CHAPTER 3

Designing the Active Directory

IN THIS CHAPTER

Active Directory is the core of the Windows Server 2003 network. It is the central component that not only serves to provide authentication and authorization, but also administration, information sharing, and information availability. It can be defined as follows:

> "A secure virtual environment where users can interact either with each other or with network components, all according to the business rules of the enterprise."

Quite a change from Windows NT, isn't it? It's no wonder people have not accepted Active Directory (AD) at a neck-breaking pace. It is a paradigm shift that is even more complex than moving from character-based computing to the graphical interface. Understanding the breadth of possibilities Active Directory brings is the biggest challenge of the enterprise network with WS03.

The first rule you must set for yourself when working to design your Active Directory is "Use best practices everywhere!" Don't try to change the way Active Directory is designed to work no matter what you might think at first. Active Directory provides a wealth of opportunities that you will discover as you implement, use, and operate it. Changes that might make sense according to IT concepts today may well have a negative impact on the operation of your Active Directory tomorrow.

The first step toward the implementation of the enterprise network—you could say the major step toward this implementation—is the design and implementation of your Active Directory. Even if you have already implemented Active Directory and are using it with Windows 2000, a quick review of how you design and plan to use directory services in your network can't hurt, unless you are completely satisfied with the way your directory delivers service. In that case, you can move on to Chapter 4 to review your communications infrastructure and begin installing the enterprise network. *If, on the other hand, you are using Windows NT and want to move to WS03, the following section is a must and cannot be overlooked under any circumstances.*

Introducing Active Directory

Countless books, articles, and presentations have been written on the subject of Active Directory, and it is not the intention of this book to repeat them. However, it is important to review a few basic terms and concepts inherent in Active Directory. Figure 3-1 illustrates the concepts that make up an Active Directory.

Active Directory is first and foremost a database. As such it contains a *schema*—a database structure. This schema applies to every instance of Active Directory. An instance is defined as an Active Directory *forest*. The forest is the largest single partition for any given database structure. Every person and every device that participates in the forest will share a given set of attributes and object types. That's not to say that information sharing in Active Directory is limited to a single forest. Forests can be linked together to exchange certain information, especially with Windows Server 2003. WS03 introduces the concept of *forest trusts* which allow forests to share portions of their entire Active Directory database with others and vice versa.

If you compare the WS03 forest to Windows NT, you can easily see that while NT also included an identity management database—the domain—its scope was seriously limited compared to Active Directory. NT could basically store the user or computer name along with passwords and a few rules affecting all objects. The basic WS03 AD database includes more than 200 object types and more than 1,000 attributes by default. You can, of course, add more object types or attributes to this database. Software products that take advantage of information stored in the Active Directory will

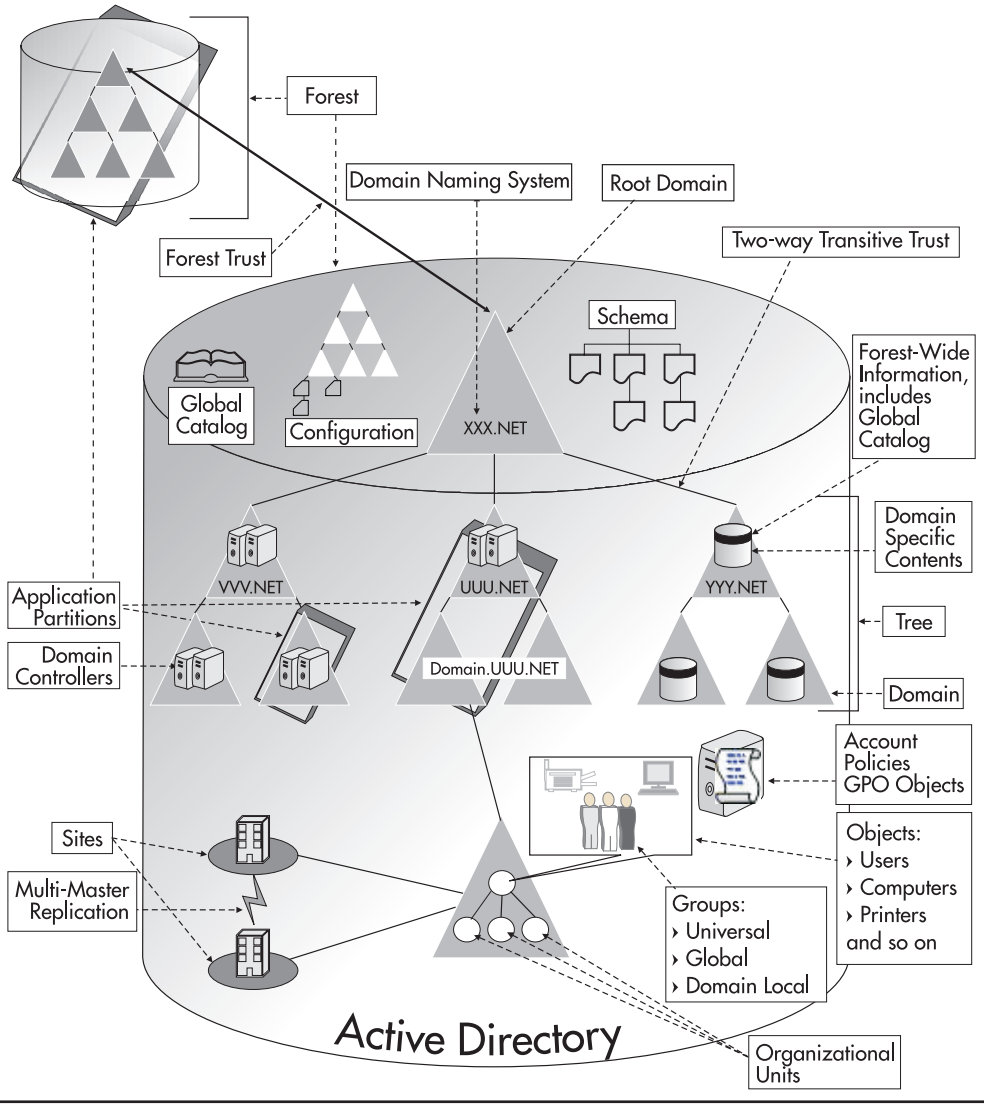


Figure 3-1 The Active Directory database

also extend the AD schema. Microsoft Exchange, for example, practically doubles the number of objects and attributes in a forest because it is integrated to the directory.

Like any database, AD categorizes the objects it contains, but unlike relational databases, Active Directory's database structure is hierarchical. This is because it is based on the structure of the Domain Naming System (DNS), used on the World Wide Web. On the Web, everything is hierarchical. For example, the root of Microsoft's Web site is www.microsoft.com. Everything spans from this page.

Moving to any other section, such as TechNet or MSDN, sends you to pages whose names are based on the microsoft.com root.

Forests act in the same way except that in a forest, the root point (analogous to the home page) is the root domain. Every AD forest must have at least one domain. Domains act as discrete object containers in the forest. Domains can be regrouped into *trees*. Trees are segregated from each other through their DNS name. For example, Microsoft has a multitree forest. Its namespace, the DNS element that defines the boundaries of the forest, is microsoft.com. As such, all domains in this tree have names similar to domain.microsoft.com. Microsoft created a second tree when it incorporated MSN.com in its forest. The MSN.com namespace automatically created a tree and all domains under it are named domain.MSN.com.

Every forest will include at least one tree and at least one domain. The domain is both a security policy and an administration boundary. It is required to contain objects such as users, computers, servers, domain controllers, printers, file shares, applications, and much more. If you have more than one domain in the forest, it will automatically be linked to all others through automatic transitive two-way trusts. The domain is defined as a security policy boundary because it contains rules that apply to the objects stored in it. These rules can be in the form of security policies or Group Policy Objects (GPOs). Security policies are global domain rules. GPOs tend to be more discrete and are applied to specific container objects. While domains are discrete security policy boundaries, the ultimate security boundary will always be the forest.

Domain contents can be further categorized through grouping object types such as *Organizational Units* (OUs) or *groups*. Organizational Units provide groupings that can be used for administrative or delegation purposes. Groups are used mainly for the application of security rights. WS03 groups include Universal, which can span an entire forest, Global, which can span domains, or Domain Local, which are contained in a single domain. OUs are usually used to segregate objects vertically. Objects such as users and computers can only reside inside a single OU, but groups can span OUs. Thus they tend to contain horizontal collections of objects. An object such as a user can be included in several groups, but only in a single OU.

Users also have it easier with Active Directory. Working in a distributed forest composed of several different trees and subdomains can become very confusing to the user. AD supports the notion of user principal name (UPN). The UPN is often composed of the username along with the global forest root name. This root name can be the name of the forest or a special alias you assign. For example, in an internal forest named TandT.net, you might use name.surname@tandt.com as the UPN, making it simpler for your users by using your *external* DNS name for the UPN. Users can log on to any domain or forest they are allowed to by using their UPN. In their local domain, they can just use their username if they prefer.

Forests, Trees, Domains, Organizational Units, Groups, Users, and Computers are all objects stored in the Active Directory database. As such, they can be manipulated globally or discretely. The single major difference between Active Directory and a standard database is that in addition to being hierarchical, it is completely decentralized. Most Active Directory databases are also distributed geographically because they represent the true nature of an enterprise or an organization.

Managing a completely distributed database is considerably more challenging than managing a database that is located in a single area. To simplify distributed database issues, Active Directory introduces the concept of *multimaster replication*. This means that even though the entire forest database is comprised of distributed deposits—deposits that, depending on their location in the

logical hierarchy of the forest, may or may not contain the same information as others—database consistency will be maintained. Through the multimaster structure, AD can accept local changes and ensure consistency by relaying the information or the changes to all of the other deposits in the domain or the forest. This is one of the functions of the Domain Controller object in the directory.

The only deposits that have exactly the same information in the AD database are two domain controllers in the same domain. Each of these data deposits contains information about its own domain as well as whatever information has been determined to be of forest-wide interest by forest administrators. At the forest level, you can determine the information to make available to the entire forest by selecting the objects and the attributes from the database schema whose properties you want to share among all trees and domains. In addition, other forest-wide information includes the database schema and the forest configuration, or the location of all forest services. Published information is stored in the Global Catalog. AD publishes some items by default, such as the contents of Universal groups, but you can also add or subtract published items to your taste. For example, you might decide to include your employees' photos in the directory and make them available forest-wide.

NOTE

Not all items are unpublishable; some items are prerequisites for the proper operation of Active Directory Services.

Whatever is published in the Global Catalog is shared by all domain controllers who play this role in the forest. Whatever is not published remains within the domain. This data segregation controls the individuality of domains. Whatever is not published can contain discrete information that may be of the same nature, even use the same values, as what is contained in another domain. Properties that are published in the Global Catalog of a forest must be unique just as in any other database. For example, you can have two John Smiths in a forest so long as they are both in different domains. Since the name of the object includes the name of its container (in this case, the domain), Active Directory will see each John Smith as a discrete object.

Figure 3-2 illustrates the contents of the directory store, or the NTDS.DIT database, that is located on every domain controller in the forest. Three items are in every directory store—the schema, the configuration and the domain data—and two are optional—the Global Catalog and the application partition (defined later).

The Global Catalog, schema, and configuration are information that is replicated throughout the forest. Domain data is information that is replicated only within the domain. Replication over local and distant networks is controlled through regional database partitions. Organizations may decide to create these partitions based on a number of factors. Since the domain is a security policy boundary, authoritative organizations—organizations that span a number of geographic locations they control—may want to create a single domain that spans these locations. To segregate each region, and control the amount and timing of database replication between regions, the domain would be divided into *sites*. Sites are physical partitions that control replication by creating boundaries based on Internet Protocol (IP) addressing.

Organizations that are not authoritative, have independent administrations, do not control their regional locations, or have slow links between each location, may want to further control replication through the creation of regional domains. Regional domains greatly reduce replication since only

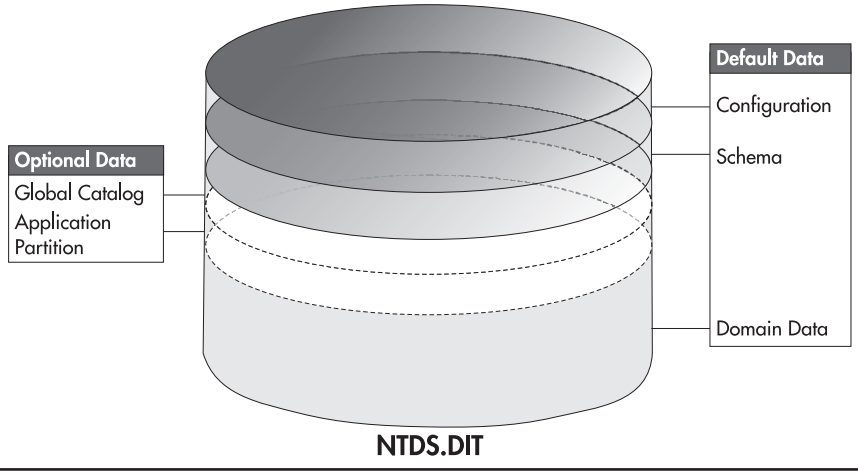


Figure 3-2 The structure of the directory store

forest-wide information is replicated from location to location. Forest-wide information rarely exceeds 20 percent of global forest data. In addition, organizations that only have the control of a portion of the forest namespace will be owners of the trees in the forest. Organizations that cannot guarantee a minimum level of consensus or authority between groups will always create separate forests.

There is one more replication partition in the Active Directory. This partition is new to Windows Server 2003. It is the *application partition*. This partition has several features such as the ability to host several instances of the same application and COM+ components on the same physical machine, but for the purposes of replication, this partition can be defined as a specific group of domain Controller IP addresses or DNS names. For example, WS03 automatically creates a forest-wide application partition for forest-wide DNS data so this information will be available on all domain controllers with the DNS role in the forest.

That's it. That's the basis of Active Directory. What's truly impressive about this database is that once it's in place, it can let you do some amazing things. You can manage an entire network from a central location. All management interfaces are the same throughout the forest, even across forests. Since everything is hierarchic, you can implement forest-wide standards for naming conventions, operations, database structure, and especially, security policy implementations. If you do it right, you can implement these standards automatically. This must be done before you create anything below the root domain. Though simple to understand, Active Directory is indeed quite powerful.

New Features for Active Directory

Windows Server 2003 boasts several improvements in regards to Active Directory. While this technology was introduced in Windows 2000, it has been refined and enhanced in WS03. Table 3-1

> ***QUICK TIP***
>
> *A complete glossary of Active Directory terms is available at http://www.Reso-Net.com/WindowsServer/.*

lists the new features found in WS03 for Active Directory since Windows 2000. This table first identifies new features that can operate within a mixed Windows 2000 and WS03 forest, and then identifies features that can only operate in a native WS03 forest.

Feature	Description
Multiple selection of directory objects	Modify common attributes of multiple users at one time.
Drag-and-drop functionality	Move directory objects from container to container in the domain hierarchy. Add objects to group membership lists.
Improved search capabilities	Search functionality is object-oriented and provides an efficient browse-less search that minimizes network traffic associated with browsing objects because it focuses on the local directory store.
Saved queries	Save commonly used search parameters for reuse in Active Directory Users and Computers.
Active Directory command-line tools	Run new directory service commands for administration scenarios.
InetOrgPerson class	This class has been added to the base schema as a security principal and can be used in the same manner as the user class. The userPassword attribute can also be used to set the account password.
Application directory partitions	Configure the replication scope for application-specific data among domain controllers running WS03S, WS03E, and WS03D. The Web Edition does not support the Domain Controller role.
Add additional domain controllers to existing domains using backup media	Reduce the time it takes to add an additional DC in an existing domain by using backup media instead of replication.
Universal group membership caching	Prevent the need to locate a Global Catalog across a WAN during logon by caching user Universal group memberships on an authenticating domain controller.
New domain- and forest-wide Active Directory features (in a Windows Server 2003 native domain or forest mode)	
Domain controller rename	Rename domain controllers without first demoting them.
Domain rename	Rename any domain running Windows Server 2003 domain controllers. This applies to NetBIOS or DNS names of any child, parent, tree-, or forest-root domain.
Forest trusts	Create a forest trust to extend two-way transitivity beyond the scope of a single forest to a second forest.
Forest restructuring	Move existing domains to other locations in the domain hierarchy.
Defunct schema objects	Deactivate unnecessary classes or attributes from the schema.
Selective class creation	Create instances of specified classes in the base schema of Windows Server 2003 forest, such as country, person, organizationalPerson, groupOfNames, device, and certificationAuthority.

Table 3-1 New Active Directory Features

Feature	Description
Dynamic auxiliary classes	Provide support for dynamically linking auxiliary classes to individual objects, and not just to entire classes of objects. Auxiliary classes that have been attached to an object instance can subsequently be removed from the instance.
Global Catalog replication tuning	Preserve the synchronization state of the Global Catalog by replicating only what has been changed.
Replication enhancements	Linked value replication allows individual group members to be replicated across the network instead of treating the entire group membership as a single unit of replication.
Reduced directory store	In native WS03 forest mode, the directory store is 60 percent smaller than in Windows 2000 because it can take advantage of the Single Instance Store feature, which does not duplicate redundant information on a disk.
Unlimited site management	In a native WS03 forest, the Knowledge Consistency Checker (KCC)—the service that automatically manages replication topology—can manage the topology for an unlimited number of sites. In Windows 2000, this service had to be turned off if your directory had more than 200 sites.

Table 3-1 New Active Directory Features *(continued)*

You can see from Table 3-1 that WS03 supports several functional modes for Active Directory. You can run AD domains in Windows NT mixed mode, which limits the functionality of AD to Windows NT capabilities; you can run domains in Windows 2000 native mode, which limits WS03 functionality to Windows 2000 AD capabilities; or you can run them in WS03 native mode. This last mode precludes the inclusion of any domain controllers other than WS03 within a domain. WS03 includes a second native mode: the WS03 native forest mode. While a WS03 forest can still include domains that operate in any of the three modes, a native WS03 forest can only include native WS03 domains. Table 3-2 identifies the differences between domain modes: Windows NT mixed mode, Windows 2000 native mode, and WS03 native mode. It serves to identify the limitations of Windows NT and Windows 2000 domain modes. It also includes the features of a native WS03 forest.

Both Tables 3-1 and 3-2 will be useful for the next step, designing your enterprise Active Directory.

The Nature of Active Directory

One final key element to understand before you move on to the creation of your Active Directory design is the nature of the directory. You already understand that a directory is a distributed database and as such must be viewed as distributed data deposits. But databases and data deposits include two basic components:

- **The database service** The engine that allows the database to operate
- **Data** The data contained in the database

Feature	Windows 2000 Mixed	Windows 2000 Native	Windows Server 2003 Native
Domain-wide Features			
Number of objects in domain	40,000	1,000,000	Same as Win2K
Domain controller rename	Disabled	Disabled	Enabled
Update logon timestamp	Disabled	Disabled	Enabled
Kerberos KDC key version numbers	Disabled	Disabled	Enabled
User password on InetOrgPerson object	Disabled	Disabled	Enabled
Universal groups	Disabled (security groups). Allows distribution groups.	Enabled. Allows security and distribution groups.	Same as Win2K
Group nesting	Disabled (for security groups, allows only group nesting for groups with domain local scope that have groups with global scope "Windows NT 4.0 rule" as members). For distribution groups, allows full group nesting.	Enabled. Allows full group nesting.	Same as Win2K
Converting groups	Disabled. No group conversions allowed.	Enabled. Allows conversion between security groups and distribution groups.	Same as Win2K
SID history	Disabled (security groups). Allows universal scope for distribution groups.	Enabled. Allows universal scope for security and distribution groups.	Same as Win2K
Forest-wide Features			
Global Catalog replication tuning	N/A	Disabled	Enabled
Defunct schema objects	N/A	Disabled	Enabled
Forest trust	N/A	Disabled	Enabled
Linked value replication	N/A	Disabled	Enabled
Domain rename	N/A	Disabled	Enabled
Improved replication	N/A	Disabled	Enabled
Dynamic auxiliary classes	N/A	Disabled	Enabled
InetOrgPerson object class	N/A	Disabled	Enabled
Reduced NTDS.DIT size	N/A	Disabled	Enabled
Unlimited site management	N/A	Disabled	Enabled

Table 3-2 Windows NT Mixed, Windows 2000 Native, and WS03 Native Domains

The WS03 directory is the same as any other database. Active Directory management is divided into two activities: service management and data management. AD management is comparable to intranet Web site management. Technicians and technical staff are required to manage the service behind AD just like the Web service for the intranet site, but users and user departments must be responsible for and administer the data contained in the AD as they would for information contained in the intranet pages.

For AD, the management of the data contained in the database can and should be delegated. Users should be responsible for their own information—telephone number, location, and other personal information—and departments should be responsible for information that is department-wide—organization structure, authority structure, and so on. Of course, user and departmental information should be validated before it is stored in the directory. Often, the best way to manage and delegate this information is through the use of a Web form located on the intranet. This allows the concentration of all delegated data in a single place. In addition, the Web form can support a content approval process before being put into the directory. For example, this content approval process could be delegated to the Human Resources department.

Service management—management of domains, Operation Masters, domain controllers, directory configuration, and replication operations—must be maintained and operated by IT. Delegating data management tasks takes the pressure off IT staff and allows them to focus on IT-related operations within the directory such as database service management.

Designing the Solution: Using the Active Directory Blueprint

Like the Enterprise Network Architecture Blueprint presented in Chapter 1 (refer back to Figure 1-5), the Active Directory Design Blueprint emerges from the structure of the Microsoft Certification Exam number 70-219, "Designing a Microsoft Windows 2000 Directory Services Infrastructure." It also includes the same prerequisites: business and technical requirements analyses. The advantage of using the same blueprint structure for both operations is that you should already have most of this information in hand. If not, now's the time to complete it. Without this information, you can go no further. You simply cannot achieve a sound Active Directory design without fully understanding your organization, its purpose, its objectives, its market, its growth potential, its upcoming challenges, and without involving the right stakeholders.

Your Active Directory design must be flexible and adaptive. It must be ready to respond to organizational situations that you haven't even anticipated yet. Remember, Active Directory creates a "virtual space" where you will perform and manage networked operations. Being virtual, it is always adaptable at a later date, but if adaptability is what you're looking for, you need to take it into account at the very beginning of the design.

Once you have the information you need, you can proceed to the actual design. This will focus on three phases: partitioning, service positioning, and the implementation plan.

QUICK TIP

To help simplify the AD Design Process for you, sample working tools are available at http://www.Reso-Net.com/WindowsServer/. The first is a glossary of Active Directory terms. You can use it along with Figure 3-1 to ensure that everyone has a common understanding of each feature. There is also an AD Design Blueprint Support Checklist that follows the steps outlined in Figure 3-3. It is a working process control form that lets you follow the AD Design Process stage by stage and check off completed tasks. In addition, there is an OU documentation table that will support your OU creation process. These tools will help you design the AD that best suits your organization's requirements.

AD Partitioning

Partitioning is the art of determining the number of Active Directory databases you want to manage and segregating objects within each one. This means you will need to determine the number of forests your organization will create, remembering that each is a separate database that will require maintenance and management resources. Within each forest, you will need to identify the number of trees, the number of domains in each tree, and the organizational unit structure in each domain. Overall, you'll need to identify if your Active Directory database will share its information with other, non-AD databases. This will be done either through integration of the two database structures (if the other database is compatible to the Active Directory format) or information exchange. In this case, you will need to identify the information exchange strategy.

To control data replication, you will identify and structure sites, design replication rules, and identify replication methodology. This is *Site Topology* Design. Microsoft provides an excellent tool to support you in this process, the Active Directory Sizer. It is found at http://www.microsoft.com/windows2000/downloads/tools/sizer/.

Since you intend to fully exploit the AD database (after all, why go through all this trouble if you're not going to use it?), you'll have to put in place a Schema Modification Strategy. Since every schema modification is replicated to every domain controller in the forest, you'll want to ensure you maintain a tight control over these. You might even decide to separate application from network-based schema modifications. Of course, all schema modifications will go through lab testing before making it to the production network.

AD Service Positioning

Site Topology Design is closely related to Service Positioning. Each Active Directory domain controller performs important operations that support the proper functioning of the database. In fact, the object of Site Topology Design is to determine how each of these database containers will be linked to the others. Since AD is a distributed database, domain controllers should be positioned as close as possible to the user. These points of service should be convenient without becoming overabundant.

Operation Masters are special domain controllers that manage global forest or global domain operations. Global Catalog (GC) servers are domain controllers that maintain copies of information that is published throughout the forest. But since WS03 domain controllers can cache frequently requested

global information, GC servers do not need to be as widely spread as domain controllers. Finally, DNS servers are a must since they provide namespace management functionality to the directory. They should be seen as subsidiary functions for directory support and married to every domain controller. Proper positioning of each of these services can vastly improve directory performance.

Implementation Plan

The last step of the blueprint is the AD Implementation Plan—the actual procedure you will use to put your Active Directory design in place. Indeed, this is where the Parallel Network Strategy comes in handy. It gives you the freedom to implement a brand new Active Directory without any limitations. This directory can immediately operate in native mode since it does not have to share database space with previous technologies. The limitations of Windows NT can be contained in specific domains or can even be excluded entirely from your Windows Server 2003 enterprise forest. In this way, you can obtain immediate benefits from native-mode Active Directory functionalities. The blueprint for AD design is illustrated in Figure 3-3.

Putting the Blueprint into Action

While the information collected for business requirements is the same as the information collected for the Enterprise Network Architecture Blueprint, your view of the information collected for technical requirements has to be slightly different. In particular, the second section, "Impact of the Enterprise Network," is changed to "Impact of Active Directory." Here you need to see how existing systems and applications will be affected by the arrival of a central database containing primary information such as usernames and user identity. You also need to see how these systems and applications can be integrated with this new central data repository.

You need to review planned upgrades and rollouts to make sure they will be compatible with Active Directory and that these projects will not negatively affect the rollout of an enterprise AD. In terms of IP infrastructure, your focus needs to be the internal network Domain Naming System since this function becomes integrated with the directory itself. You need to identify how the technological support structure functions in your organization in order to determine who has authority over what. This will allow you to determine where your authoritative AD boundaries (Forests, Trees, and Domains) will lie and where you will be able to perform delegation (through Organizational Units). You also need to review your system management structure (both current and planned) to see which functions you will want to delegate or integrate to Active Directory. Finally, you need to review your current identity management deposits, either Windows NT or Windows 2000 domains or other deposits such as Novell Directory Services or even UNIX systems to see how they will be integrated or how they will interact with the WS03 directory.

Once this is complete, you can proceed to the third step of the blueprint, the partitioning design. The directory partitioning exercise allows you to determine the scope, naming strategy, Organizational Unit strategy, integration model, position for core services, topology, and Schema Modification Strategy for each forest in your enterprise.

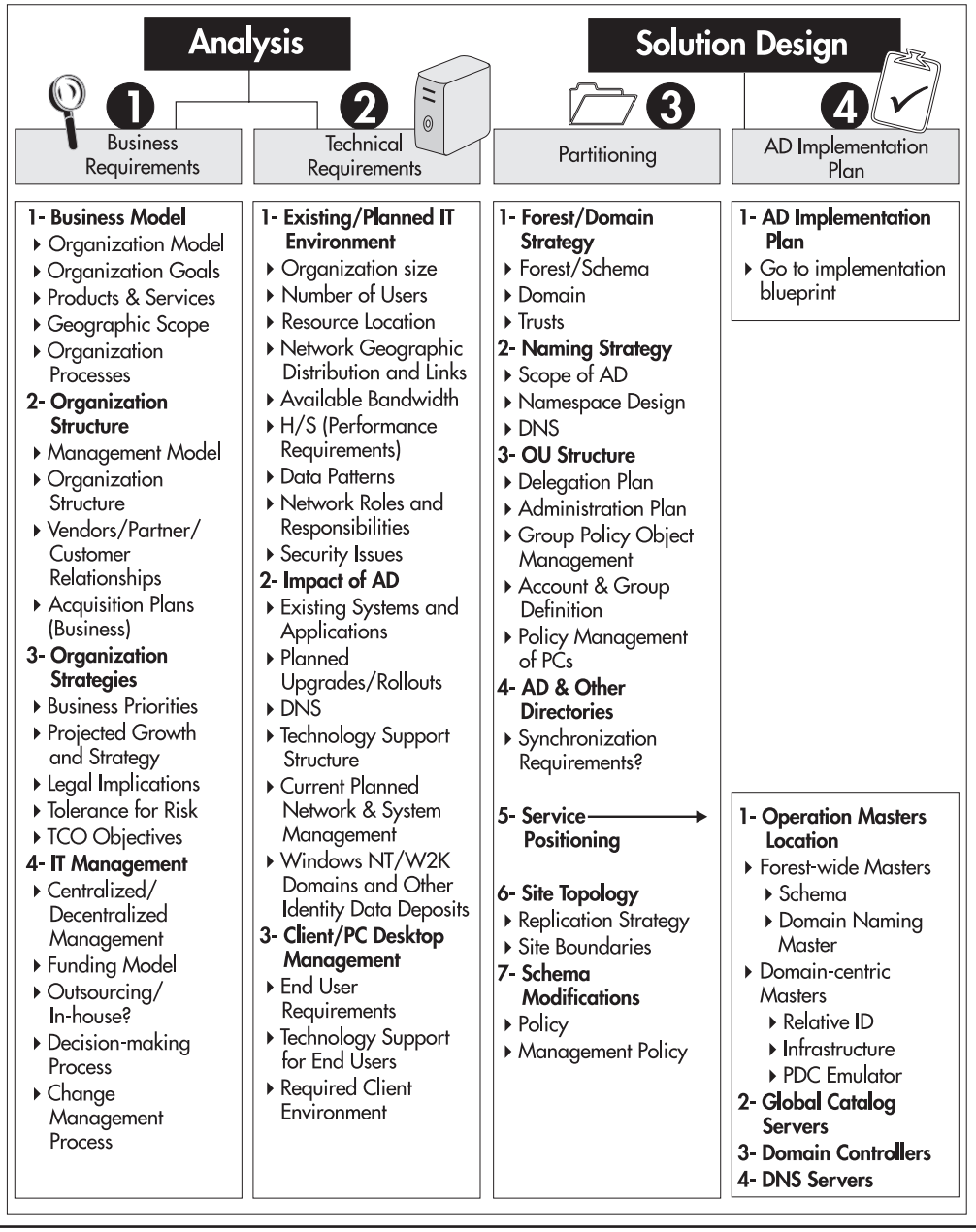


Figure 3-3 The Active Directory Design Blueprint

NOTE

Microsoft produced an excellent partitioning guide, "Best Practice Active Directory Design for Managing Windows Networks." It can be found at www.microsoft.com/windows2000/techinfo/planning/activedirectory/bpaddsgn.asp.

Forest/Tree/Domain Strategy

The first step in the partitioning exercise is to determine the number of forests, the nature of the trees in each forest, and the nature of the domains in each of the trees your enterprise will require. Forests are the partitions that contain:

- **Database schema** Only one database structure can be stored in a single forest. If someone in your organization needs to modify the schema and does not want to share this modification with others in the organization, they should be placed in their own forest. Obviously, this would not be departments that share physical locations, but it could be a subsidiary or a partner organization. Commercial and/or corporate applications will also personalize the schema. With the advent of forest trusts, you might decide to use an *application forest* to store an application with its own schema inside a different forest and link it to your enterprise network Active Directory through a trust. This strategy keeps the two schemas completely separate.

QUICK TIP

You could also use Active Directory in Application Mode (AD/AM) for this purpose. AD/AM is a special directory service that is an add-on to WS03 and that is designed to run as a pure lightweight application protocol (LDAP) directory. Its schema is much smaller than AD's, though—it contains 30 objects and 160 attributes. More information is available at http://www.microsoft.com/windowsserver2003/techinfo/overview/adam.mspx.

- **Configuration data** The structure of the forest, the number of trees it contains, and the domains in each tree as well as the structure of replication sites make up the configuration data for the forest.
- **Global Catalog** The Global Catalog includes all of the searchable objects for the forest. It contains the values and properties for all of the objects you deem important to users in the entire forest.
- **Trust relationships** Trust relationships between the domains in a forest are also forest-wide information. This is because of the transitive nature of Windows Server 2003 intra-forest or inter-domain trusts. Every domain in a forest will automatically be linked to its parent domain. The parent domain will be linked to its parent and so on. Since all domains of a forest include two-way transitive trusts, all domains trust all other domains of the forest.

In Windows NT, you needed to create specific trusts between each domain if you wanted domains in a group to trust each other. Trusts were not transitive. That means that Domain A would not trust Domain C even if they both trusted Domain B. For Domain A to trust Domain C, you had to create an explicit trust. You do not need to create direct trusts between domains in a forest. If Domain A and Domain C both trust Domain B in a forest, Domain A will automatically trust Domain C without an explicit trust. You can create shortcut trusts if the hierarchical path between two domains that share a lot of information is too long or too complex. This is illustrated in Figure 3-4.

Forests can contain millions of objects, so most small, medium, and even large organizations will usually require a single production forest. The main reason for the creation of separate forests within the same organization is to protect the database schema. Schema modifications are complex and must be tightly controlled if you want to minimize their impact on production environments. If you need to play or experiment with the schema, you need to create a forest that is separate from your production forest. Most medium to large organizations have development and test forests as well as at least one production forest.

A second reason for the segregation of forests is the level of authority of the central organization. You can only include organizations, divisions, or departments over which you have political and economic control in your forest. This is because of the hierarchical nature of the forest and the inheritance model that is derived from it. The organization at the root of the forest has influence and even authoritative control over all of the organizations or departments that are grouped into its trees and subdomains. For example, the Ford Motor Company and Volvo would have had separate forests before the acquisition of Volvo by Ford. But once Ford bought Volvo, it established financial authority over Volvo. In an Active Directory, Volvo could then become a tree under the Ford production forest. Much depends on how well the Volvo and Ford IT staffs get along and whether Ford will impose the joining even if the Volvo staff does not agree.

As you can see, no matter the size of your production forest—whether it is in a small enterprise located within a single site or a multinational spanning the world, the role of the forest owner is an important one. Forest owners manage forest-wide services. This means they are:

- **Forest-wide operation masters administrators** The forest owner is the administrator of the domain controllers that execute the Schema and Domain Naming Master services and thus can impact the entire forest.

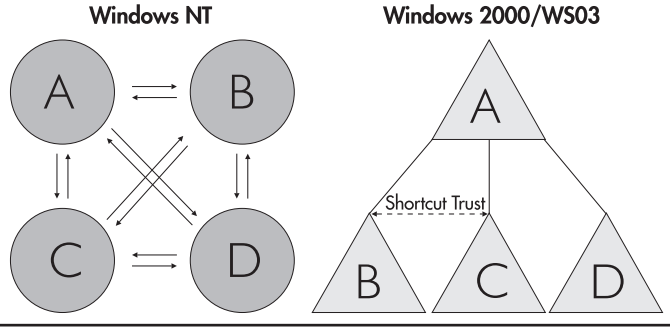


Figure 3-4 Windows NT trusts versus Active Directory trusts

- **Root domain administrators** Every forest, even if it only has a single tree and a single domain, includes a root domain. The first domain in a forest is the root domain because all other domains in the forest are created as subdomains of the root domain. The operation of the root domain is critical if the forest is to run properly.
- **Root domain data owner** Since the root domain is the basis of the forest, the forest owner is also the root domain data owner.
- **Schema and configuration owner** Since the forest operation is based on the structure of its schema and configuration containers, the forest owner is responsible for their integrity.
- **Forest-wide security group owner** The forest owner is also responsible for forest-wide security groups. These groups reside in the root domain. Active Directory creates two management forest-wide groups: Enterprise Administrators and Schema Administrators. Membership in these groups is limited because they can affect the operation of the entire forest.
- **Root domain security group owner** In addition to the two universal administration groups, the root domain contains its own administrative group, Domain Administrators. The forest owner is also owner of this security group.

If there is more than one domain in the forest, the forest owner will have to communicate frequently with subdomain owners to coordinate forest-wide efforts.

In fact, determining the number of forests in your organization can be summarized as the identification of all forest owners. These will be the highest level of IT administration in the organization for any given network. Once this is done, you will be able to proceed to identifying the forest content.

Forests share a lot of elements. Many are required elements; others are recommended elements based on common sense. Forests require the sharing of:

- **Security** Only include people you trust in a forest. This would include employees as well as administrative staff. Since a forest is made up of distributed database containers, domain controllers, you need to trust the people who will be responsible for all domain controllers that will be placed outside your office site.

▶ CAUTION

This point is extremely important. Even though you can secure domain controllers by locking down the system and placing the servers in locked rooms, you should be absolutely sure that any DCs that will be in distributed locations are under the responsibility of people in whom you have absolute trust. Because of the multimaster replication model of Active Directory, a rogue domain administrator who has physical access to a DC can do a lot of damage in a forest. For example, they can take the DC offline and edit the directory store in debug mode, adding special access rights for themselves. Once the DC is back on line, these access rights are replicated throughout the forest. There are ways to control this remotely, and they will be covered in Chapter 8. For now, include remote offices in your forest only if you can trust that your DCs will be safe from tampering.

- **Administration** Everyone who participates in a forest is willing to use the same schema and configuration.

- **Name resolution** Everyone who participates in a forest will use the same Domain Name System to resolve names throughout the forest.

In addition to the required elements, you might decide to share the following:

- **Network** If all organizations in a forest trust each other, they may have put a private network in place. Though it is not impossible to separate forest sites with firewalls, it is recommended to minimize the exposure of your Active Directory information to the outside world. If forest members must use public network links to transport replication traffic, they may opt for separate forests.
- **Collaboration** If you work with other organizations and have implemented domain trusts with them, they may well be candidates for joining your new AD forest.
- **IT groups** If organizations share IT groups, it is a good idea to create single forests to simplify network administration.

You must also keep in mind that creating more than one forest will have administrative impacts:

- Forests do not share transitive trusts. In WS03, these trusts must be created manually, but once created will allow two entire forests to trust each other. If forests need to interact at a specific domain level, you can still use explicit domain trusts between the two specific domains limiting the trust relationship between the forests. Both forest and domain trusts can either be one- or two-way trusts.
- The Kerberos security protocol (the native Windows Server 2003 authorization protocol) will only work between forests that have implemented forest trusts.
- Using an email-like logon name (*name@domain*), the UPN, will also only work if a forest trust is in place.
- Global Catalog replication is limited to a single forest unless there is a forest trust in place.

Forest Design Example

Now that you're comfortable with the forest concept, you can identify the number of forests you need. Use the following examples to review the forest creation process.

The first design example focuses on the identification of the number of forests for a medium-sized organization with 5,000 users. It is distributed geographically into ten regions, but each region is administered from a central location. The organization operates under a single public name and delivers the same services in each region. Since the organization has a "buy, don't build" policy, it tries to make use of commercial software whenever possible, but even with this policy, it still needs to create custom code or adapt existing applications. Thus it requires a separate development environment.

In addition, it has had a lot of growing pains in the past because of friction between IT and IS. In fact, IS was seriously disappointed when IT created a single master domain network with Windows NT.

In their forest design, this organization would create at least two, possibly three or more, permanent forests:

- A production forest that replaces the single master Windows NT domain.
- A staging forest to test, analyze, and prepare new products for integration, especially those that may integrate with Active Directory and modify its basic database schema.
- A development forest to allow the testing and development of corporate applications that take advantage of schema customizations.
- A separate forest will also be created for the extranet. Because this forest is exposed through the security perimeter of the network, it is separate from the production forest.

No trust would be established between three of these forests: production, staging, and development. In an illustrated model, this is represented by solid lines separating each Active Directory database. There may, however, be a trust established between the perimeter forest and the production forest, but since the nature of this trust (one-way, explicit, domain-to-domain) is not completely precise at this time, its boundary with the production forest is displayed as a dotted line.

Production Forest Design

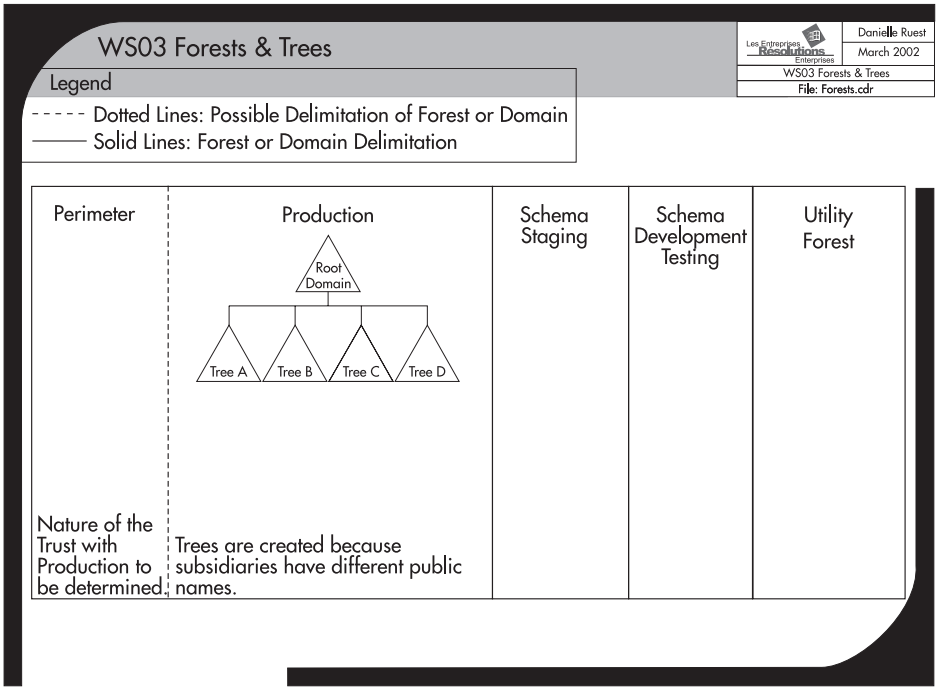
In the production forest design you will determine the structure of the forest you use to run your network. Once again, authority boundaries will determine the structure you create. Here you need to determine the number of trees and the number of domains your forest will contain.

Begin with the trees. Does your organization operate with a single public name? If not, these are good candidates for different trees. Even though the tree structure is completely internal and will rarely be exposed to the external world, its structure should reflect the names your organization uses publicly. Good candidates for trees are organizations that rely on others for service completion; organizations that form a partnership and want to collaborate closely; enterprises that merge with each other; and organizations that share IT management resources.

The second design example covers a tree design for a worldwide organization that has four subsidiaries. The organization is a single enterprise, but each of its business units is known under a different public name. It understands the complexity of interbusiness administration, but wants to implement operational and security standards throughout the corporation. IT budgets are controlled centrally, but most of the administrative work is performed by large IT groups from each of the business units.

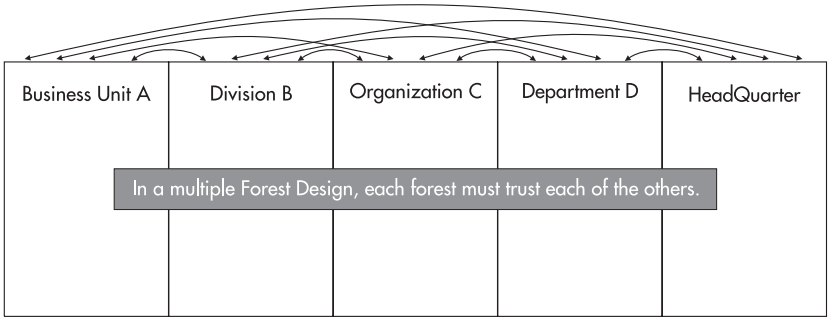
After a series of discussions, the different IT groups decided on a single production forest with multiple trees. The forest owner identified and began ongoing discussions with each tree owner. As a group, they determined the level of integration for each tree and the level of authority the forest root domain would be allowed.

This model uses the same number of forests as before, but now trees are created in the production forest. It allows the organization to set standards while supporting regional diversity.



Had the different IT groups not been able to agree, they would have created multiple production forests. In this case, the organization would not have met its goals for standardization because there is no technical way to ensure forests use the same standards. These goals could only have been obtained through political enforcement measures and not through the operational infrastructure of Active Directory.

Using forest trusts, an organization can interact through multiple forests, and thus gain benefits such as single sign-on and global interforest searches, but cannot enforce standards through AD.



Domain Strategy Design

The first thing to remember when working with Windows Server 2003 domains is that they are *not* like Windows NT domains. In Windows NT, the largest identity database boundary was the domain. If you wanted multiple domains to work with each other in either a master/master or a master/resource relationship, you had to enable trusts between each of the domains. In WS03, domain trusts in a forest are transitive. Here the domain must be viewed as what it is—a security policy boundary that can contain:

- **Authentication rules** Domains form the boundary for the rules used to authenticate users and computers since they are the container into which these objects are created.
- **Group Policies** Policies are limited by domain boundaries because they are objects that reside within the domain container.
- **Security policies for user accounts** Security policies applying to user accounts are stored in the domain. These can be different from one domain to another.
- **Publication services for shared resources** All of the resources that can be shared in a domain are published through Active Directory. By default, these resources—shared printers and folders—are published only to members of the domain.

Your domain design will depend on a number of factors: for example, the number of users in a forest and the available bandwidth for replication from remote sites. Even though domains can contain one million objects each, it doesn't mean you need to fill them up. You might decide to create multiple domains to regroup objects into smaller portions. If you find that you are applying the same policies to two different domains and it's not for replication control, you've got one too many. In fact, you may consider upgrading wide area network links to eliminate the need for multiple domains.

In addition, you can use several domain models just as in Windows NT. WS03 forests support the unique domain model, the multiple domain model, and the mixed model. Because of the hierarchical nature of the forest, these models are not like their Windows NT predecessors. Few organizations today opt for the unique domain model. Small businesses with fewer than 500 employees may decide to use this model, but it is very rare in larger organizations.

Most large organizations will decide to create a *Protected Forest Root Domain* (PFRD). There are several advantages to this approach. A Protected Forest Root Domain is often much smaller than production domains because it only contains forest management groups and users. As such, it has a minimum amount of data to replicate, which makes it easier to rebuild in case of disasters. It contains a small group of forest-wide administrators, which reduces the possibility of mistakes that may affect the entire forest. It is never retired since it does not contain production data. Because domains are created below the forest root domain, organizational restructuring is easier to accomplish. Because it is small and compact, it is easier to secure. And should transfer of ownership be required, it is easier to transfer an empty domain than to transfer your entire production domain which contains all of your hundreds of users.

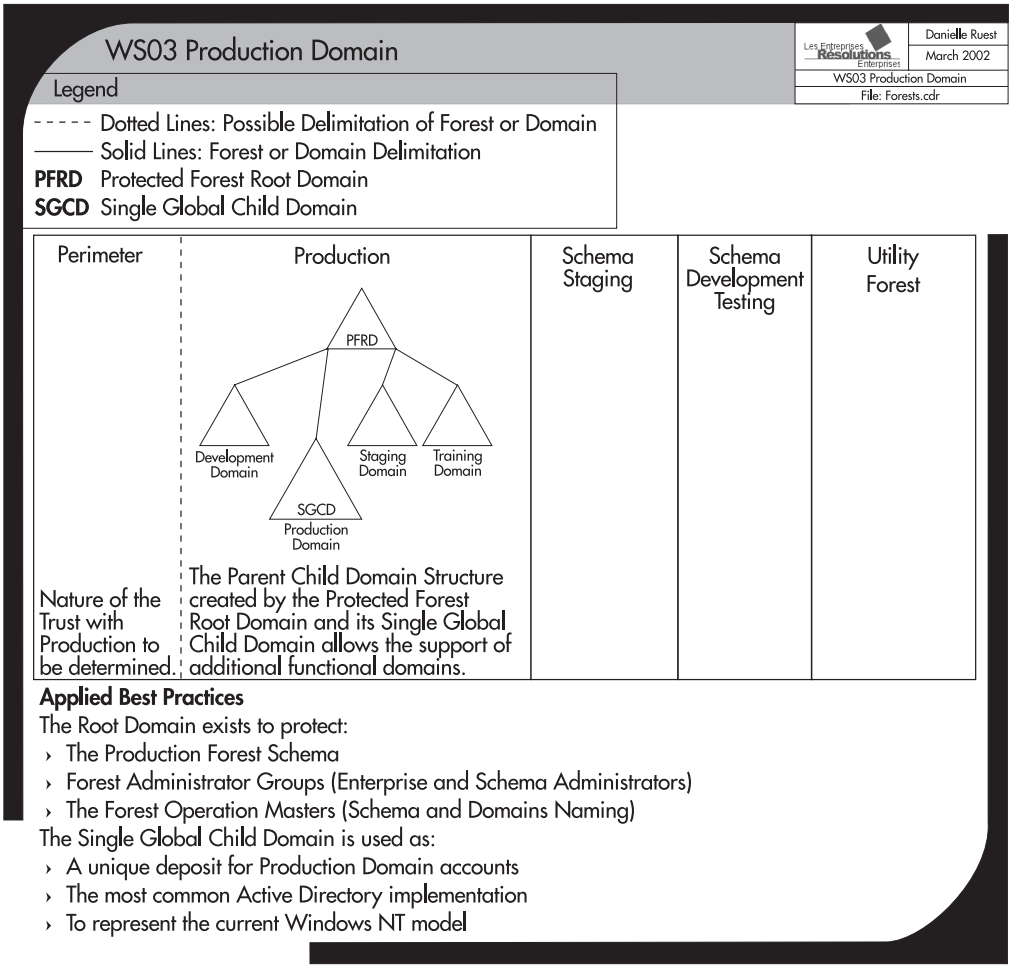
CAUTION

The Protected Forest Root Domain is the most commonly overlooked feature of an AD design. If your organization has more than a few hundred users and you can afford the domain controllers the PFRD requires, it is highly recommended that you implement a PFRD in your AD design, as it gives you the most flexibility in AD.

Production domains are created under the Protected Forest Root Domain. Any medium to large organization that has a single master domain in Windows NT should create a Single Global Child Domain. This Single Global Child Domain (SGCD) has the same purpose of the single NT domain: regroup all of the users of your network into a single production environment. The only users that are not in this child domain are the forest root domain users.

Now that you have a parent and child domain structure, you can expand forest contents to include other security policy boundaries. The main requirement of a Single Global Child Domain is that users be identifiable and that their actions be traceable within the network. As such, you will definitely want to exclude generic user accounts from the production domain. Generic accounts—accounts that are named according to function rather than individual—are most often used for three activities: testing, development, and training. You can use security policy boundaries—domains—to segregate these accounts from the production domain. In this manner, you can create other domain containers where rules can either be more or less stringent than in the production domain to enclose testing, development, and training. In fact, not all tests or development will require schema modification. In most organizations, 95 percent of all tests and/or development will *not* require schema modifications. The creation of both testing (or rather, staging) and development subdomains becomes quite easy since the parent/child structure is already in place. The same applies to a training domain. This is the *functional domain* design model. This model does not include multiple trees, but rather, multiple child domains.

Domains can be required in other situations as well. For example, an organization whose operations span several countries will often require multiple subdomains because of the legal restrictions in some of these countries. If there are legal requirements that differ from country to country and that may even require special security settings, you will need to create additional domain boundaries.



The final reason for domain segregation is WAN bandwidth. If your *available* bandwidth is inappropriate to support intra-domain replication, you will need to create regional domains. Specific information on bandwidth requirements is detailed later in this chapter, in the section "Site Topology Design."

Keep in mind that every domain you create will require an administration team. In addition, each new domain requires at least two domain Controllers for redundancy and reliability. The hardware costs may become prohibitive if too many domains are created. In addition, each new domain means new trust relationships. While they are transitive and automatic, they still need to be monitored. Finally, the more domains you create, the more it is likely that you will need to move resources and objects between domains.

Other Forest Domain Designs

Now that you have determined the domain structure to implement in your production forest, you can use it to derive the structure for the other forests you created. The staging forest is simple. It should represent the same structure as the production forest. As such, it requires a parent and a child domain. Since it is designed to represent only the production environment, it does not require additional domains for training, development, or other purposes.

The development and utilitarian forests require a single combined root and production domain since schema development testing is not dependent on the parent/child naming structure found in the production forest. Finally, the perimeter (extranet) forest is made of a single domain because this structure reduces the complexity of its management. Since it is exposed to the outside world (through a firewall, of course), its structure is also kept as simple as possible.

There you go! Your forest design is complete. It should resemble the illustration in Figure 3-7.

QUICK TIP

Development forests are created when organizations want to integrate their applications with the Active Directory they will use to manage their network. Because they must change the schema to integrate applications, developers working on these projects must be located in a separate forest. There are costs, of course, associated with this approach. You may decide that your AD production forest will be used only for network management (remember Figure 3-1—a network operating system directory can be complex). If so, you can use AD/AM to perform application integration with Active Directory. Using AD/AM eliminates the need for a development forest because it is a service that can reside either on a member server or even a Windows XP workstation. This can greatly reduce your AD development costs. More on this strategy is discussed later when you prepare your Schema Modification Strategy.

Forest Design Best Practices

The forest design process includes the following best practices:

- Identify the number of forests and write a justification for each one.
- Identify the number of trees and write a justification for each one.
- Wherever possible, create a Protected Forest Root Domain.
- Wherever possible, create a Single Global Child Domain for production in each tree.
- Identify the number of additional domains required in each tree.
- Identify the scope and contents of each domain.
- Justify each domain.
- Choose the generic name for each domain.
- Once the domain structure for the production forest is complete, design the domain structure for the other forests you created.

Designing the Naming Strategy

The next step is defining the Active Directory namespace. The namespace defines the scope of the Active Directory. It is based on the hierarchical nature of the Domain Naming System. Not only does it define the naming boundaries of the Active Directory database, but it also defines the structure of the database and the relationships between its objects. The actual object naming convention for Active Directory is not DNS. It is based on an X.500 naming scheme that identifies containers when naming objects. This allows for the creation of duplicate objects so long as they are located in different containers. For example, dc=com/dc=root/ou=IT/cn=User/cn=Mike Smith means that Mike Smith's user account is contained within the IT organizational unit in the root.com domain.

As you can see, the X.500 naming scheme is not practical for everyday use. But most everyone is familiar today with the Domain Naming System, so it is the naming scheme presented to users and administrators. Since it is hierarchical, DNS can be used to subdivide the forest into trees. This is done through the modification of the DNS root name. For example, MSN.com is a root name change from Microsoft.com, thus it is a second tree that is created in the Microsoft.com forest.

Since the domain name of your forest is a DNS name, you should use only registered DNS names. When you register a name, you ensure that you have complete ownership over it. For instance, if you use Microsoft.com as your external name, you might use Microsoft.net as your internal network name. By buying the rights to the Microsoft.net name, you ensure that no outside event will ever affect your internal network. You are also segregating your internal namespace from your external namespace. This allows you to identify the source of all traffic more easily and track intruders more effectively. Domain names can be registered with Internic. A complete list of domain name registrars by location can be found at http://www.internic.net/origin.html.

If, for some reason, you choose to use a name you do not own, be sure you verify that it does not exist on the Internet before creating your first domain controller. Organizations that do not perform this step often find themselves using an internal name that is used externally by a different organization. This will cause problems that range from having to rename your forest to being unable to reach the external domain from inside the network. Even though renaming an entire forest is possible with Windows Server 2003, it doesn't mean that you'll find it pleasant to have to change your internal name because someone outside your organization forces you to do so. Use a real DNS name with standard DNS naming conventions; with the .gov, .com, .org, .net, .edu, .biz, .info, .name, .cc, .tv, .ws, or .museum name extension and register it. That way, you'll control your namespace.

Never use the same forest name twice even if the networks are not interconnected. If you know that your sister organization has named their development testing forest DEVTEST, name yours something else. You'll also have to worry about NetBIOS names. NetBIOS names are composed of 15 characters with a reserved 16th character. They must be unique within a domain. The first part of the DNS names you choose should be the same as the NetBIOS name. Since DNS names can contain 255 characters per fully qualified domain name (FQDN), you will have to limit the size of the DNS names you use (in fact, you have 254 characters to choose from; DNS places a final dot in the name, the 255th character). Use short, distinct, and meaningful names, and distinguish between domain and machine names.

You should also identify your object naming scheme at this stage. All objects such as servers and PCs will have a distinct DNS name (or host name). This name, like the universal principal name for users, will have a DNS structure and use the domain and forest root names to complete its own. You

can use the naming scheme illustrated in Figure 3-5. In this scheme, every object uses TandT.net, a registered DNS name, as a forest root. Next, it uses either a geographic naming scheme for child domains (single letter code for region and three-digit number code for each region), or a functional scheme (function name such as Intranet.TandT.net). Finally, servers and PCs can use up to five letters for the function code along with three digits to identify the number of machines offering this function. An example would be ADDC001.Intranet.TandT.net for an Active Directory DC in the Intranet child domain of the TandT.net forest.

Forest, tree, and domain names should be considered static. You should try to find a name you will not need to change, even if you know you can later. The domain and domain controller renaming process in Windows Server 2003 is complex and can cause service outages. Geographic names are often the best. In most cases, it takes a lot of momentum to change a geographic name, so they are considered quite stable. Don't use organizational structure to name domains unless you are confident that it is and will remain stable.

Table 3-3 lists the type of objects that you could place within domains and the holding domain for each object. Each object will require naming.

Naming Best Practices

Use the following best practices to name your AD forests:

- Use standard Internet characters. If they work on the Internet, they will definitely work in your network. Avoid accents and solely numeric names.
- Use 15 characters or less for each name.
- For the root name, use a simple, short name that is representative of the identity of the organization.
- Follow all DNS standards and make sure the internal DNS name is different from your external name.
- Finally, before proceeding, buy the name.

DNS is a cornerstone of Active Directory. Since it is designed to manage the AD namespace, Microsoft has vastly enhanced the Windows DNS service. It can now be completely integrated with Active Directory. In fact, it should be because proper AD operation depends on DNS since DNS is

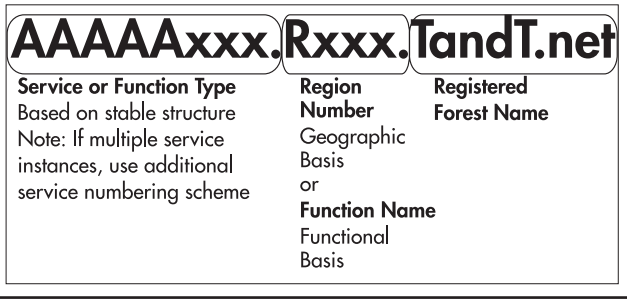


Figure 3-5 An object naming scheme

Objects	Production	Development	Training
Internal regular PCs	✓	✓	
Internal portables	✓	✓	
External PCs for development		✓	
Managed external PCs	✓	✓	
Unmanaged external PCs	✓	✓	
Multimedia PCs	✓	✓	
Member Servers (Services: HIS, SMS, SQL, and so on)	✓	✓	
Domain Controllers	✓	✓	✓
Quotas (shared folders)	✓	✓	
Printers and printer queues	✓	✓	✓
Meeting rooms	✓		
Projectors, shared PCs	✓		
Service accounts	✓	✓	✓
Users	✓	✓	
Administrators	✓	✓	✓
Technicians/installers	✓	✓	
Groups	✓	✓	✓
Generic accounts		✓	✓
Organizational Units	✓	✓	✓
Group Policy objects	✓	✓	✓
Domain administrators	✓	✓	✓
Applications	✓	✓	✓

Table 3-3 Domain Objects

used to locate domain controllers at logon. For this reason, you should avoid using third-party DNS servers with Windows, especially if they are non-Windows based. WS03 brings several enhancements to the DNS service so long as it is integrated with AD. With WS03, the DNS service has moved from being simply a network infrastructure service to become an Active Directory and Windows base service. More on this topic will be covered in Chapter 4.

The forest design can now be named. The production forest belongs to the T&T Corporation. Their Internet name is TandT.com. They have researched and bought TandT.net. It will be the name for their forest root. Subdomains are named after their function. The production domain is named with something more meaningful to users, such as Intranet.TandT.net. Development, training, and staging domains are named as such. The external forest found in the perimeter is named TandT.com. The staging forest is named TandTLab. This forest does not require a registered DNS name since it is not a production environment. The impact of recreating or renaming a staging forest is always much smaller than for the production forest. Volatile or utility forests can be named when needed. The development forest will not be retained because T&T Corporation has decided to use AD/AM for

application integration and will reserve its production Active Directory for NOS operations only. This model is illustrated in Figure 3-7.

Designing the Production Domain OU Structure

What's truly amazing with Active Directory is how a simple database can be used to manage objects and events in the real world. That's right, the objective of Active Directory is to manage the elements you store inside its database. But to manage objects, you must first structure them. Forests, trees, and domains begin to provide structure by providing a rough positioning for objects throughout the Active Directory database. This rough positioning needs to be vastly refined, especially when you know that a single domain can contain more than a million objects.

The tool you use to refine the structure of objects is the organizational unit (OU). An OU is a container that, like the domain, is designed as an object repository. OUs must be contained within a domain, however. But since they can act as object repositories, they can and should be used to identify your network administration structure. Remember also that OUs can store other OUs, so you can create an administrative structure that reflects reality.

A second advantage of an OU is the ability to delegate its management to someone else. This means that when you design the structure of the Organizational Units within the domains of your Active Directory, you design the way the objects in your network will be managed. In addition, you identify who will manage which components of your network.

For example, you might decide that users in a given business unit are the responsibility of the business unit, delegating the management and administration of this group of users to a local business unit administrator. In this way, the OU in Active Directory is comparable to the domain in Windows NT. Whereas in Windows NT you needed to give "Domain Administrator" rights to anyone responsible for groups of users, in Active Directory you delegate ownership of an organizational unit, thus limiting access rights to the contents of the OU and nothing else.

In short, the OU is designed to help support the data/service concept of Active Directory. Since OUs contain AD objects and their properties, they contain data. By controlling access to OUs through security settings, in much the same way you would for a folder on an NTFS volume, you can give someone ownership of the data contained in the OU. This frees domain administrators to manage AD services. Making sure that all AD services are healthy and operating properly is the new role of the domain administrator. In a well-rounded Active Directory, you have a series of new interaction roles such as the OU administrator, the domain operator, the service administrator—roles that have significantly less authority in a domain than their Windows NT counterparts. You can now limit the Domain Administrator group to a small, select group of people.

The OU Design Process

In this design process, administrators must create a custom OU structure that reflects the needs of their organization and proceed to the delegation of its contents where appropriate. The best place to start the design process is with the Single Global Child Domain. Since this is the production domain, it will be the domain with the most complex OU structure. Once this domain's structure is complete,

it will be simple to design the structure for other domains both within and outside the production forest since they are all derived from the production forest's requirements.

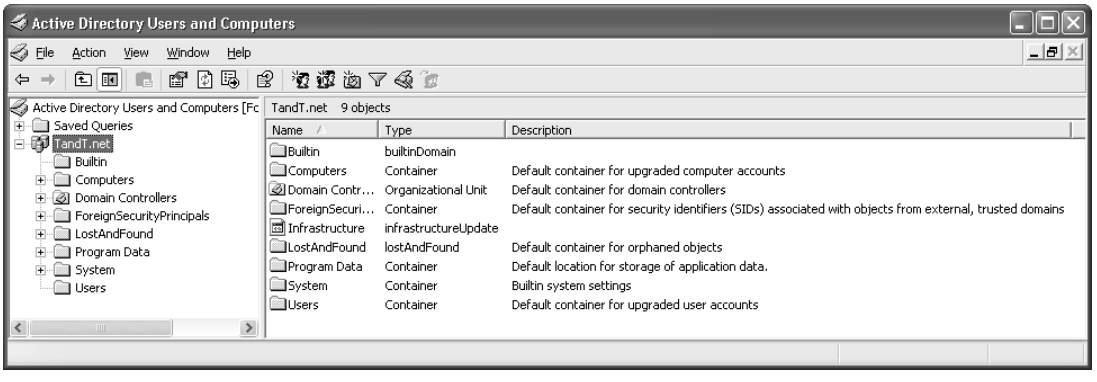
There are four reasons to create an Organizational Unit:

- It is required to regroup AD object types.
- It is required to administer AD objects.
- It is required to delegate the administration of AD objects.
- It may be required to hide objects.

Because OUs can include objects, you should first use them to regroup the different types of objects your network contains. There are three basic object types: People, PCs, and Services. These should create your first level of custom OUs.

Second, objects are regrouped for administrative purposes. You manage objects in AD through the application of Group Policy objects (GPOs). More on this is covered in Chapters 5, 6, and 7, but what is important to understand here is that the way you design your Organizational Unit structure will directly affect the way you apply Group Policy objects.

WS03 applies two policies by default to each domain: the Default Domain Policy and the Default Domain Controller Policy. You should review the contents of these policies to ensure that they conform to your security requirements. More on security is discussed in Chapter 8. WS03 also creates a number of default containers such as Users, Computers, and Domain Controllers. The only one of the three that is an OU is Domain Controllers. The other two are *not* OUs and cannot contain either GPOs or other OUs. If you want to control users and computers, you need to create a custom OU structure to regroup these types of objects.



The third reason for OU creation is delegation. Delegation should be considered hand in hand with administration to create the sublayer structure of OUs. For each type of OU, you must identify potential object subtypes and determine if they are significantly different. Each significantly different object, either at the administrative or delegation level, will require a separate OU. WS03 will support a hierarchy of more than ten levels of OUs, but you should try for as flat an OU structure as you can. Objects buried in multiple layers of Organizational Units will be very demanding to index and locate when you need to find them in the directory. Aim for a five-layer OU structure as much as possible.

Keep in mind that if you control only the top layers of the structure and you need to delegate its finalization, you should leave at least two untapped layers for local departments to use.

The final reason is to hide objects. Since OUs contain access control lists, it is possible to hide sensitive objects in the directory. These objects are placed in special OUs that have very tightly controlled access control lists. Thus the objects become "invisible" to non-administrative users of the directory.

The administration design process begins when you create the three different object type OUs—People, PCs, and Services—and regroup objects under them. To do so, you need to identify every manageable object in your network and use a questioning process for each. As an example, Table 3-4 lists a series of objects that require management within the directory. In addition, it defines a classification and expected contents for each object. Two questions need to be answered for each object: Will I ever delegate this object? Do I need to manage this object through Group Policy objects? Each "Yes" answer means that a custom OU needs to be created.

Objects	Classification	Contents	Delegation?	GPO?
Workstations	Resource OU	Users with elevated rights Generic users Multimedia PCs		✓
Portables	Resource OU	Users with elevated rights Generic users		✓
External PCs	Resource OU	PCs for development projects (managed)		✓
External PCs	Resource OU	Consultant PCs (managed)		✓
External PCs and Portables	Resource OU	Consultant PCs (unmanaged)		
Member Servers	Resource OU	Services: HIS, SMS, SQL, Exchange	✓	✓
Domain Controllers	Service OU	Services: Authentication, identity management, security		✓
Quotas (shared folders)	Resource OU	Information sharing		✓
Printers	Service OU	Delegate printer queues	✓	✓
Meeting rooms	Resource OU	Reservation system	✓	
Projectors, shared PCs	Resource OU	Reservation system	✓	
Service accounts	Service OU	System process tracking		✓
Users	Data OU	(Similar to organizational structure)	✓	✓
Administrators	Data OU	Master OU in a delegated OU		✓
Domain administrators	Service OU	Located in default OU		✓
Technicians/ installers	Service OU	Global objects, but with limited delegation rights	✓	✓
Groups	Service OU	Universal, Global, Domain Local	✓	✓

Table 3-4 Manageable Objects in AD

Objects	Classification	Contents	Delegation?	GPO?
Generic accounts	Data OU	Domains other than production	✓	✓
Applications	Service OU	COM+ objects, MSMQ	✓	✓

Table 3-4 Manageable Objects in AD *(continued)*

Though the OU design process begins with object categorization, it is not complete until you have also designed the following plans:

- Group Policy Object Management Strategy (Chapter 5)
- PC Management Strategy (Chapter 5)
- Account and Group Definition (Chapter 6)
- Service Management Strategy (Chapter 7)
- Security Design (Chapter 8)
- Delegation Plan (Chapters 5, 6, 7, and 8)
- Service Resilience Plan (Chapter 9)
- Administration Plan (Chapter 10)

Though you begin the OU design here, its design will not be complete until you consider each of the elements in the remaining chapters of this book. Each of these items has some impact on your OU design.

The PCs Object OU Structure Design

You'll begin by categorizing PCs. Table 3-4 identified six possible types of PCs in the organization. The organization has its own PCs and includes PCs from external sources as well. They are first divided into two categories: internally owned and external PCs. The former are all managed PCs, but have a few differences. Mobile computers have different policies than desktops. Among the desktops are basic PCs as well as multimedia and shared workstations. Among external PCs are managed and unmanaged systems. External PCs that are onsite for the development of code must be tightly controlled and must be the image of the internal PC build in order to ensure code quality. Other consulting PCs may be for productivity purposes only. PCs that are used only to produce documentation should not be the organization's responsibility. Therefore they need to be segregated within the OU structure. Of course, this structure assumes that PCs are managed centrally. If not, the PCs OU structure may resemble the People OU structure outlined later.

The Services Object OU Structure Design

Next, organize the services in your network. This means creating OUs to delegate application servers such as those from the Microsoft .NET Enterprise family: SQL Server, Exchange, Systems Management Server, and Host Integration Server. By placing the server objects within these OUs, you can delegate their management and administration without having to give global administrative rights. Each of

these servers should be a member server. Windows Server 2003 no longer requires services to be installed on domain controllers. Even Microsoft Message Queuing services, which required domain controllers in Windows 2000, now operate on Member Servers. You should always beware in WS03 when someone wants to install an application on a domain controller. Each of these services should be created under the Services root OU. In this way, if you need to apply a policy to all member server objects, such as a security policy, you only need to apply it to the root OU.

In addition to application servers, this OU should include services such as File and Print Servers. In fact, every server type identified in Chapter 1 and reviewed in Chapter 2 (except for Identity Management Servers) should be placed within an OU in this structure. This OU should also include all of the service accounts—special administrative accounts that are used to run services in a Windows Server 2003 network. These accounts are all data objects of the same type, so they can be managed through a single container. Finally, operational groups such as support technicians or system installers can be located in an Installer/Technician OU, making it easier to give them rights to other objects in the domain. The additional advantage of the Services OU is that it is much easier to locate objects of the service category.

The People Object OU Structure Design

The last OU structure to populate is the People OU. These OUs will contain either user accounts and/or groups. This is also the OU structure that will most resemble the organizational chart. In fact, the organizational chart is a good information source for the regrouping of the people in your enterprise in the directory. Few people know the organization's structure as well as the Human Resources group. This is a good place to enlist their assistance.

Like the organizational chart, the People OU structure defines a hierarchy of distinctiveness. The difference is that the two are inversed. The organizational chart defines a hierarchy of authority (who controls whom), whereas the People OU structure goes from the most common to the most distinctive. In the organizational chart, the employee mass is at the bottom. In the People OU structure, it is at the top.

When you want to manage all of the People object types, you can do so by applying a Group Policy to the root OU. The second level of this OU structure should reflect the business unit structure of the organization. Though the organizational chart is a source of information, it should not be used in its exact form because organization charts tend to change too often. You need to create as stable an OU structure as possible to minimize change in the directory. Because of this, many organizations only use lines of business (LOB) at the second level of OUs for the People object.

This OU level may also include special team groupings—business units whose purpose is to support all other business units across the enterprise on an administrative basis. It will also contain regional groupings if your organization spans a large geographic territory. In this case, regional groupings are essential since you must delegate ownership of regional objects to regional administrative representatives.

In most cases, you will generate three general levels of OU within this OU structure:

- **Root level** Used to manage all People objects (user accounts and groups). This level contains only other OUs and administrative groups for the structure.
- **Line of business level** Used to manage all user accounts that are within defined segments or lines of business within the corporation and located at headquarters or central offices, as well as

all groups for the entire line of business. The groups to whom this level is delegated are all located in the root OU.

- **Regional level** Used to manage regional offices. This includes user accounts for every line of business located in the regional office as well as regional groups. The parent OU for the regional OUs contains every regional administrative group.

  The third OU level may also represent groups or administrative services within the line of business level. For example, IT and IS will be found within the organization's administrative line of business, but you can be sure that they will not have the same policies and rights, so they are segregated at the third OU level. IT especially will most probably be segregated into further sublevels as well, but this will most likely be done through a process internal to the IT department. The final structure for IT will be delegated to the IT group.

The complete OU structure is illustrated in Figure 3-6. Here the OU shape identifies the purpose and contents of each OU.

Replicating the OU Structure to Other Domains

Now that you have a solid and complete OU structure, you can replicate it to other domains. Table 3-5 identifies the OU structure in other domains.

The completed Forest, Tree, Domain, and OU structure is illustrated in Figure 3-7.

Production OU Design Best Practices

Keep the following rules in mind when you create OU structures:

- Think in terms of equipment and objects in the directory.
- Determine how you will implement the administrative delegation process.
- Identify standards for all administrative categories in the organization.
- Use the administrative service or function or the line of business to name OUs. These tend to be more stable than organizational structure.
- Limit your structure to five levels, three if you are not responsible for the finalization of the structure. Recommend a maximum of five levels even though ten are possible. This gives you some breathing room.
- Remember the four reasons for the creation of OUs: categorization, administration, delegation, and segregation.
- Each OU you create must add value to the system.
- Never create an OU that does not contain any objects.
- Never create an OU that does not have a specific purpose.
- If an OU reaches an empty state, consider removing it. This may not be necessary because it may only be temporarily empty. If not, remove it.
- Identify an OU owner for each one you create. If no owner can be identified, remove the OU.

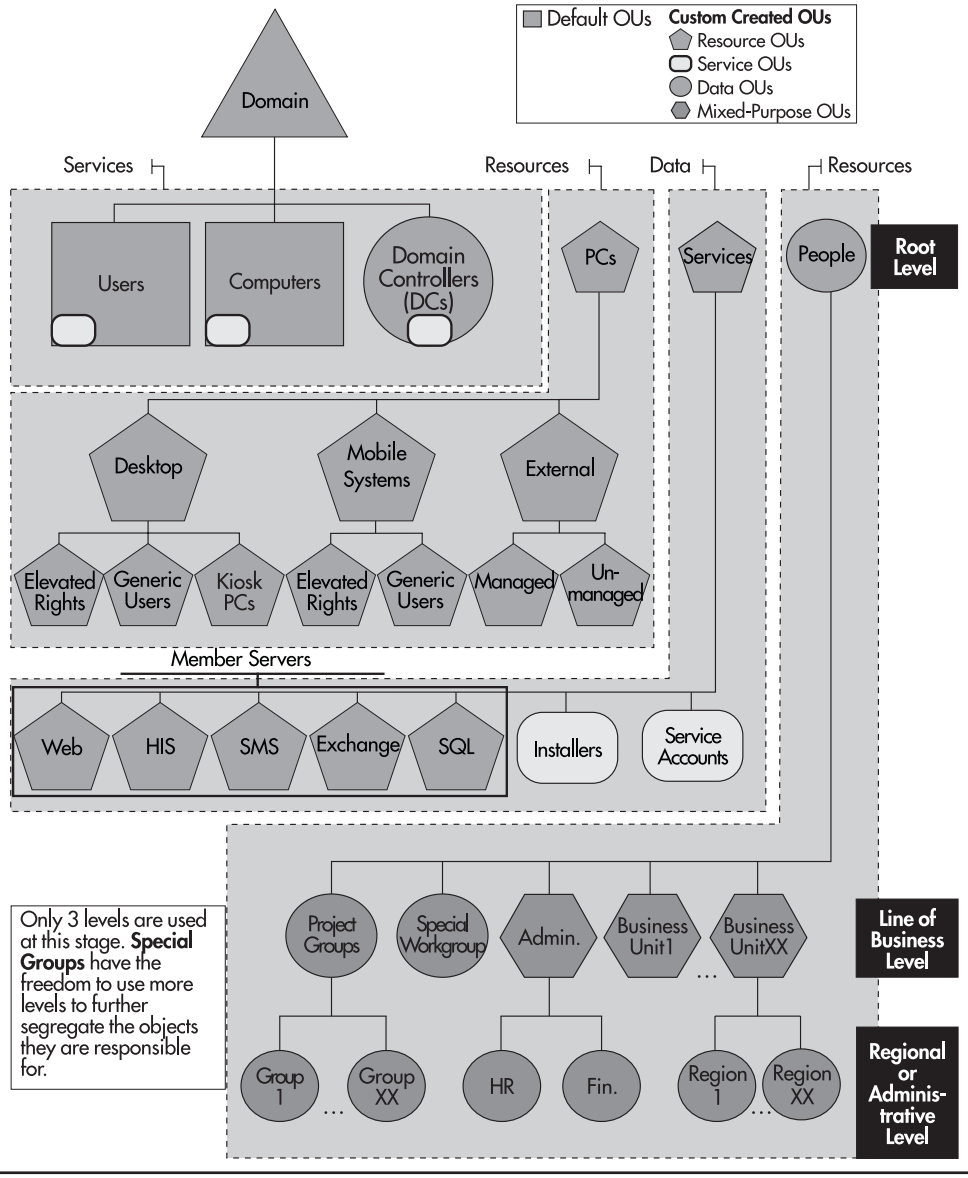


Figure 3-6 A complete production OU structure

- Justify all OUs you create.
- If you find that two OUs have the same purpose, merge them. This means that the combination of owner plus GPO plus delegation strategy is the same for both OUs.

Forest	Domain	PCs OU	Services OU	People OU
Production	Training	One level only, all objects in root	Same basic structure as production	Same as first two levels as in production
	Staging	One level only, all objects in root	Same basic structure as production	One level only, all objects in root
	Development	Same as first two levels as in production	Same basic structure as production	Same as first two levels as in production
	Protected Forest Root	Default OUs only	Default OUs only	Default OUs only
Perimeter	Perimeter	Default OUs only	Default OUs only	Default OUs only
Staging	Protected Forest Root	Default OUs only	Default OUs only	Default OUs only
	Production	Same as production	Same as production	Same as production
Development testing (if required)	Forest root	Default OUs only	Default OUs only	Default OUs only or may require same as production OU for testing
Utility forests	Forest root	Defined as required	Defined as required	Defined as required

Table 3-5 OU Structure in Other Domains

- Use default OUs to administer the whole domain. Domain controllers should be kept in the DC OU. Domain Administrator accounts and groups should be kept in the Users OU. Domain Administrator PCs should be kept in the Computers OU.
- Use the production domain OU strategy to define the OU strategy for other domains and forests.
- Don't forget to define and put in place standards for the recurring creation and deletion of OUs. These will help control the proliferation of OUs in your directory.

Your OU strategy should be based on the information in Table 3-4. While its categorization may differ with the final results of your own Object Categorization Exercise, those differences will be minor. They will vary due to factors such as political situation, business strategy, and IT management approach, rather than because of fundamental differences. Keep in mind that your OU design will not be the answer to every management process in the directory. It is only a first level of object management.

The OU design process should result in the following deliverables:

- An OU hierarchy diagram
- A list of all OUs
- A description of the contents of each OU
- The purpose of each OU
- The identification of each OU owner

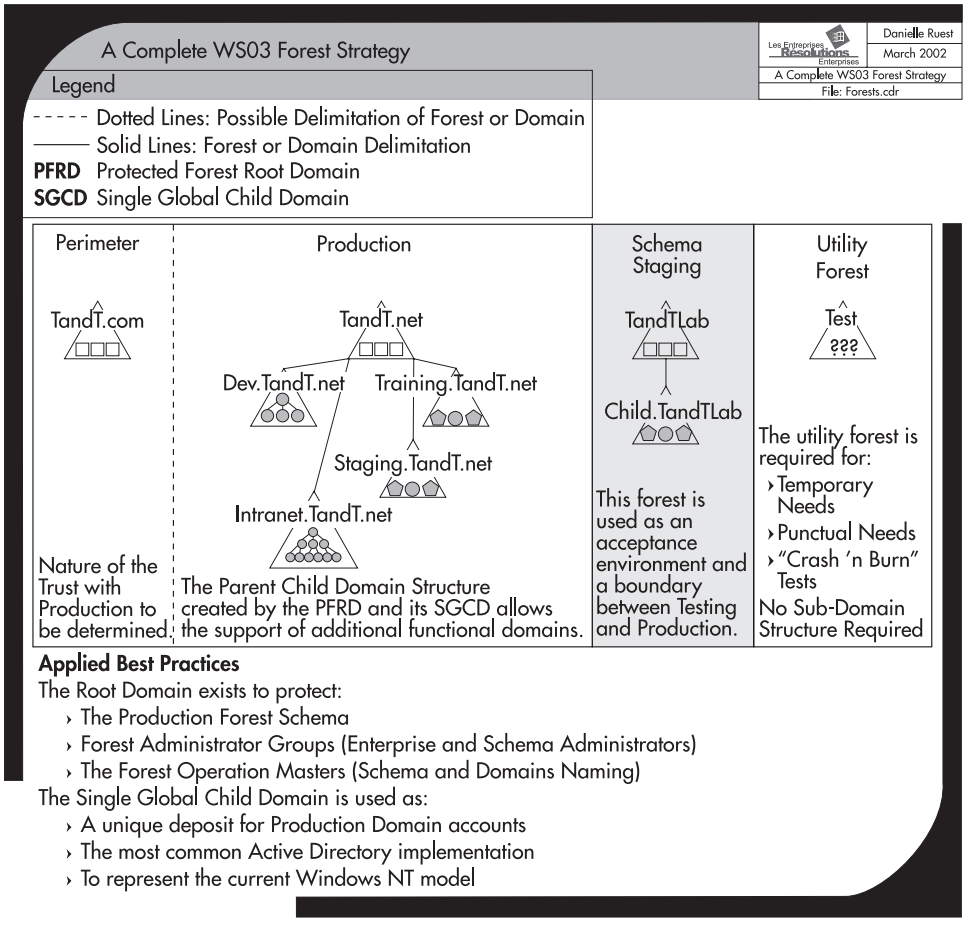


Figure 3-7 The complete Forest, Tree, Domain, and OU design for T&T Corporation

- A list of groups that have control over each OU
- A list of the object types each group can control in each OU
- The rules for the creation and deletion of OUs in regular operations

AD and Other Directories

As you have seen so far, Active Directory is much more than a simple authentication and authorization system. It is a central identity management system. As such, it will interact with other systems on your network, some of which may already hold identity data. Since AD is a central identity management system, new systems you develop can integrate with AD and should not require identity management

components. Microsoft Exchange is an excellent example of this level of integration. In version 5.5, Exchange required its own complete identity management infrastructure. In current versions, it integrates completely with Active Directory and uses AD's functions to manage all identity components.

In this manner, Active Directory is comparable to Windows itself. When programmers prepare software for Windows, they do not need to be concerned with how the application will print or how it will interact with a display device; Windows manages all of these components. The developer only needs to make sure the new code will work with Windows and concentrate on the functions to be built in the application itself. AD provides the same integration features to applications. Application developers no longer need to worry about identity and security management, as AD provides all of these functions. They can now concentrate on richer product-specific features. In addition, they can use AD/AM to integration object extensions. For example, if you want to include a fingerprint hash in your authentication scheme but don't want to modify your NOS directory, you can add it to an AD/AM directory and link it to the NOS directory. This avoids custom schema extensions that must be replicated throughout the enterprise.

In addition, you may already have systems that may not integrate directly with Active Directory, such as human resource systems, custom corporate applications, or third-party software. For each of these systems, you will need to determine which data deposit, the original system or Active Directory, is the primary source for specific data records. For example, if AD can store the entire organizational structure through the information properties you can add to each user account (location, role, manager, and so on), shouldn't AD be the primary source for this information since it is also the primary source for authentication?

These are the types of decisions you need to make when determining how your Active Directory will interact with other directories. Will it be the primary information source? If so, you need to ensure that information is fed into and maintained in the directory. This information feed must be part of your initial AD deployment process. You will also need to consider the changes you must make to your corporate systems so they will obtain primary data from AD, otherwise you will need to maintain several authoritative sources for the same data. If this is the case, you should consider using Microsoft MetaDirectory Services 2003 (MMS).

Microsoft MetaDirectory Services

MMS is a special application that is designed to oversee multiple directory services. MMS manages the operations of several directories to ensure data integrity. If you install MMS over AD and you identify AD as the primary source of information, MMS will automatically modify the values in other directory services when you modify values in AD.

The Standard Edition of MMS is available for free (http://www.microsoft.com/mms/) and is designed to support the integration of data between AD, AD/AM, and Exchange. The Enterprise Edition is designed to integrate heterogeneous data sources. Both run as services on Member Servers and both include simplified deployments. It is important to keep in mind that MMS implementations are additional and separate from AD implementations. But the advantages are clear. If you need to integrate several directories such as in-house databases, third-party software applications, and even other forests, or if you need to integrate AD and AD/AM, MMS is the best way to ensure that data is populated from one information source to all others or automatically synchronize data in multiple deposits. It will also help you manage the employee move, add, and change process since it provides

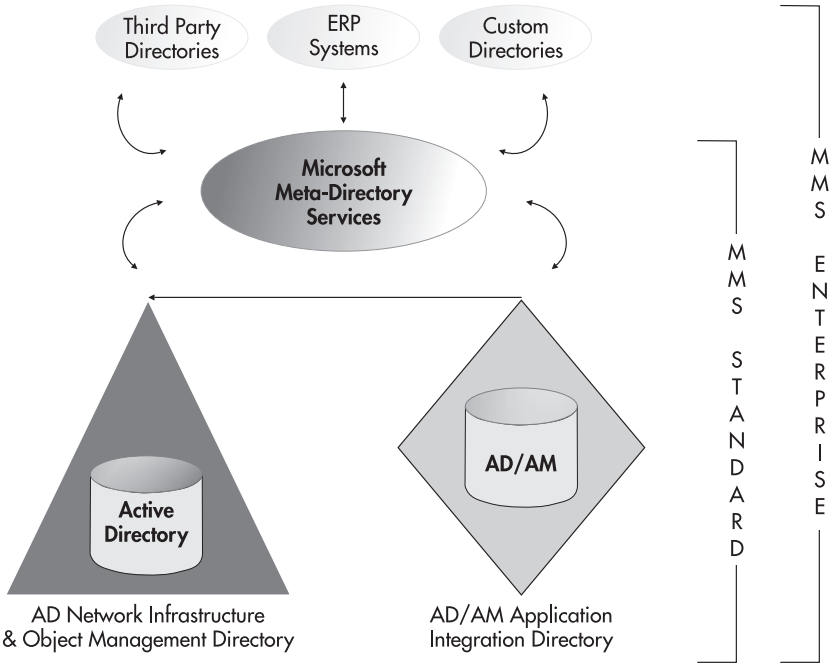


Figure 3-8 Integrating AD, AD/AM, and other directories with MMS 2003

a single, integrated view of all employee data. The integrated operation of MMS, AD, and AD/AM is illustrated in Figure 3-8.

NOTE

For more information on the integration of WS03's directory services, search for "A New Roadmap to Directory Services" at http://www.thedotnetmag.com.

Integrated Applications for NOS Directories

Microsoft has outlined a new application certification program with Windows 2000, the Windows Logo program. This program continues with Windows Server 2003. Logo-approved applications will integrate with Active Directory to use its identity management and authentication capabilities. Today, several applications fall in this category. For a complete and up-to-date list of Logo-certified applications for Windows Server 2003, go to http://www.veritest.com/certified/win2000server/.

Integrating a few applications with the directory is inevitable, especially management or directory extension applications. A good example is the .NET Enterprise Server family. Several of these integrate directly to Active Directory and through this integration bring modifications or extensions to AD database schema. These extensions are necessary because each application adds functionality which is not usually required in a base Active Directory. Among these applications, you will find:

- **Exchange Server** Exchange in fact doubles the size of the AD schema, adding twice the object classes and twice the properties.
- **Internet Security and Acceleration Server** ISA modifies the schema to add special ISA objects. This integration streamlines the security, authentication, and management processes for ISA.
- **Host Integration Server** If you require integrated access between a legacy environment and Windows Server 2003, you will require HIS. HIS also extends the AD schema to streamline HIS management and authentication.
- **Systems Management Server** Version 2003 of Microsoft's enterprise management tool, SMS integrates with AD to extend its network and infrastructure management capabilities.

The reason why it is important to identify how your Active Directory will integrate with other applications or information sources is because of schema extensions. If this is your first implementation of an Active Directory, you should install all schema modifications when you install your forest root domain. In this way, you will limit the amount of replication on your production network. That's right, every time you make a schema modification, it will be replicated to every domain controller in the forest. If you have regional domain controllers that replicate over WAN lines, massive modifications may incur service outages. Extending the schema in the forest root domain, before installing child domains, will contain replication and limit it to the installation process for each server.

In fact, WS03 supports the population of a domain controller from backup media at installation. This means that while you had to build all domain controllers while they were connected to a high-speed network with Windows 2000, in WS03, you will be able to rebuild and repair DCs remotely so long as you have created an offline copy of the directory with the Windows (or another) Backup tool. Domain controllers should still be built in a staging area using a high-speed network during AD deployment if possible.

AD Integration Best Practices

Five activities need to be performed at the AD integration stage:

- Position the Active Directory as the core directory service in the organization.
- Position the role of AD in Application Mode, if required, in your organization.
- Position the relationship other corporate directories will have with AD.
- Identify the interaction model between directory services and position the role Microsoft MetaDirectory Services will play in your organization.
- Determine which operational applications will be integrated in your directory.

Use the following best practices during this process:

- Active Directory should be the core directory service. AD can be modified through a graphical interface. You can also use scripts to perform massive modifications with AD. AD also supports a powerful delegation model. Finally, it supports PC management, something few directory services can perform.

- Use AD as your single point of interaction. AD provides a single point of interaction because it is a distributed database that uses a multimaster replication process. Users can modify data in any regional office and it will automatically be updated through the directory.
- If you need to maintain data integrity between multiple directories, use Microsoft MetaDirectory Services with Active Directory as your primary data source.
- If you need to install NOS-related applications that modify the schema, add them to the forest root domain *before* creating the child domains.
- If you need to integrate in-house applications to the directory, use AD in Application Mode. This will have no impact on your NOS directory.
- Integrate NOS-related and other applications to AD only if it is absolutely required. Schema modifications can be retired and reused, but only through a complex process that will involve replication throughout your distributed NOS directory.
- Maintain your Active Directory as a NOS directory first and foremost. This will limit the amount of replication in the forest and will make it easier to upgrade to future versions of Windows server operating systems.

Service Positioning

Now that you have identified the number of Forests, Trees, and Domains in your Active Directory, designed your OU structure, and identified how the directory service will act in your organization, you can move on to Service Positioning. Service Positioning relates to the position and role domain controllers will have in each forest and each domain. Domain controllers are the core service provider for Active Directory. They provide multimaster replication throughout the entire forest. Some types of information cannot be maintained in a multimaster format. To store and manage this information, some domain controllers have a special role, the Operation Master. Another special role is the Global Catalog; this server supports the research and indexing of forest-wide information. Core Active Directory services fall into three categories: Operation Masters, Global Catalogs, and generic domain controllers. A fourth category must also be considered if the Active Directory is to stay healthy: the DNS server.

Operation Masters Positioning

Operation Masters are AD services that manage requests for specific information changes at either the forest or the domain level. Without these services, AD cannot operate. They fall into two groups: forest-wide and domain-centric Operation Master roles. Operation Master roles are sometimes called flexible single master of operations (FSMO) because even though only a single instance in the forest or the domain can exist, this instance is not rooted to a given server; it can be transferred from one domain controller to another. Thus, it is flexible, and it is single because it must be unique in its scope of influence.

Forest-wide Operation Master roles are:

- **Schema Master** The master service that maintains the structure of the forest database and authorizes schema changes.
- **Domain Naming Master** The master service that controls and authorizes domain naming in the forest.

Only a single instance of these services can exist in the forest at a given time. Both services can be located on the same domain controller if required. In large forests, these services are distributed on two separate domain controllers.

In addition to forest-wide Operation Master roles, there are domain-centric Operation Master roles. If you only have one domain in your forest, you will have a single instance of each of these roles, but if you have more than one domain, every domain will have one instance of each. These include:

- **Relative ID (RID) Master** The master service that is responsible for the assignation of relative IDs to other domain controllers in the domain. Whenever a new object—user, computer, server, group—is created in a domain, the domain controller who is performing the creation will assign a unique ID number. This ID number consists of a domain identification number followed by a relative identification number that is assigned at object creation. When a domain controller runs out of its pool of relative IDs, it requests an additional pool from the RID Master. The relative ID role is also the placeholder for the domain. If you need to move objects between domains in the same forest, you need to initiate the move from the RID Master.
- **Primary Domain Controller (PDC) Emulator** The master service that provides backward compatibility for Windows NT. If there are Windows NT domain controllers or Windows NT network clients in the domain, this server acts as the Primary Domain Controller for the domain. It manages all replication to Backup Domain Controllers (in NT, of course).

  If there are no non-Windows 2000 or XP clients or Windows NT DCs, the forest can operate in native mode. In this case, the PDC Emulator focuses on its two other roles: Time Synchronization on all DCs and Preferential Account Modification Replication to other DCs. All domain controllers in the domain will set their clock according to the PDC Emulator. In addition, any account modification that is critical, such as password modification or account deactivation, will be immediately replicated to the PDC Emulator from the originating server. If a logon attempt fails on a given DC, it checks with the PDC Emulator before rejecting the attempt because it may not have received recent password changes. The PDC Emulator supports two authentication protocols: Kerberos V5 (Windows 2000 and more) and NTLM (Windows NT).
- **Infrastructure Master** The master service that manages two critical tasks:
  - The update of references from objects in its domain to objects in other domains. This is how the forest knows to which domain an object belongs. The Infrastructure Master has a close relationship to the Global Catalog. If it finds that some of its objects are out of date compared to the GC, it will request an update from the GC and send the updated information to other DCs in the domain.

CAUTION

The Global Catalog service and the Infrastructure Master service should not be stored on the same DC unless there is only one server in the forest or the domain is very small (for example, the forest root domain). Problems can arise if they are on the same computer because the Infrastructure Master will share the same database as the Global Catalog. It will not be able to tell if it is out of date or not. It will never request updates. In a large forest, this can cause other DCs to be out of synch with GC contents.

- The update and modification of group members in the domain. If a group includes objects from another domain and these objects are renamed or moved, the Infrastructure Master will maintain the consistency of the group and replicate it to all other domain controllers. This ensures that users maintain access rights even though you perform maintenance operations on their accounts.

These domain-centric master roles should be separated if possible. This depends on the size of each domain. Whatever its size, each domain should have at least two domain controllers for redundancy, load balancing, and availability.

Global Catalog Server Positioning

The Global Catalog server is also a special domain controller role. Any domain controller can operate as a Global Catalog server. The GC is the server that holds a copy of the forest-wide database in each domain. By default, it includes about 20 percent of forest data; everything that has been marked in the forest database schema as having forest-wide interest is published in the GC. A forest with a single DC will automatically include the Global Catalog server role.

The GC has three functions:

- **Find objects** The GC holds information about the users and other objects in your domain. User queries about objects are automatically sent to TCP port number 3268 and routed to the GC server.
- **Allow UPN logons** Users can log onto other domains across the forest by using their user principal name (UPN). If the domain controller validating the user does not know the user, it will refer to the Global Catalog server. Because the GC holds information about every user in the forest, it will complete the logon process if it is allowed by the user's rights. UPN logons are also supported across forests when a forest trust exists.
- **Support Universal groups** All Universal groups are stored in the Global Catalog so they can be available forest-wide.

Native WS03 forests have enhanced GC functionality. For example, they can replicate only Universal group modifications instead of the entire Universal group when changes are made. In addition, native WS03 DCs can cache user's universal membership data, removing the need to constantly consult the GC, so the GC service does not need to be as widespread as in Windows 2000 networks.

The GC service should be widely available, however. If your network spans several regions, you should place at least one GC server per region. In addition, you should enable Universal Group Membership (UGM) Caching for all DCs in the region. Placing the GC server in the region will ensure that Universal group logon requests are not sent over the WAN. The WAN is required for the first logon attempt if no GC is present in the region even if UGM Caching is enabled because the logon DC must locate a GC server from which to cache data. Local GC servers are also useful for applications using port 3268 for authentication requests. Consider potential cross-domain logons when determining where to place GC servers.

Domain Controller Positioning

Positioning both Operation Master roles and Global Catalog servers is positioning domain controllers because each of these services will only operate on a domain controller. As mentioned before, in a single domain forest, all of the FSMO roles and the GC could run on a single DC. But in a medium to large network, these roles are usually distributed among several domain controllers. In addition to performing these roles, Domain controllers support authentication and multimaster replication. This means that the more users you have, the more DCs you will need if you want to keep your login time short. Large multiprocessing servers running the DC service can handle millions of requests a day. Regional servers tend to have several additional functions. Regional servers are often multipurpose servers and they tend to be smaller in capacity than centralized servers. If they are multipurpose servers as well, consider adding additional DCs whenever the user load exceeds 50 users per server.

If some of your regional sites have fewer than ten users, don't place a domain controller in the site. Instead use Terminal Services to create terminal sessions for the users in the closest site containing a DC. All logons will be performed at the remote site. But if you can afford it, place a DC in each site that has more than ten users.

The best way to determine how many DCs to position across your network is to evaluate network performance. In many cases, it is a matter of judgment. Define a rule of thumb based on your network performance and stick to it. You can also predict the number of DCs during the site topology exercise.

DNS Server Positioning

Network performance is exactly the reason why the DNS service is the fourth Active Directory service that needs positioning for optimal directory operations. Since part of the AD structure is based on the Domain Naming System and since all logons must resolve the name and location of a domain controller before being validated, the DNS service has become a core Active Directory service. When positioning services for AD, you will quickly learn that you should marry the DNS service with the domain controller service.

In Windows Server 2003, as in Windows 2000, every domain controller in every domain in every forest should also be a Domain Name Server because AD uses DNS to locate objects in the network and because DNS data can be integrated to the directory. If DNS is configured to integrate with AD, it can also become completely secured. You can ensure that only trusted network objects and sources will update information in the DNS partition of Active Directory. Finally, directory integration means secure replication. Since DNS data is integrated to the directory, it is replicated with other directory information.

DNS data can also be stored in application partitions, directory partitions that can designate which domain controllers are to store the information. For example, in a multidomain forest, WS03 automatically creates a DNS data application partition that spans the entire forest. This means that since the data is replicated to every domain controller in the forest, global forest name resolution will always work everywhere.

Windows 2000 and Windows Server 2003 bring many new concepts to the Domain Naming System. This is why it changes from a simple IP service to become an integrated AD service.

Service Positioning Best Practices

Use the following rules to design your Service Positioning scenario:

- In large AD structures, place the forest-wide Operation Masters in a Protect Forest Root Domain.
- If your forest spans multiple sites, place the Schema Master in one site and the Domain Naming Master in another.
- Carefully protect the access to the Schema Master role.
- Place the RID Master and the PDC Emulator roles on the same DC.
- Create a dedicated PDC Emulator role in domains that have more than 50,000 users.
- Separate Global Catalogs and Infrastructure Masters if you can.
- Place at least two domain controllers per domain.
- If a small domain spans two sites, use at least two domain controllers, one for each site.
- Place a Global Catalog server in each geographic site that contains at least one domain controller.
- Enable Universal group membership caching in each geographic site.
- Place a domain controller wherever there are more than ten users unless the WAN link speed will adequately support remote logon attempts.
- Add a regional domain controller whenever there are more than 50 users per DC, especially if it is a multipurpose server.
- Install the Domain Naming Service on every domain controller.
- Use application partitions to designate DNS replication scopes.

Server Positioning Scenario

The best way to learn how to perform server positioning is to use scenarios. In this scenario, the T&T Corporation endeavors to create and populate its Active Directory. It has more than 10,000 users. It has decided to use a multidomain production forest as displayed in Figure 3-7. Its headquarters are in a single city, but in separate buildings. Both buildings are linked together through a metropolitan area network operating at high speed. In addition, it has fifteen regional offices, some in other metropolitan areas that are of considerable size. In these metropolitan areas, satellite offices use local links to “hop” into the wide area network.

T&T needs to position its domain controllers, Global Catalogs, DNS, and Operation Master roles. Table 3-6 describes the position of each domain in each region. The regional distribution of the organization's offices is illustrated in Figure 3-9.

	Region	Domain	Number of Users
1	HQ Main	Dedicated Root	7
2	HQ Main	Production	3000
3	HQ Main	Development	200
4	HQ Main	Training	300
5	HQ Main	Staging	20
6	HQ Site 2	Production	2200
7	HQ Site 2	Development	250
8	HQ Site 2	Training	200
9	Region 1	Production	250
10	Region 2	Production	300
11	Region 3	Production	100
12	Region 4	Production	125
13	Region 5	Production	2100
14	Region 6	Production	75
15	Region 7	Production	80
16	Region 8	Production	140
17	Region 9	Production	80
18	Region 10	Production	150
19	Region 11	Production	575
20	Region 12	Production	250
21	Region 13	Production	90
22	Region 14	Production	110
23	Region 15	Production	40
24	Satellite 1 (Region 2)	Production	10
25	Satellite 2 (Region 5)	Production	5
26	Satellite 3 (Region 5)	Production	8
27	Satellite 4 (Region 11)	Production	50
28	Satellite 5 (Region 12)	Production	35
Total			**10750**

Table 3-6 Production Forest Server Positioning Scenario Information

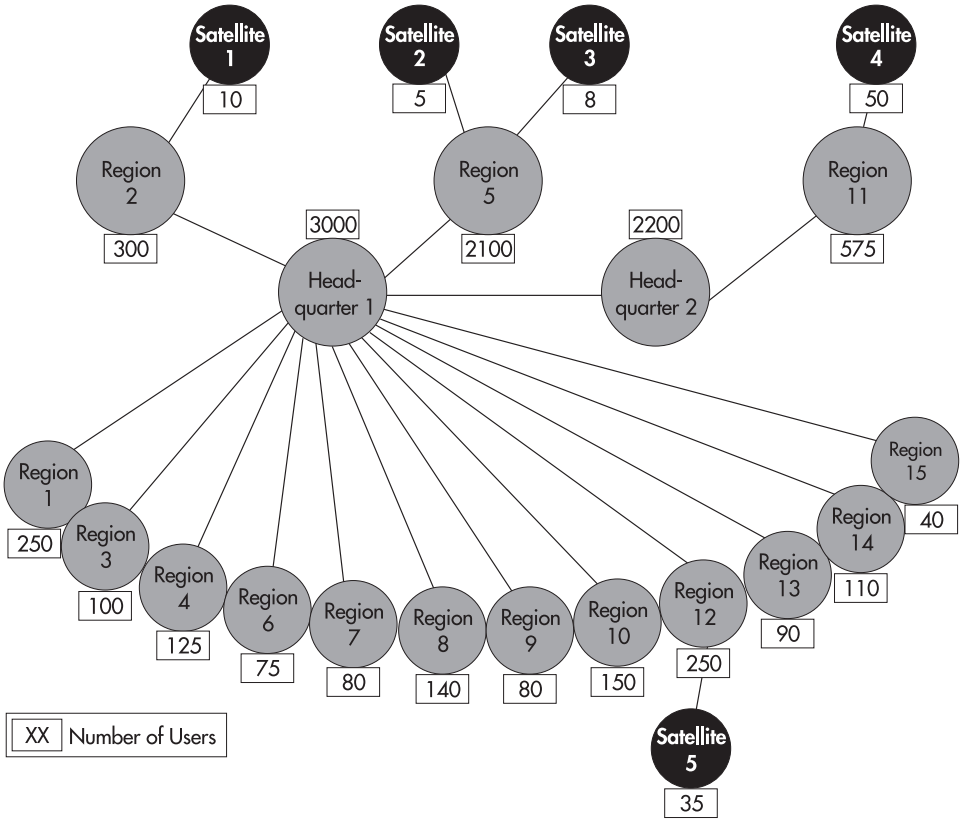


Figure 3-9 T&T Office Locations Map

NOTE

In Table 3-6, development users include the developers themselves as well as test accounts, while users in the training domain only represent generic accounts.

As you can see, the first step for T&T in this phase is to identify the geographical layout of its offices. Once this is identified, T&T can proceed to server positioning. Using the rules outlined earlier, T&T will begin the positioning process. It needs to proceed systematically, thus it will place servers in the following order:

1. The first servers to position are the forest-wide Operation Master roles. These will be in the Protected Forest Root Domain (PFRD): Schema Master and Domain Naming Master.
2. Next will be the PFRD's domain-centric Operation Master roles: RID Master, PDC Emulator, and Infrastructure Master. These should be positioned according to the best practices outlined earlier.

3. The size (number of users) and location of the PFRD will help determine the number of domain controllers required to operate the PFRD.
4. If PFRD DCs are separated physically, the Global Catalog service should be added in each location that includes at least one DC.
5. Next are the child domain DCs. Begin with the production domain because it is the most complex. The first services to position are the domain-centric Operation Master roles: RID Master, PDC Emulator, and Infrastructure Master.
6. Now that the core roles are positioned, position domain controllers. A DC should be positioned in each region with at least 50 users. Regions with more than 50 users should have more than one DC. Regions with less than 50 users should be gauged on an as-needed basis. Also set a rule of thumb for DC positioning in large sites: one DC per 1,000 users (remember, central DCs tend to be more powerful servers than regional DCs).

NOTE

The AD Sizer will tell you that you can manage more than 40,000 users per DC. This may be optimistic because DCs have other roles than simply to manage user logons. Test performance and determine if 1,000 users per DC is appropriate or not in your network.

7. Each region that has at least one DC also hosts at least one Global Catalog service.
8. Next, position Operation Master roles, GCs, and DCs for the other three domains: development, training, and staging. Staging is easy since it is located in a single geographic site; two servers are more than adequate. Training can also perform with two DCs, one in each HQ office. The positioning of development DCs will depend on its level of activity. It is not unusual for development DCs to be used for stress testing analysis. In such situations, the development DC needs to host as many users as the entire production domain.
9. The easiest is kept for the end. Position the DNS service wherever there is a DC.
10. Use application partitions to determine how DNS information should be shared from domain to domain. The result is described in Table 3-7. Keep in mind that the DNS strategy is described in more detail in Chapter 4.

Region	Domain	Users	Servers	Role
HQ Main	Dedicated Root	7	1	1st DC in the forest Forest FSMO: Schema Master Domain FSMO: PDC and RID Global Catalog Integrated DDNS

Table 3-7 T&T Server Positioning Results

Region	Domain	Users	Servers	Role
HQ Site 2	Dedicated Root	7	1	2nd DC in the forest Forest FSMO: Domain Naming Master Domain FSMO: Infrastructure Global Catalog (placing the GC with the Infrastructure Master is okay because of the small number of objects in this domain) Integrated DDNS
HQ Main	Production	3000	3	1st Domain DC Domain FSMO: PDC Global Catalog Integrated DDNS 2nd Domain DC Domain FSMO: RID Integrated DDNS Other DCs DC role only Integrated DDNS
HQ Site 2	Production	2200	3	FSMO Domain DC Domain FSMO: Infrastructure Integrated DDNS GC Domain DC Global Catalog Integrated DDNS Other DCs DC role only Integrated DDNS
Region 1	Production	250	2	GC Domain DC Global Catalog Integrated DDNS Other DCs DC role only Integrated DDNS
Region 2		300	3	
Region 3		100	2	
Region 4		125	2	
Region 5		2100	2	
Region 6		75	2	
Region 7		80	2	
Region 8		140	2	
Region 9		80	2	
Region 10		150	2	
Region 11		575	1	
Region 12		250	2	
Region 13		90	2	
Region 14		110	2	
Region 15		40	1	

Table 3-7 T&T Server Positioning Results *(continued)*

Region	Domain	Users	Servers	Role
Satellite 1 (Region 2)	Production	10	0	N/A
Satellite 2 (Region 5)		5		
Satellite 3 (Region 5)		8		
Satellite 4 (Region 11)	Production	50	1	GC Domain DC Global Catalog
Satellite 5 (Region 12)		35	1	Integrated DDNS
HQ Main	Development	200	1	1st Domain DC Domain FSMO: PDC and RID Global Catalog Integrated DDNS
HQ Site 2	Development	250	1	2nd Domain DC Domain FSMO: Infrastructure Global Catalog Integrated DDNS
HQ Main	Training	300	1	1st Domain DC Domain FSMO: PDC and RID Global Catalog Integrated DDNS
HQ Site 2	Training	200	1	2nd Domain DC Domain FSMO: Infrastructure Global Catalog Integrated DDNS
HQ Main	Staging	20	2	1st Domain DC Domain FSMO: PDC and RID Global Catalog Integrated DDNS 2nd Domain DC Domain FSMO: Infrastructure Global Catalog Integrated DDNS
Total		**10750**	**54**	

Table 3-7 T&T Server Positioning Results *(continued)*

As you can see, the server positioning stage requires the application of a set of rules to the data you have collected on your organization to produce a working result. T&T Corporation, for example, will implement the servers and the roles identified in Table 3-7. They will have two server models, one for regions (multipurpose) and one for large offices (dedicated DC). But they will also need to monitor performance on these servers to ensure that service response times run as expected. If not,

they will need to refine their model. If that is the case, they will need to update their own version of Table 3-7 to ensure that it always reflects reality. The Server Positioning Strategy for T&T Corporation is illustrated in Figure 3-10. For simplicity's sake, this figure only includes the root and production domains.

Another factor that will affect this evaluation is the network speed at which each office is linked with others. Analyzing network speeds and adjusting directory replication is what the next stage, Site Topology Design, is all about.

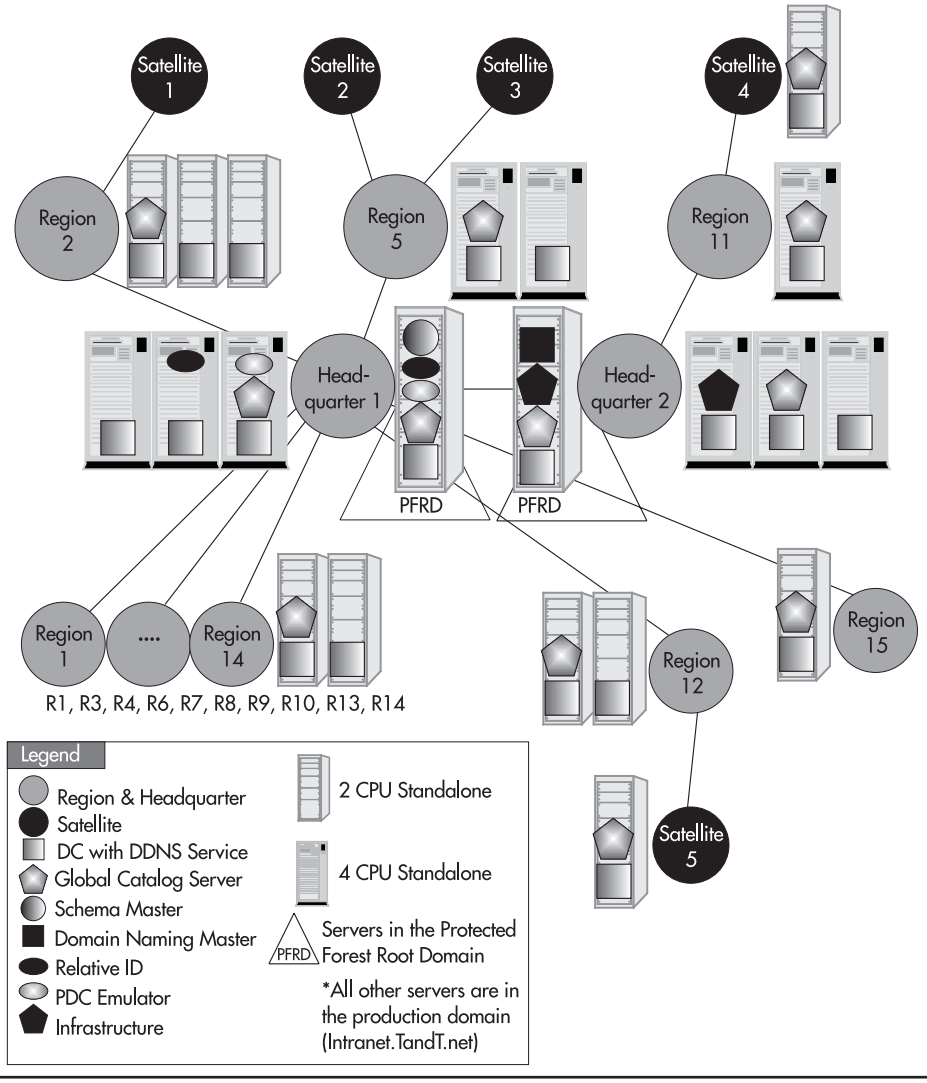


Figure 3-10 Server Positioning Scenario for T&T

Site Topology

The Active Directory design is almost complete; only two further stages are required: Site Topology Design and Schema Modification Strategy. Site Topology Design relates to the analysis of the speed of all WAN links that bind the forest together and the identification of the forest replication strategy. A site is a physical replication partition. Replication is key to proper AD operation.

Windows Server 2003 DCs replicate information on an ongoing basis because they are all authoritative for certain portions of forest information. This multimaster environment requires constant replication if the distributed forest DCs are to be kept up to date. WS03 can perform two types of replication: intra-site and inter-site. Intra-site replication is at high speed because it uses the local area network. Local servers are also often on very high speed links to ensure faster information transfer between them. Intra-site replication occurs constantly because the link speed can support it. Because it is constant and because the link speed can support it, no intra-site replication is compressed.

Inter-site replication is at lower speeds because it must cross a WAN link to other offices. Inter-site replication must be scheduled and compressed, otherwise it will use more than the available bandwidth. The process of creating Active Directory sites is based on the identification of the replication mode between servers. Is it intra- or inter-site replication? A site is also a physical regrouping of servers. A site is usually defined as a TCP/IP subnet. It can be a virtual local area network (VLAN)—a set of network nodes that are grouped together in a single subnet in a geographic location—or a regional subnet. Inter-site replication can occur at 15-minute intervals (it is set at 180 minutes by default). Two transport modes are supported: Internet Protocol (IP) and Simple Mail Transfer Protocol (SMTP). *Never* consider SMTP for inter-site replication! It is more complicated to set up than IP, and it is an asynchronous replication method because changes are sent in discrete messages. It is possible for changes to arrive out of order. Who hasn't sent an email message to someone only to have it come back a week later telling you the person never received it? You can't take the chance that this will happen with directory replication data.

IP uses the Remote Procedure Call (RPC) to send changes to other DCs. It uses the Knowledge Consistency Checker (KCC) service to determine automatic routes between replication partners. For this to occur between sites, a Site Link must be created between each site that contains a domain controller. This Site Link includes costing information. The KCC can use this information when determining when to replicate, how to replicate, and the number of servers to replicate with. Special values such as password changes or account deactivations are replicated immediately to the PDC Emulator in the domain despite site-specific schedules. Inter-site replication data is also compressed. AD compresses replication data through a compression algorithm. Data is automatically compressed whenever it reaches a certain threshold. Usually anything greater than 50 KB will automatically be compressed when replicated between sites.

In a native WS03 forest, you should enable linked value replication. This option greatly reduces replication by sending only the values that have changed for any multivalued attribute such as groups. Whenever a change is made to a group member such as a new member addition, only the changed value (the new member) is replicated instead of the entire attribute.

Site Topology Design

To perform Site Topology Design, you need the following elements:

- A map of all site locations.
- The WAN topology and link speeds for each location. Router configuration is also important. TCP/IP ports that are required for replication are often closed by default. These ports are identified in Chapter 4.
- The number of DCs in each site.

Site design is simple: it should follow the enterprise TCP/IP network design. Sites are IP subnets, thus they are the same as structures you already have in place for TCP/IP. When you proceed with the design, it will result in the creation of:

- Site boundaries for each geographic location
- Site replication links
- Backup replication links
- Costing scheme for each link

Sites are independent of the domain structure. This means that you could have multiple domains in a site, multiple sites in a domain, as well as multiple sites and multiple domains in a wide area network.

Forest replication is divided into three categories: forest-wide, application partition, and domain-centric replication. Both forest-wide and application partition replication span domains. Fortunately, the data replicated through these partitions is relatively small. This is not the same for domains. Production domains especially contain vast amounts of information. This is the core reason for Site Topology Design: data availability between separate domain sites.

Production domains should be split if they must replicate over link speeds of 56 kilobits per second or lower. Very large production domains require high speed WAN links if they are to span regional offices even though data is compressed and replication is scheduled. If vast amounts of data must be sent, the "pipeline" sending it must be big enough for the time allowed. In very large sites with low-speed links, it is possible to have a situation where replication never completes because the replication window opens at intervals that are shorter than the time it takes to replicate all changed data. This is a good opportunity to use the AD Sizer.

Site Link routes should resemble the basic IP structure of your WAN. The cost of each link should reflect the link speed; the lower the cost, the higher the speed. Lower costs also mean faster replication. Table 3-8 identifies sample link costs for given bandwidths.

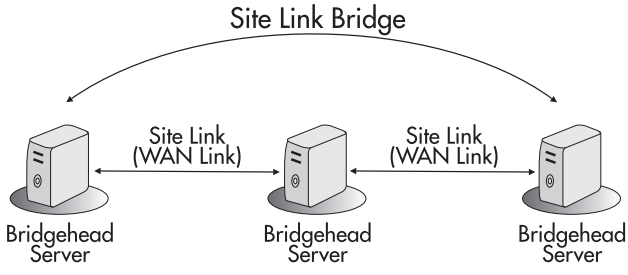
Creating Site Link Bridges

In some cases, it is necessary to bridge replication. If you create Site Links that overlap, you should create a Site Link Bridge. This will allow the replication to use the bridging site to create a direct connection to the destination site. If you want to further control inter-site replication in given sites, you can designate Preferred Bridgehead Servers at the site. The Bridgehead Server manages all

Available Bandwidth	Suggested Cost for Prime Link	Suggested Cost for BU Link
56	Separate domain	N/A
64	750	1000
128	500	750
256	400	500
512	300	400
1024	150	300
T1	100	150

Table 3-8 Recommended Link Cost per Available Bandwidth

inter-site replication in a site. All updates are received and sent through the Bridgehead Server. Thus no other DCs in the site need dedicate resources to inter-site replication. On the other hand, if you designate Bridgehead Servers, the Knowledge Consistency Checker will no longer be able to calculate replication routes automatically. You will have to monitor replication closely to ensure that all sites are up to date.



It is a good idea to calculate replication latency—the time between a modification on a DC and the reception of the modification on all other DCs—in the site topology. This will allow you to identify what the longest possible replication delay can be within your network. Replication latency is calculated with the replication interval, the time it can take to replicate data, and the number of hops required to perform replication. For example, if your site topology includes two hops, your replication interval is set at 180 minutes, and it takes 30 minutes to complete a replication change, your replication latency is 420 minutes (180 times 2, plus 30 minutes times 2). Also, remember to base all your replication calculations on available bandwidth, not global bandwidth. If only 10 percent of bandwidth is available for AD replication, it will affect your calculations.

Finally, as mentioned before, the Univeral Group Membership Caching option is assigned to sites in a native WS03 AD forest. This option should be set for all sites. DCs will be able to cache requesting users' universal group memberships, reducing the amount of communications with central Global Catalog servers.

Best Practices for Site Topology Design

Use the following best practices to design your site topology:

- Use the default configuration for inter-site replication.
- Do not disable the Knowledge Consistency Checker.
- Do not disable transitive trusts.
- Do not specify Bridgehead Servers.
- Calculate replication latency between sites.
- Create sites according to network topology; Site Links and WAN links should correspond.
- Make sure that no single site is connected to more than 20 other sites.
- Each site must host at least one DC.
- Do not use SMTP for domain-centric replication.
- Do not use SMTP replication if at all possible.
- Use 128 Kbps as the minimum WAN circuit for a Site Link.
- Associate every site with at least one subnet and one Site Link, otherwise it will be unusable.
- Create backup Site Links for each site. Assign higher costs to backup Site Links.
- Create Site Link Bridges wherever there are two hops between sites to reduce replication latency.
- If your available network bandwidth can afford it, ignore replication schedules in all sites. Replication will be performed when required with this option, but it will be more demanding on WAN bandwidth.
- Enable Universal Group Membership Caching in all sites.
- Use Preferred Bridgehead Servers only if replication must cross a firewall.
- Size your DCs accordingly.
- Monitor replication once your forest is in place to determine the impact on your WAN links.

T&T Corporation's Site Topology Scenario

T&T's site topology is based on the information displayed previously in Figure 3-9 as well as the WAN Link Speed for each site. Using this information, T&T produced the grid outlined in Table 3-9.

NOTE

The perimeter forest is also identified in Table 3-9 and in Figure 3-11 to demonstrate the potential use of Bridgehead Servers.

T&T used some global settings in their Site Topology Design. These include:

- Open schedules for all sites.
- KCC on by default in all sites.

- All Site Link costs decrease as they get closer to HQ1, so HQ1 replication is prioritized.
- Replication is only performed with the RPC through IP.
- Default schedules are enabled in all sites (replication every 180 minutes).
- High priority replication can occur immediately.
- Every site has a backup replication route at a higher cost.

Site Link Name	Link Speed to HQ	Site Link Type	Site Link Cost	Options
HQ Main	LAN	VLAN	1	Site Link available (VLAN for server connections) KCC on (setting for all sites) Site Links with all sites Site Link Bridge with S5 and R11
HQ Main to Security Perimeter Security Perimeter to HQ Main	LAN with Firewall	VLAN	50	Preferred Bridgehead Server
HQ Site 2 Region 5	T1	VLAN	100	Site Links with HQ1 and R11 BU Site Links with all sites Site Link Bridge with S4
Region 1 Region 3 Region 4 Region 6 Region 7 Region 8 Region 9 Region 10 Region 13 Region 14	256	Regional	400	Site Link with HQ1 BU Site Link with HQ2
Region 2 Region 12	512	Regional	300	Site Link with HQ1 BU Site Link with HQ2
Region 11	T1	VLAN	150	Site Link with HQ2 Site Link Bridge with HQ1 BU Site Link with HQ1
Region 15	128	Regional	500	Site Link with HQ1 BU Site Link with HQ2
Satellite 1 (Region 2) Satellite 2 (Region 5) Satellite 3 (Region 5)	64	N/A	N/A	N/A
Satellite 4 (Region 11) Satellite 5 (Region 12)	128	Regional	500	Site Link with R11 Site Link Bridge with HQ2 BU Site Link with HQ2

Table 3-9 T&T Site Topology

- Everything is based on calculated available bandwidth.
- Every site is set to cache universal group memberships.
- Firewall replication is controlled through preferred Bridgehead Servers.

Of course, T&T will need to monitor AD replication performance during the operation of the directory to ensure that the values in this table are appropriate to meet service levels. If not, both the table and the Site Links will need to be updated. This Site Topology Design for T&T Corporation is illustrated in Figure 3-11.

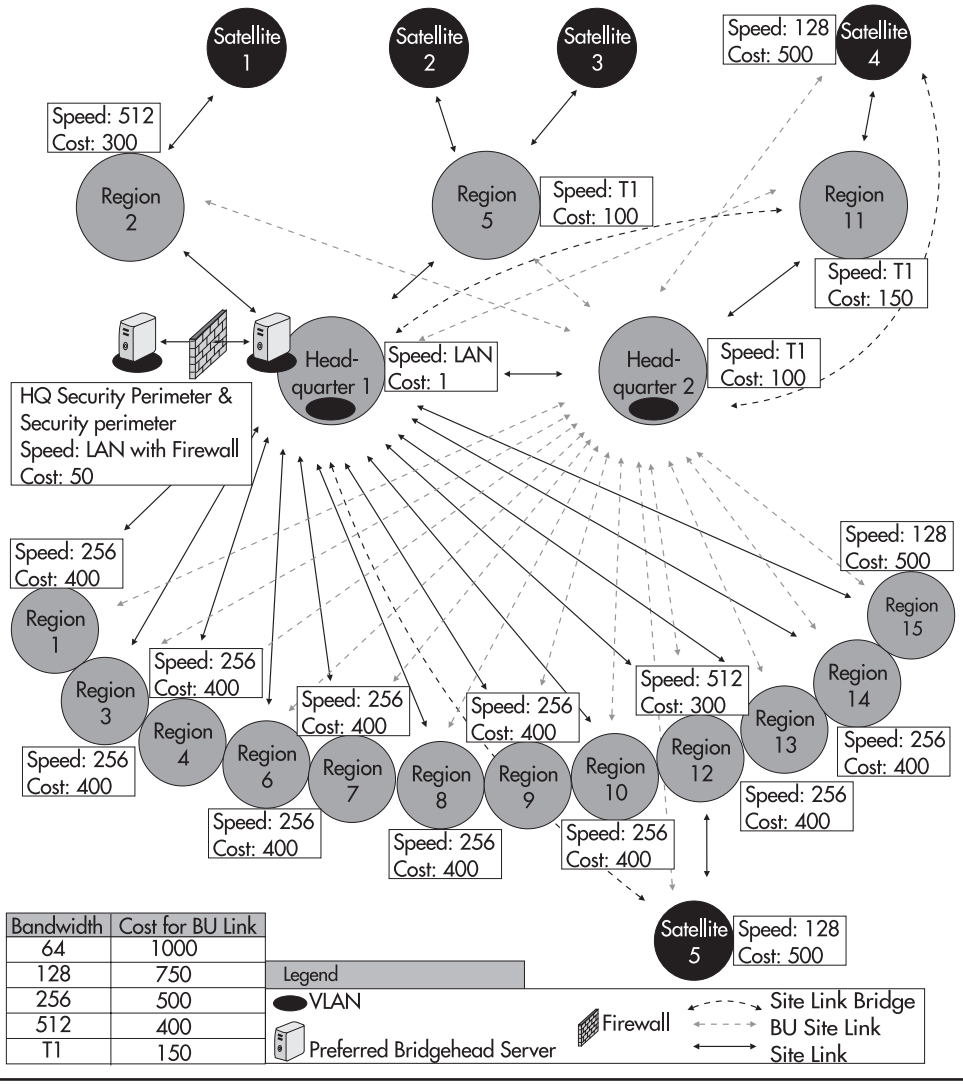


Figure 3-11 T&T's Site Topology Design

Schema Modification Strategy

Now that your forest design is done, you can put it in place. The final process you need to complete is the outline of your Schema Modification Strategy. Operating an Active Directory is managing a distributed database. Modifying the structure of that database has an impact on every service provider in the forest. Adding object classes or object class attributes must be done with care and in a controlled manner. Adding components always implies added replication at the time of the modification. It may also mean added replication on a recurring basis. Retiring components also implies added replication at the time of modification, though it may also mean reduced ongoing replication. Native Windows Server 2003 forests support the reuse of certain types of deactivated object classes or attributes.

Expect your AD database schema to be modified. Even simple tools such as enterprise backup software will modify the schema to create backup objects within the directory. Without a doubt, some of the commercial server tools you acquire—be they only Microsoft Exchange—will modify your production AD schema.

In addition, you may also want to take advantage of schema extensions for your own purposes. You will definitely shorten application development timelines if you choose to use the directory to store frequently requested information. AD will automatically replicate information throughout your enterprise if it is part of the directory. Be careful what information you include in the directory. Because of its multimaster and hierarchical models, AD is not designed to provide immediate data consistency. There is always replication latency when more than a single DC is involved. Use the directory to store static information that is required in every site, but is unlikely to change very often. You may also decide that you do not want to modify the schema for your own purposes. The arrival of AD/AM with WS03 means that AD can now be solely used as a NOS directory. This is the recommended approach. It will make it simpler to upgrade your directory when the next version of Windows comes out.

However you decide to use your directory, one thing is sure, you must always be careful with schema modifications within the production directory. The best way to do so is to form a Schema Modification Policy. This policy is upheld by a Schema Change Policy Holder (SCPH) to whom all schema changes are presented for approval. The policy will outline not only who holds the SCPH role, but also how schema modifications are to be tested, prepared, and deployed. Assigning the SCPH role to manage the schema ensures that modifications will not be performed on an ad hoc basis by groups that do not communicate with each other.

In addition, the X.500 structure of the AD database is based on an object numbering scheme that is globally unique. A central authority, the International Standards Organization (ISO), has the ability to generate object identifiers for new X.500 objects. Numbers can also be obtained from the American National Standards Institute (ANSI). X.500 numbering can be obtained at http://www.iso.org/ or http://www.ansi.org/. Microsoft also offers X.500 numbering in an object class tree it acquired for the purpose of supporting Active Directory. You can receive object IDs from Microsoft by sending email to oids@microsoft.com. In your email, include your organization's naming prefix and the contact name, address, and telephone number. To obtain your organization's naming prefix, read the Active Directory portion of the Logo standards at http://www.microsoft.com/winlogo/downloads/software.asp.

Object identifiers are strings in a dot notation similar to IP addresses. Issuing authorities can give an object identifier on a sublevel to other authorities. The ISO is the root authority. The ISO has a number of 1. When it assigns a number to another organization, that number is used to identify that organization. If it assigned T&T the number 488077, and T&T issued 1 to a developer, and that developer assigned 10 to an application, the number of the application would be 1.488077.1.10.

To create your Schema Modification Strategy, you need to perform three steps:

- Identify the elements of the Schema Modification Policy.
- Identify the owner and the charter for the Schema Change Policy Holder role.
- Identify the Schema Change Management Process.

The Schema Modification Policy includes several elements:

- List of the members of the Universal Enterprise Administrators group.
- Security and management strategy for the Universal Schema Administrators group. This group should be kept empty until modifications are required. Members are removed as soon as the modification is complete.
- Creation of the SCPH role.
- Schema Change Management Strategy documentation including:
  - Change request supporting documentation preparation with modification description and justification.
  - Impact analysis for the change. Short term and long term replication impacts. Costs for the requested change. Short term and long term benefits for the change.
  - Globally unique object identifier for the new class or attribute, obtained from a valid source.
  - Official class description including class type and location in the hierarchy.
  - System stability and security test results. Design standard set of tests for all modifications.
  - Modification recovery method. Make sure every modification proposal includes a rollback strategy.
  - Schema write-enabling process. By default, the schema is read-only and should stay so during ongoing production cycles. It should be reset to read-only after every modification.
- Modification Authorization Process; meeting structure for modification recommendation.
- Modification Implementation Process outlining when the change should be performed (off production hours), how it should be performed, and by whom.
- Modification report documentation. Did the modification reach all DCs? Is replication back to expected levels?

This process should be documented at the very beginning of your implementation to ensure the continuing integrity of your production schema. If this is done well, you will rarely find your staff performing midnight restores of the schema you had in production yesterday.

Schema Modification Strategy Best Practices

Use the following schema modification best practices:

- Don't make your own modifications to the schema unless they are absolutely necessary.
- Use AD primarily as a NOS directory.
- Use AD/AM to integrate applications.
- Use MMS 2003, Standard Edition to synchronize AD and AD/AM directories.
- Make sure all commercial products that will modify the schema are Windows Server 2003 Logo approved.
- Limit your initial modifications to modifications by commercial software.
- Create a Schema Change Policy Holder role early in the AD Implementation Process.
- Document the Schema Modification Policy and Process.

AD Implementation Plan

The first stage of AD preparation is complete. You have designed your AD strategy. Now you need to implement the design. To do so, you require an AD Implementation Plan. This plan outlines the AD migration process. Basically, this plan identifies the same steps as the design process, but is focused only on those that deal with implementation. It is reduced to four major steps:

- Forest, Tree, and Domain Installation
- OU and Group Design
- Service Positioning
- Site Topology Implementation

Once these four steps are complete, your AD will be in place. These four steps are outlined in Figure 3-12 through the AD Implementation Blueprint.

This blueprint is designed to cover all the major steps in a new AD implementation. It uses the parallel network concept outlined in Chapter 2 to create a separate new network that can accept users as they are migrated from the existing production network. Because the AD Implementation Process is closely tied to the design of the IP network, the deployment of a new Active Directory and the IP network infrastructure are covered together in Chapter 4. If you already have a Windows 2000 AD in place, however, you are more likely to use the upgrade process outlined at the end of Chapter 4.

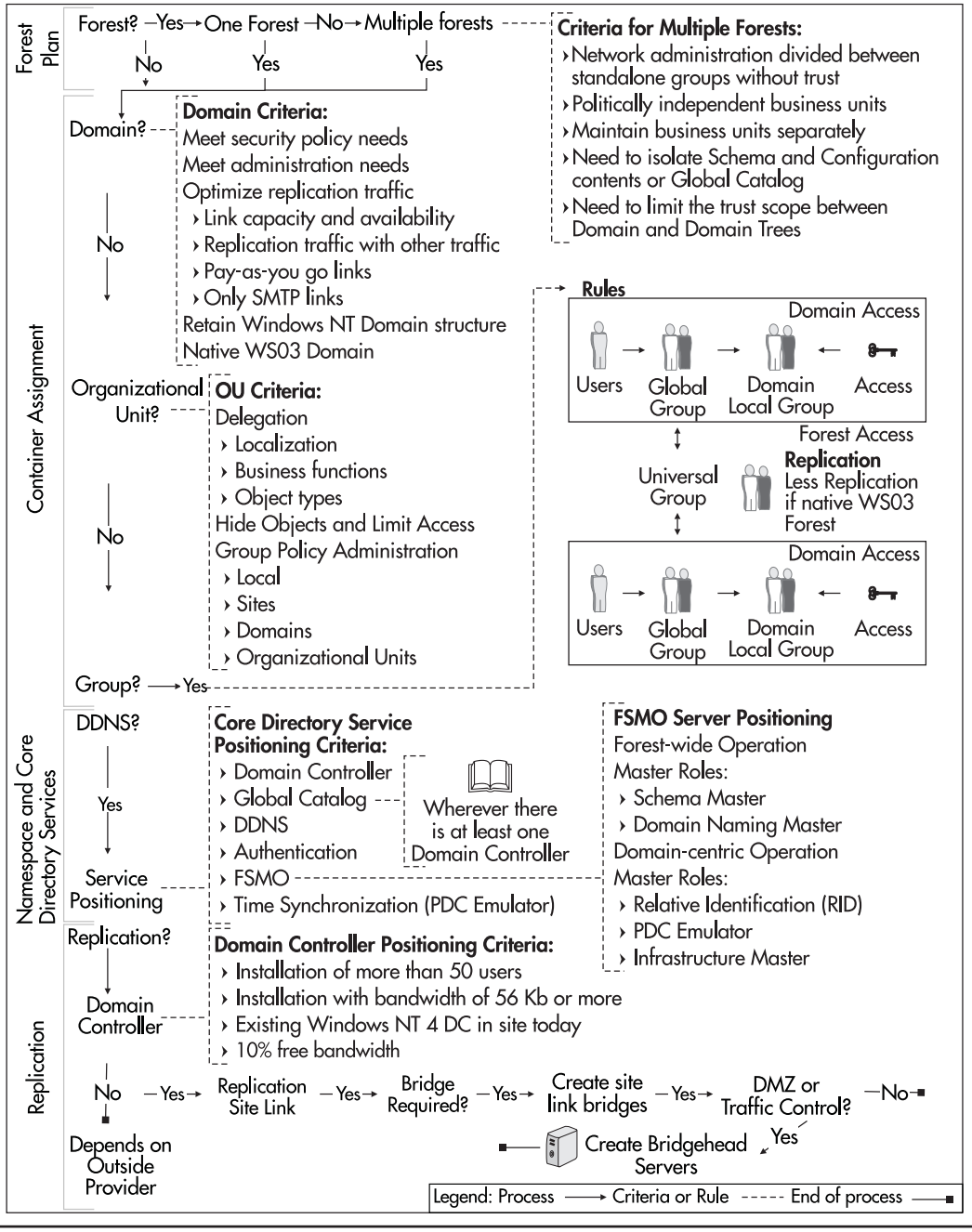


Figure 3-12 The AD Implementation Blueprint

The Ongoing AD Design Process

In summary, the AD Design Process is complex only because it includes a lot more stages than the Windows NT design. One of the things you need to remember is that creating a production AD is creating a virtual space. Since it is virtual, you can manipulate and reshape it as your needs and comprehension of Active Directory evolve. WS03 makes this even easier by supporting drag and drop functionality in the AD Management Consoles: Active Directory Users and Computers, Active Directory Domains and Trusts, and Active Directory Sites and Servers. WS03 also supports multiple object attribute changes—for example, if you need to change the same attribute on several objects.

Also, a tool that is very useful in the Active Directory Design Process is Microsoft Visio Professional, especially the version for Enterprise Architect. In fact, you can actually draw and document your entire forest using Visio. Once the design is complete, it can be exported and then imported into Active Directory. Microsoft offers a complete step-by-step guide to this task at http://www.microsoft.com/technet/treeview/default.asp?url=/TechNet/prodtechnol/visio/visio2002/deploy/vsaddiag.asp.

These tools can only *assist* you in the design process. The success or failure of the Active Directory Design Process you will complete will depend entirely on what your organization invests in it. Remember, AD is the core of your network. Its design must respond to organizational needs. The only way to ensure this is to gather all of the AD stakeholders and get them to participate in the design process. In other words, the quality of the team you gather to create your AD design will greatly influence the quality of the output you produce.

Best Practice Summary

This chapter is chock-full of best practices. It would be pointless to repeat them here. One final best practice or recommendation can be made: Whatever you do in your Windows Server 2003 migration, make sure you get the Active Directory part right! It must be designed properly if you want to meet all of the objectives of a migration to WS03.

Chapter Roadmap

Use the illustration in Figure 3-13 to review the contents of this chapter.

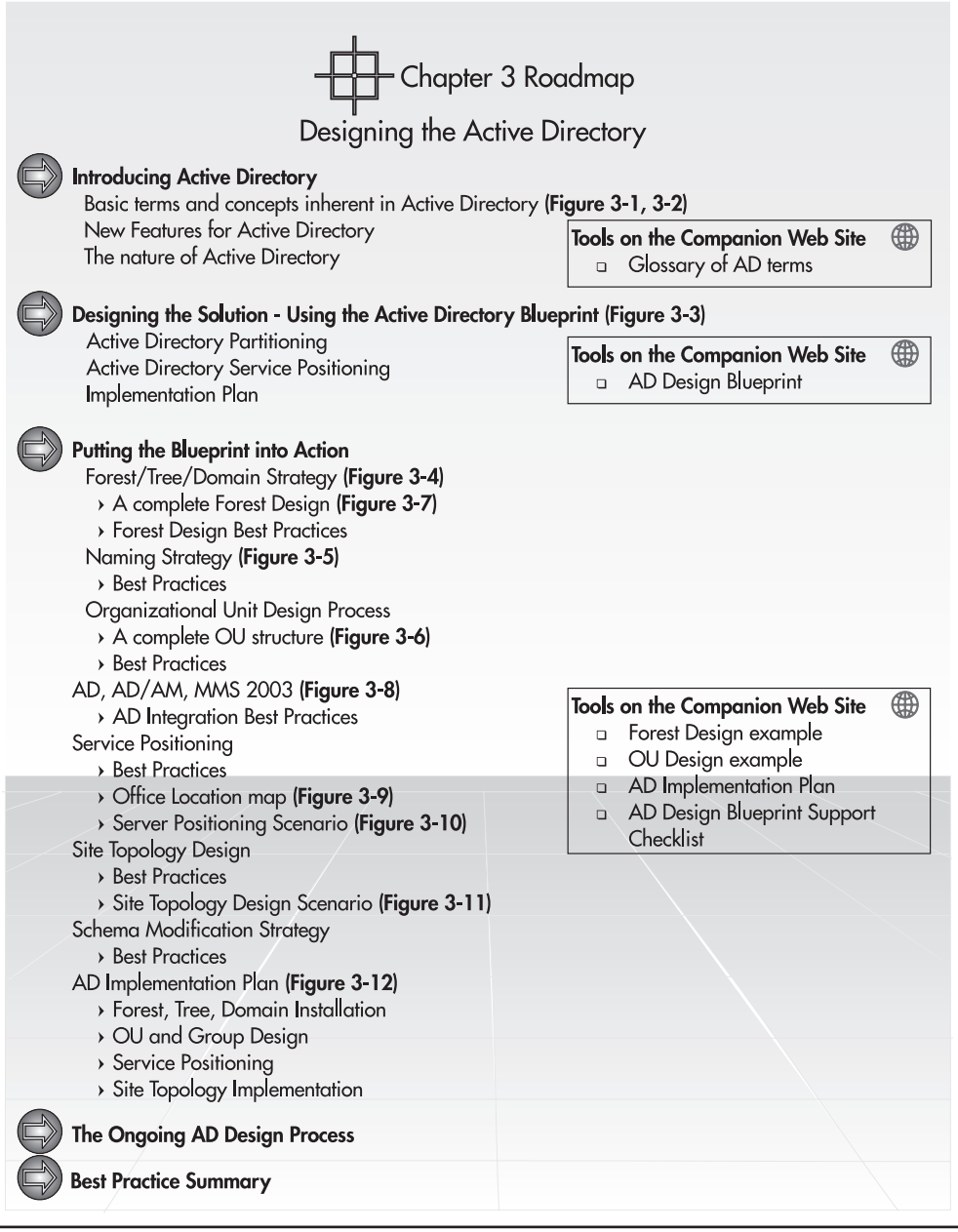


Figure 3-13 Chapter Roadmap