telling users, "What's that

you say? You can see your server, but you can't authenticate to it? Call back in a week if it's still a problem.")

- Reducing the age-out period would greatly increase DNS replication traffic, especially for older or non-Windows 2000 DNS servers.
- To prevent service advertisement problems in an Active Directory environment, you must use Microsoft's Windows 2000 DNS server.
- How will these extensions affect other non-Windows 2000 DNS servers?

Microsoft could easily solve its service advertisement problem by adopting the Internet standard for advertising and locating network services SLP.

Unlike Active Directory, which is limited to static advertisement services, NDS

eDirectory can use both static and dynamic advertisement services: NDS static advertisements services include the HOSTS file and DNS. NDS dynamic services include SAP (for IPX environments), SLP (for pure IP environments), and NDS itself.

### REPLICATION BUGS THAT BITE

NDS and Active Directory also handle replication differently. In fact, replication problems are a known issue in Active Directory—an issue that probably won't be addressed for some time.

To understand the replication problem in Active Directory, you must first understand multivalued attributes. As the name suggests, a multivalued attribute is a directory attribute that may have multiple independent values. Multivalued attributes are used by many directory systems, including both NDS eDirectory and Active Directory, to represent such things as group membership, account security policy, and access rights.

For example, a group may have an attribute called *members*, which may also have multiple values. (Each value represents one user in the group.) Or, each user may have an attribute called *phone numbers*, which may have multiple values (such as home phone, cellular phone, and work phone).

Directories can use one of two methods to replicate multivalued attributes. The first replication method is the simplest: Multivalued attributes (such as group members) are replicated as a single update.

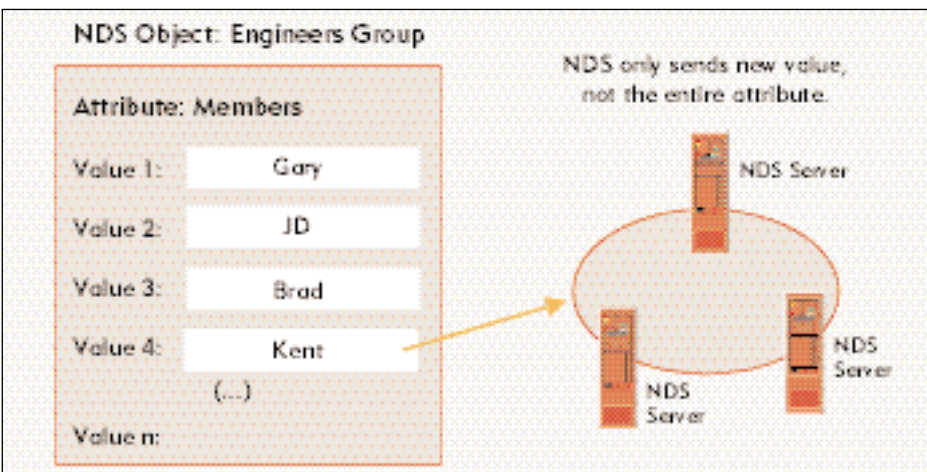
The second replication method is a bit more sophisticated: Multivalued attributes are replicated through multiple updates.

Active Directory currently uses the first replication method, and NDS eDirectory uses the second replication method. What does this difference mean to you as a network administrator? The following comparison shows how both replication methods work:

Suppose that the NDS Engineers group included the following users: Gary, JD, and Brad. These users were stored in the group members multivalued attribute.

If you added Kent to the Engineers group, the group members multivalued attribute would be updated with Kent's name. During NDS replication process, only the Kent multivalued attribute would be sent to the other servers.

This replication method is efficient: Rather than sending the entire list of



**Figure 7.** If you add a member to an NDS group, NDS eDirectory sends only that change to other NDS servers in the replication ring.

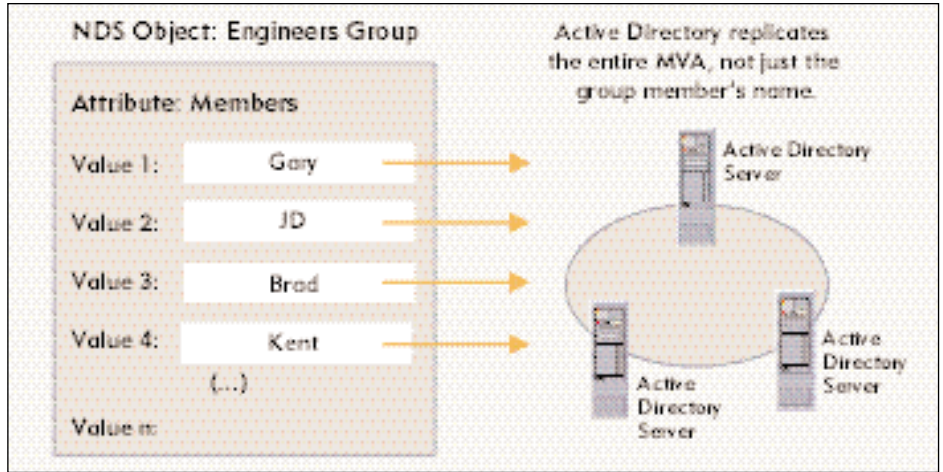
members, NDS eDirectory would send only the new member (Kent) the other servers. (See Figure 7.)

Now suppose that the Active Directory Engineers group included the following users: Gary, JD, and Brad. These users were stored on the group members multivalued attribute.

If you added Kent to the group, the group members multivalued attribute would be updated with Kent's name. During the Active Directory replication process, the entire multivalued attribute (containing all group members) would be sent to all other servers.

This replication method is less efficient: Rather than sending just the new member (Kent), Active Directory would send the entire list of members across the wire. (See Figure 8.)

Although NDS eDirectory uses a more efficient method of replication, you may wonder why you should worry about a few extra megabytes of wire traffic. If you are using Active Directory, however, you may need to worry about more than some extra network traffic.

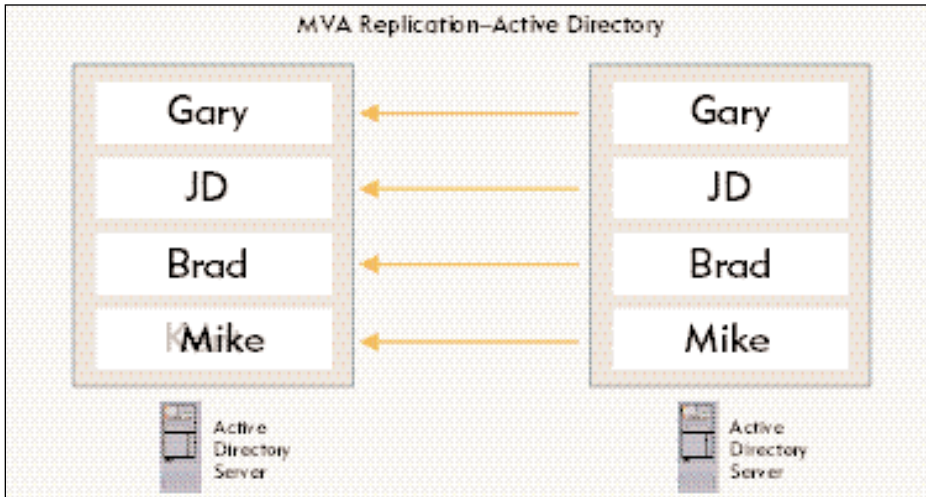


**Figure 8.** If you add a member to a domain group, Active Directory sends the entire list of group members to the servers in the replication ring.

For example, what would happen if two network administrators edited the same multivalued attribute at the same time (or, more correctly, before Active Directory had a chance to replicate the changes)? In this very real-world scenario, one of the changes would be lost.

Suppose a network administrator in Los Angeles added Kent to the Engineers group and a network administrator in New York added Mike to the Engineers group before the Los Angeles server and the New York server have a chance to replicate these changes. Because the Los Angeles and the New York servers would

**Visit our advertiser,  
CyberSTATEU, at  
[www.cyberstateu.com](http://www.cyberstateu.com).**



**Figure 9.** Active Directory has a known replication problem that may cause errors if two network administrators have responsibility for the same domain and make changes concurrently.

send their replication changes at the same time, one change would overwrite the other: That is, the Los Angeles server would send its entire group members multivalued attribute, and the New York server would send its entire group members multivalued attribute. Depending on when the changes arrived at each server, one of the two new users would be lost from the group! (See Figure 9.)

The problem gets worse as more domain controllers are added to the domain, or as Active Directory replication policies are implemented that increase the time between replication. (Increasing time between replication is a common way of gaining scalability in a directory. By increasing the periods between directory replication, you can reduce WAN traffic and server overhead.)

Worse yet, this replication problem doesn't affect just Active Directory groups. This replication problem also affects security settings on the domain and OUs, user account properties, and permissions.

As previously mentioned, this replication problem is a known design limitation in Active Directory that may be fixed in future releases (possibly the Whistler release). Until this time, Microsoft suggests a simple fix: Perform all domain administration from a single domain controller by "focusing" all Active Directory management consoles to use the same Active Directory server, essentially bypassing the multimaster feature of Active Directory.

Although this workaround may prevent the replication problem from occurring, it also raises several questions:

- How do you choose a focus server?
- What if the focus server is down?
- What if the focus server is on the other side of a slow WAN link? What if the WAN link to the focus server is down?
- What happens when a network administrator forgets to set the focus server?
- What about applications that interact with Active Directory? How will these applications know the focus server?

#### MAKING BEAUTIFUL MUSIC TOGETHER

For all the differences between NDS eDirectory and Active Directory, for all the strengths of one and weaknesses of the other, you may face the challenge of managing both systems. Rather than pulling out what is left of your hair, you can take advantage of solutions from the following companies:

- **Microsoft.** Over the years, Microsoft has developed or acquired different technologies for integrating NDS eDirectory and Active Directory. For example, Microsoft Directory Synchronization Services (MSDSS) is a useful tool for managing mixed NDS eDirectory and Active Directory environments. MSDSS synchronizes Active Directory and NDS objects by monitoring changes in both systems. MSDSS periodically polls each directory service, scans for changes in each directory, and then reconciles changes between the systems. MSDSS is a relative inexpensive solution, but it is limited in the types of objects that it can synchronize. In addition, MSDSS has an inefficient design, which limits it to smaller NDS instal-

lations, and it does not synchronize passwords. (For more information about MSDSS, visit [www.microsoft.com](http://www.microsoft.com).)

- **Novell.** With Novell's DirXML technology as the foundation, Novell User Account Manager (UAM) for Windows 2000 will provide synchronization services for linking Active Directory and NDS eDirectory. Novell's UAM for Windows 2000 will synchronize users and OUs between NDS eDirectory and Active Directory. UAM will also allow you to use ConsoleOne to manage Active Directory domains and groups.

In addition, this solution will provide strong password synchronization capabilities, allowing users and administrators to change passwords from either Novell or Microsoft client desktops or utilities. Because Novell's synchronization solution will be directory event-driven, it will be the only reasonable choice for medium- and large-scale deployments. (For more information about this soon-to-be-released product, visit [www.novell.com](http://www.novell.com).)

- **NetVision.** NetVision's Synchronicity for Active Directory links NDS eDirectory and Active Directory together through NetVision's NDS event system. NetVision's Synchronicity for Active Directory includes password synchronization capabilities and includes a snap-in module for the NetWare Administrator (NWADMIN) utility, which allows you to manage both environments. (For more information about Synchronicity for Active Directory, visit [www.netvision.com](http://www.netvision.com).)

#### CONCLUSION

No one argues that Active Directory improves upon Windows NT 4.0 management. Yet as a 1.0 release, Active Directory may take years to stabilize to a level that NDS eDirectory has enjoyed for some time. When working with Active Directory, keep in mind that it is much different than NDS eDirectory. Never make assumptions about Active Directory functionality based on how NDS eDirectory works. Try to keep an open mind and, whenever possible, test Active Directory in a lab before you put it into production.

Gary Hein has studied, designed, and implemented directory solutions for Novell's Fortune 500 customers since the 1.0 release of NDS. Hein has also worked closely with Microsoft's Active Directory since the initial Windows NT 5.0 betas were released in 1996. ●