

Designing the Active Directory Structure



Microsoft® Windows® 2000 Server includes a directory service called *Active Directory™*. The Active Directory concepts, architectural elements, and features presented in this chapter will help the IT architect and strategic planner in your organization to produce design documents essential to a successful Microsoft® Windows® 2000 Active Directory deployment.

Prior to reading this chapter, it is important that you obtain detailed knowledge of the IT administration groups, administrative hierarchy, and network topology in your organization. This knowledge will help you apply the planning guidelines in this chapter to your own unique environment.

In This Chapter

- Overview of Active Directory 255
- Planning for Active Directory 257
- Creating a Forest Plan 261
- Creating a Domain Plan 268
- Creating an Organizational Unit Plan 295
- Creating a Site Topology Plan 305
- Planning Task List for Designing the Active Directory Structure 315

Chapter Goals

This chapter will help you develop the following planning documents:

- Forest Plan
- Domain Plan for each forest
- Organizational Unit (OU) Plan for each domain
- Site Topology Plan for each forest

Related Information in the Resource Kit

- For more information about migrating domains to Windows 2000, see “Determining Domain Migration Strategies” in this book.
- For more information about Windows 2000 security standards, such as the Kerberos protocol, see “Planning Distributed Security” in this book.
- For more information about advanced networking, see “Determining Network Connectivity Strategies” in this book.
- For more information about Microsoft® IntelliMirror™ or Group Policy, see “Applying Change and Configuration Management” in this book.
- For more technical information about Active Directory, see the *Microsoft® Windows® 2000 Server Resource Kit Distributed Systems Guide*.
- For more information about Domain Name System (DNS), see “Introduction to DNS” and “Windows 2000 DNS” in the *Microsoft® Windows® 2000 Server Resource Kit TCP/IP Core Networking Guide*.

Overview of Active Directory

Active Directory plays many roles, from being the backbone of distributed security to providing a service publishing framework. Active Directory provides a central service for administrators to organize network resources, to manage users, computers, and applications; and to secure intranet and Internet network access.

As an increasing number of distributed applications take advantage of Active Directory, you can benefit by not having to implement and manage application-specific directory services. The result is that you save administrative and hardware costs.

Note You can deploy Windows 2000 Server and Microsoft® Windows® 2000 Professional before, in parallel with, or after Active Directory. It is not necessary to deploy Active Directory first. You can take advantage of many of the new features in Windows 2000 by upgrading member servers and client computers right away. For more information about upgrading member servers, see “Upgrading and Installing Member Servers” in this book.

Primary Active Directory Features

Windows 2000 Active Directory features offer many advantages for your network, including the following:

Security

Active Directory provides the infrastructure for a variety of new security capabilities. Using mutual authentication, clients can now verify the identity of a server before transferring sensitive data. Using public key security support, users can log on using smart cards instead of passwords.

Simplified and Flexible Administration

Objects in the Active Directory have per-attribute access control, which allows fine-grained delegation of administration. Delegation of administration allows you to more efficiently distribute administrative responsibility in your organization, and reduce the number of users that must have domain-wide control.

Scalability

Active Directory uses the *Domain Name System (DNS)* as a locator mechanism. DNS is the hierarchical, distributed, highly scalable namespace used on the Internet to resolve computer and service names to Transmission Control Protocol/Internet Protocol (TCP/IP) addresses.

The directory stores information using *domains*, which are partitions that let you distribute the directory over a large network of varying speed and reliability. The directory uses database technology and has been tested to accept millions of *objects* (users, groups, computers, shared file folders, printers, and more). This combination of scalable locator, partitioning, and scalable storage ensures that the directory scales gracefully as your organization grows.

High Availability

Traditional directories with single master replication offer high availability for query operations, but not update operations. With multimaster replication, Active Directory offers high availability of both query and update operations.

Extensibility

The schema, which contains a definition for every object class that can exist in a directory service, is extensible. This allows both administrators and software developers to tailor the directory to their needs.

Open Standards Support

Active Directory is built on standards-based protocols such as:

- DNS, for locating servers running Active Directory.
- *Lightweight Directory Access Protocol (LDAP)* as a query and update protocol.
- The *Kerberos* protocol for logging on and authentication.

This support for open standards makes it possible to use a wide variety of software with Active Directory, such as LDAP-based address book clients.

Simple Programmatic Access

The Active Directory Service Interfaces (ADSI) are accessible from a variety of programming platforms, including script languages such as Visual Basic Script. When using ADSI, administrators and software developers can quickly create powerful directory-aware applications. An example of a directory-aware application is an application that reads the directory for data or configuration information.

Providing a Foundation for New Technologies

In addition to the fundamental advantages previously discussed, Active Directory plays an important role in your Windows 2000 deployment as an enabling infrastructure for other new technologies and capabilities, such as the following:

IntelliMirror

Windows 2000 provides a variety of Change and Configuration Management technologies. IntelliMirror and Remote Operating System Installation Management can help you reduce the amount of work and costs associated with managing and supporting clients. For more information about implementing these technologies, see “Applying Change and Configuration Management” and “Defining Client Administration and Configuration Standards” in this book.

Directory Consolidation

The scalability and extensibility of Active Directory makes it an ideal point of consolidation for applications on your network that use separate, internal directories. For example, you can:

- Have complete directory consolidation, where products like Microsoft® Exchange Server shed the directory components and rely solely on Active Directory for administration and operation.
- Consolidate administration, where you manage directory information in Active Directory and use directory synchronization to keep remote directories up to date.
- Consolidate your existing Microsoft® Windows NT® domains, potentially reducing the total number of objects and hardware to be managed on your network.

Advanced Networking

Internet Protocol security (IPSec), networking Quality of Service features, and new remote access capabilities are examples of advanced networking features that are enabled by Active Directory.

Planning for Active Directory

When you plan for and deploy your enterprise-scale Active Directory, you are defining a significant part of the network infrastructure of your organization. In this plan, you create a set of structures that best reflects your organization. The structures you create will determine:

- The availability and fault tolerance of the directory.
- The network usage characteristics of directory clients and servers.
- How efficiently you can manage the contents of the directory.
- The way users view and interact with the directory.
- The ability of your directory structures to evolve as your organization evolves.

Having a well thought-out Active Directory plan is essential to a cost-effective deployment. Investing time in the planning phase will help you avoid spending time and money in the future reworking structures that you have already put in place. To create your directory structure plans, follow the sequence of planning steps as presented in this chapter. While you create your plans:

- Learn the key Active Directory concepts that influence structure planning, and adjust the suggested planning steps as necessary to best suit your organization.
- Identify the people in your organization who should participate in structure planning.
- Understand how existing business practices might need to change or evolve to take full advantage of Active Directory.
- Understand the flexibility of the structures you create, and realize which of your choices will be easy to change or hard to change in the future.

Figure 9.1 illustrates the primary steps for designing the Active Directory structure. This chapter will take a close look at each one of these steps.

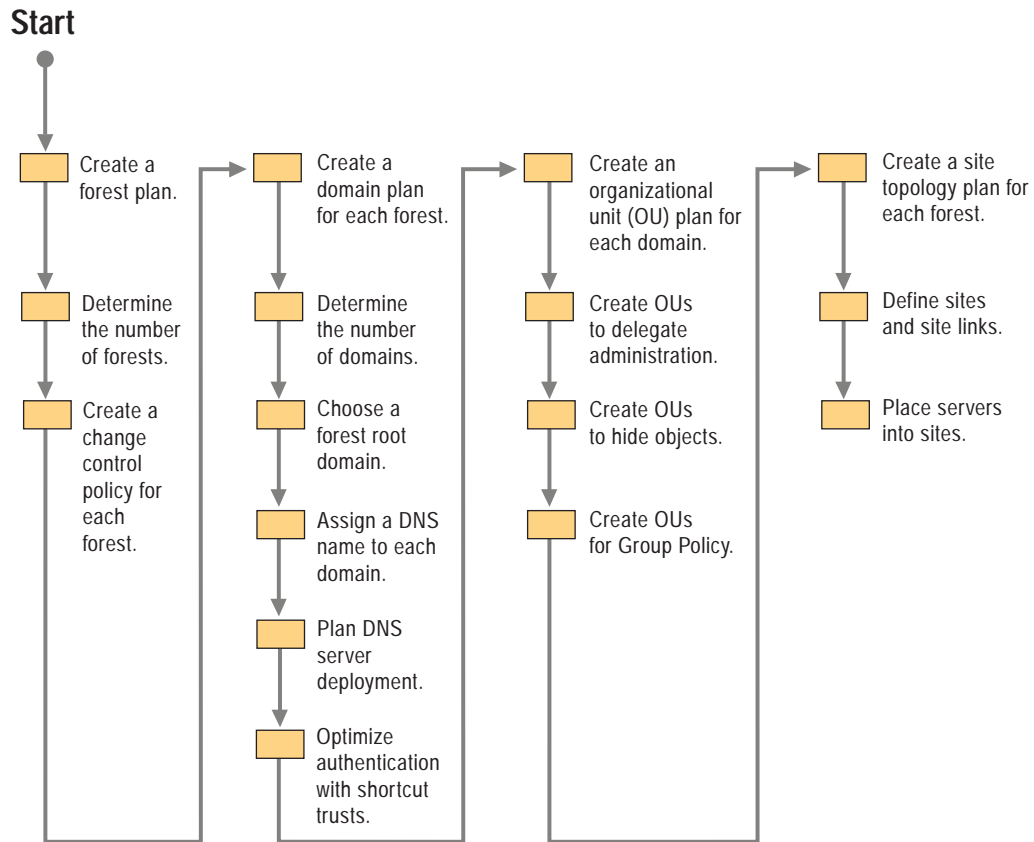


Figure 9.1 Process for Designing an Active Directory Structure

General Design Principles

When working on your Active Directory plan, use the following design principles to guide your decision making:

Simplicity is the best investment.

Simple structures are easier to explain, easier to maintain, and easier to debug. Although some added complexity can add value, be sure to weigh the incremental added value against the potential maintenance costs in the future. For example, the maximum optimization of query and replication traffic might require a complex site topology. However, a complex site topology is harder to maintain than a simple site topology. Always evaluate the tradeoff between added capabilities and added complexity before deciding on a complex structure.

Everything that you create will require some maintenance over its lifetime. When you create a structure without well-defined reasons, it will end up costing you more in the long run than any value that it adds. Justify the existence of any structure you create.

Your business and your organization will always change.

The normal changes that occur within any organization, ranging from employee moves to enterprise-wide reorganizations or acquisitions, will affect your Active Directory structure. When designing your structure, consider how these potential changes will affect end-user and administrator interaction with the directory. For example, consider the impact that your last major business reorganization would have had on the structures you have designed. What changes would be necessary if you add a new location or branch office? Would the changes have required significant and expensive changes to the Active Directory structure? Make sure your design is general enough and flexible enough to accommodate constant and significant change.

Aim for the ideal design.

In your first design pass, design what you consider to be the ideal structure, even if it does not reflect your current domain or directory infrastructure. It is useful and practical to understand what would be ideal, even if it is not currently attainable. For more information about the costs involved in migrating your network to the ideal plan, see “Determining Domain Migration Strategies” in this book. Weigh those costs against the long-term savings of the ideal plan, and refine the design appropriately.

Explore design alternatives.

Make more than one pass at each design. The value of a design becomes more evident when you compare it to other design ideas. Combine the best of all designs into the plan that you will implement.

Composing Your Active Directory Structure Plans

There are four basic components that make up an Active Directory structure: forests, domains, organizational units, and sites. The objective of an Active Directory Structure Plan is to create a planning document for each component of the structure, capturing important decisions and justifications along the way. These planning documents then serve as a starting point for your next planning task, migration. The four planning documents that make up the Active Directory Structure Plan are the following:

- Forest Plan
- Domain Plan for each forest
- Organizational Unit (OU) Plan for each domain
- Site Topology Plan for each forest

Creating a Forest Plan

A forest is a collection of Active Directory domains. Forests serve two main purposes: to simplify user interaction with the directory, and to simplify the management of multiple domains. Forests have the following key characteristics:

Single Schema

The Active Directory schema defines the *object classes* and the attributes of object classes that can be created in the directory. Object classes define the types of objects that can be created in the directory. The schema exists as a naming context that is replicated to every domain controller in the forest. The schema administrators security group has full control over the schema.

Single Configuration Container

The Active Directory *Configuration container* is a naming context that is replicated to every domain controller in the forest. Directory-aware applications store information in the Configuration container that applies forest wide. For example, Active Directory stores information about the physical network in the Configuration container and uses it to guide the creation of replication connections between domain controllers. The enterprise administrators security group has full control over the Configuration container.

Sharing a single, consistent configuration across the domains of a forest eliminates the need to configure domains separately.

Complete Trust

Active Directory automatically creates transitive, two-way trust relationships between the domains in a forest. Users and groups from any domain can be recognized by any member computer in the forest, and included in groups or access control lists (ACLs).

Complete trust makes managing multiple domains simpler in Windows 2000. In previous versions of Windows NT, a popular model for deploying domains was the Multiple Master Domain model. In that model, a domain containing primarily user accounts was called a master user domain, and a domain that contained primarily computer accounts and resources was called a resource domain. A common deployment consisted of a small number of master user domains, each of which was trusted by a large number of resource domains. Adding a new domain to the deployment required several trusts to be created. With Windows 2000 Active Directory, when you add a domain to a forest it is automatically configured with two-way transitive trust. This eliminates the need to create additional trusts with domains in the same forest.

Single Global Catalog

The *global catalog* contains a copy of every object from every domain in the forest but only a select set of the attributes from each object. The global catalog enables fast, efficient searches that span the entire forest.

The global catalog makes directory structures within a forest transparent to end users. Using the global catalog as a search scope makes finding objects in the directory simple. Logging on is made simpler through the global catalog and user principal names, described as follows:

Users Search the Global Catalog In the directory search user interface, the global catalog is abstracted as the **Entire**

Directory when selecting a search scope. Users can search the forest without having any prior knowledge of the forest structure. Having a single, consistent search interface reduces the need to educate users on directory structure, and allows administrators to change the structure within a forest without affecting the way users interact with the directory.

Users Log on Using User Principal Names A *user principal name (UPN)* is an e-mail-like name that

uniquely represents a user. A UPN consists of two parts, a user identification portion and a domain portion. The two parts are separated by an “@” symbol, to form *<user>@<DNS-domain-name>*, for example, *liz@noam.reskit.com*. Every user is automatically assigned a default UPN, where the *<user>* portion of the name is the same as the user’s logon name, and the *<DNS-domain-name>* portion of the name is the DNS name of the Active Directory domain where the user account is located.

When logging on using a UPN, users no longer have to choose a domain from a list on the logon dialog box.

You can set UPNs to arbitrary values. For example, even if Liz's account is in the noam.reskit.com domain, her UPN could be set to liz@reskit.com. When the user logs on, the user account to be validated is discovered by searching the global catalog for a user account with a matching UPN value. By making UPN values independent from domain names, administrators can move user accounts between domains, leaving UPN values unchanged and making interdomain moves more transparent to users.

Forest Planning Process

The primary steps for creating a forest plan for your organization are as follows:

- Determine the number of forests for your network.
- Create a forest change control policy.
- Understand the impact of changes to the forest plan after deployment.

When creating the forest plan, you will probably need to consult:

- Your current domain administrators that are responsible for user accounts, groups, and computers.
- Your network security team.

Determining the Number of Forests for Your Network

When you begin to plan your forest model, start with a single forest. A single forest is sufficient in many situations; however, if you decide to create additional forests, ensure that you have valid, technical justification.

Creating a Single Forest Environment

A single forest environment is simple to create and maintain. All users see a single directory through the global catalog, and do not need to be aware of any directory structure. When adding a new domain to the forest, no additional trust configuration is required. Configuration changes only need to be applied once to affect all domains.

Creating a Multiple-Forest Environment

If administration of your network is distributed among many autonomous divisions, it might be necessary to create more than one forest.

Because forests have shared elements, such as schema, it is necessary for all the participants in a forest to agree on the content and administration of those shared elements. Organizations such as partnerships and conglomerates might not have a central body that can drive this process. In short-lived organizations like joint ventures, it might not be realistic to expect administrators from each organization to collaborate on forest administration.

It might be necessary to create more than one forest if individual organizations:

Do not trust each other's administrators. A representation of every object in the forest resides in the global catalog. It is possible for an administrator who has been delegated the ability to create objects to intentionally or unintentionally create a "denial of service" condition. You can create this condition by rapidly creating or deleting objects, thus causing a large amount of replication to the global catalog. Excessive replication can waste network bandwidth and slow down global catalog servers as they spend time to process replication.

Cannot agree on a forest change policy. Schema changes, configuration changes, and the addition of new domains to a forest have forest-wide impact. Each of the organizations in a forest must agree on a process for implementing these changes, and on the membership of the schema administrators and enterprise administrators groups. If organizations cannot agree on a common policy, they cannot share the same forest. Creating a forest change policy is discussed later in this chapter.

Want to limit the scope of a trust relationship. Every domain in a forest trusts every other domain in the forest. Every user in the forest can be included in a group membership or appear on an access control list on any computer in the forest. If you want to prevent certain users from ever being granted permissions to certain resources, then those users must reside in a different forest than the resources. If necessary, you can use explicit trust relationships to allow those users to be granted access to resources in specific domains.

Incremental Costs for an Additional Forest

Each forest you create incurs a fixed management overhead as follows:

- Each additional forest must contain at least one domain. This might require you to have more domains than you had originally planned. There are fixed costs associated with creating and maintaining a domain. These costs are detailed later in this chapter.
- You must manage the forest-wide components of each forest separately (for example, the Schema and Configuration container elements and their associated administration group memberships), even if they are essentially the same.

In order for a user in one forest to use a resource in a different forest, you need to perform additional configuration as follows:

- For users in one forest to access resources in another forest, you must create and maintain an explicit trust relationship between the two domains. An explicit trust relationship between domains in different forests is one-way and not transitive. Without an established trust relationship, users in one forest cannot be granted access to objects in another forest.
- By default, users in one forest are only aware of objects in the global catalog of their own forest. To discover objects in a different forest, users must explicitly query domains that are outside their forest. Alternatively, administrators can import data from other domains into the forest where the user resides. This can add cost because:
 - Users must be trained to understand the directory structure so that they know where to direct queries when global catalog queries fail.
 - When you import data from a domain in a separate forest, you must put a process into place to keep the imported data up-to-date when it changes in the source domain.

Figure 9.2 is an example of an interforest configuration where a user in one forest needs to access a published resource in a different forest. An explicit, one-way trust relationship is created so the user can be granted access to the resource. The representation of the resource in the directory is imported into the user's forest, where it appears in the global catalog.

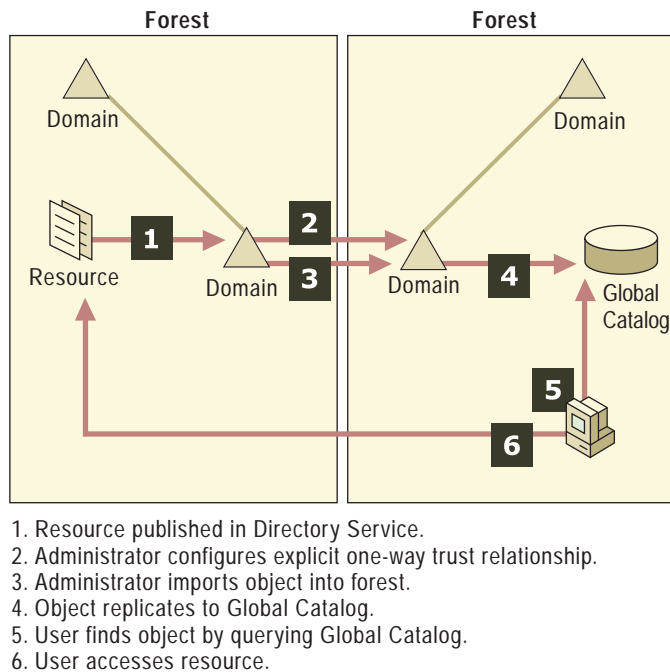


Figure 9.2 Additional Configuration for Interforest Resource Access

Some features that are available within a forest are not available between forests, such as the following:

- You can only use default UPNs if a user account is in a different forest than the computer being used for logging on. Default UPNs are required because a domain controller in the computer's forest will not find a user account with a matching UPN in the global catalog. The user account appears in the global catalog of a different forest. The domain controller handling the logon must then use the *<DNS-domain-name>* portion of the UPN to try to find a domain controller to verify the user's identity.
- Logging on using a smart card relies on a user principal name. Default UPNs must be used for a cross-forest logon process that uses smart cards to work.
- You can move security principals between domains in the same forest, but they must be cloned between domains in separate forests. Cloning is not as transparent to an end user as moving a user between domains. For more information about cloning, see "Determining Domain Migration Strategies" in this book.

When deciding on the number of forests that you will need, keep in mind that what is important to users is not necessarily the same as what is important to administrators. However, users stand to lose the most from a multiple forest scenario. For example, some organizations outsource their network administration to several different contractors. Generally, the contractor is paid based on network performance, and the number one responsibility is to maintain a stable network. One contractor might not want another contractor being able to influence computers under their control, and having separate forests can solve that challenge. However, separate forests can be a disadvantage for the users who no longer have a single, consistent view of the directory. In these situations, try not to create separate forests to solve the boundary of administration problem.

In cases where it is not important for all users to have a consistent view of the directory, it might be appropriate to have multiple forests. For example, consider a company such as an Internet service provider (ISP) that hosts Active Directory on behalf of several companies. The users in the different client companies have no reason to share a consistent view of the directory. Additionally, there is no reason to have a transitive trust relationship between the companies. In this case, maintaining separate forests is useful and appropriate.

Creating a Forest Change Control Policy

Each forest you create should have an associated Forest Change Control Policy as part of your Forest Plan document. You will use this policy to guide changes that have forest-wide impact. You do not need to determine the individual processes before continuing, but understanding their ownership is important. The policy should include information about each of the shared elements in a forest.

Schema Change Policy

The schema administrators group has full control over the schema for a forest. The schema change policy should include:

- The name of the team in your organization that controls the schema administrators group.
- The starting membership of the schema administrators group.
- Guidelines and a process for requesting and evaluating schema changes.

For more information about Active Directory schema, see the *Microsoft® Windows® 2000 Server Resource Kit Distributed Systems Guide*.

Configuration Change Policy

The enterprise administrators group has full control over the Configuration container that is replicated throughout the forest. The configuration change policy should include:

- The name of the team in your organization that controls the enterprise administrators group.
- The starting membership of the enterprise administrators group.
- Guidelines and a process for creating new domains in the forest.
- Guidelines and a process for modifying the forest site topology. (Site topology is discussed in “Creating a Site Topology Plan” later in this chapter.)

Changing the Forest Plan After Deployment

When a domain is created, it can be joined to an existing forest. You can create a domain by promoting a Windows 2000–based server to the Active Directory domain controller role, or by upgrading a Microsoft® Windows NT® version 3.51 or Microsoft® Windows NT® version 4.0 primary domain controller to Windows 2000.



Critical Decision Point Two forests cannot be merged in a one-step operation, nor can you move a domain between forests as a one-step operation. It is important that you design your forest plan so that it requires a minimum amount of restructuring as your organization evolves.

Individual objects can be moved between forests. The type of object being moved determines the particular tool that you use to move it. Most bulk importing and exporting can be achieved with the LDAP Data Interchange Format (LDIFDE.EXE) command-line tool; security principals can be cloned using the ClonePrincipal tool. For more information about these tools, see Tools Help on the *Windows 2000 Resource Kit* companion CD.

Creating a Domain Plan

The following are some of the key characteristics of a Windows 2000 domain that you will need to consider when you begin creating your domain structure plan:

Partition of the Forest

An Active Directory forest is a distributed database, where the partitions of the database are defined by domains. A *distributed database* is a database that is made up of many partial databases spread across many computers, instead of a single database on a single computer. Splitting a database into smaller parts and placing those parts where the data is most relevant allows a large database to be distributed efficiently over a large network.

Service by Domain Controller Servers

As in Windows NT 4.0, servers running Windows 2000 that host a domain database are called domain controllers. A *domain controller* can host exactly one domain. You can make changes to objects in the domain on any domain controller of that domain. All of the domain controllers in a particular forest also host a copy of the forest Configuration and Schema containers.

Unit of Authentication

Each domain database contains security principal objects, such as users, groups, and computers. Security principal objects are special in that they can be granted or denied access to the resources on a network. Security principal objects must be authenticated by a domain controller for the domain in which the security principal objects are located. Authentication is done to prove the identity of the objects before they access a resource.

Boundary of Administration and Group Policy

Each domain has a domain administrators group. Domain administrators have full control over every object in the domain. These administrative rights are valid within the domain only and do not propagate to other domains.

Group Policy that is associated with one domain does not automatically propagate to other domains in the forest. For a Group Policy from one domain to be associated with another domain, it must be explicitly linked.

Security Policy for Unique Domain User Accounts

A small set of security policies that apply to domain user accounts can only be set on a per-domain basis:

- Password policy. Determines the rules that must be met, such as password length, when a user sets a password.
- Account lockout policy. Defines rules for intruder detection and account deactivation.
- Kerberos ticket policy. Determines the lifetime of a Kerberos ticket. A Kerberos ticket is obtained during the logon process and is used for network authentication. A particular ticket is only valid for the lifetime specified in the policy. When tickets expire, the system automatically tries to obtain a new ticket.

For more information about security policy for domain user accounts, see “Authentication” in the *Microsoft® Windows 2000 Server Resource Kit Distributed Systems Guide*.

DNS Domain Names

A domain is identified by a DNS name. You use DNS to locate the domain controller servers for a given domain. DNS names are hierarchical, and the DNS name of an Active Directory domain indicates its position in the forest hierarchy. For example, reskit.com might be the name of a domain. A domain named eu.reskit.com can be a child domain of reskit.com in the forest hierarchy.

Domain Planning Process

Your domain plan will determine the availability of the directory on the network, the query traffic characteristics of clients, and the replication traffic characteristics of domain controllers.

Each forest that you create will contain one or more domains. The steps to creating a domain plan for a forest are:

- Determine the number of domains in each forest.
- Choose a forest root domain.
- Assign a DNS name to each domain to create a domain hierarchy.
- Plan DNS server deployment.
- Optimize authentication with shortcut trust relationships.
- Understand the impact of changes to the domain plan after deployment.

When creating the Domain Plan for each forest, you will most likely need to consult the following groups:

- Current domain administrators that are responsible for user accounts, groups, and computers
- Teams that manage and monitor your physical network
- Teams that manage the DNS service for your network
- Security teams

Determining the Number of Domains in Each Forest

To determine the number of domains that you will have in each forest, start by considering a single domain only, even if you currently have more than one Windows NT 4.0 domain. Next, provide a detailed justification for each additional domain. Every domain that you create will introduce some incremental cost in terms of additional management overhead. For that reason, be certain that the domains you add to a forest serve a beneficial purpose.

How Creating Domains Has Changed

Some of the factors that lead to the creation of multiple domain environments in previous versions of Windows NT Server no longer apply to Active Directory and Windows 2000. These factors are as follows:

Security Accounts Manager (SAM) Size Limitations

In previous versions of Microsoft® Windows NT® Server, the SAM database had a practical limitation of about 40,000 objects per domain. Active Directory can scale easily to millions of objects per domain. It should never be necessary to create additional domains in order to handle more objects.

Primary Domain Controller (PDC) Availability Requirements

In previous versions of Windows NT Server, only a single domain controller, the PDC, could accept updates to the domain database. In an organization with a large network, this limitation made it difficult to ensure high availability of the PDC, because a network outage could prevent administrators on one part of the network from being able to update the domain. To satisfy the availability requirement, you created additional domains so that PDC servers could be distributed throughout the network. This is no longer necessary in Windows 2000, because all Active Directory domain controllers can accept updates.

Limited Delegation of Administration Within a Domain

In previous versions of Windows NT Server, you delegated administration using built-in local groups such as the Account Operators group, or by creating multiple domains and having distinct sets of domain administrators. For example, to delegate the management of a set of users, you created a new user domain. To delegate management of resource servers like file or print servers, you created resource domains. In Windows 2000, it is possible to delegate administration within a domain using *organizational units (OUs)*. An OU is a container that you use to organize objects within a domain into logical administrative subgroups. OUs are easier to create, delete, move, and modify than domains, and they are better suited to the delegation role.

For more information about using OUs to delegate administration, see “Creating an Organizational Unit Plan” later in this chapter.

When to Create More Than One Domain

Three possible reasons for creating additional domains are:

- Preserving existing Windows NT domains.
- Administrative partitioning.
- Physical partitioning.

Preserving Existing Windows NT Domains

If you already have Windows NT domains in place today, you might prefer to keep them as they are instead of consolidating them into a smaller number of Active Directory domains. If you decide to keep or consolidate a domain, be sure to weigh those costs against the long-term benefits of having fewer domains. The costs associated with domain consolidation are discussed in the chapter “Determining Domain Migration Strategies” in this book. If this is your first time through domain design, it is recommended that you aim for the fewest domains possible, and reevaluate this plan after reading that chapter.

Administrative Partitioning

More domains might be necessary depending on the administrative and policy requirements of your organization, described as follows.

Unique domain user security policy requirements You might want to have a set of users on your network abide by a domain user security policy that is different from the security policy applied to the rest of the user community. For example, you might want your administrators to have a stronger password policy, such as a shorter password change interval, than the regular users on your network. To do this, you must place those users in a separate domain.

Division requires autonomous domain administration supervision

The members of the domain administrators group in a particular domain have complete control over all objects in that domain. If you have a subdivision in your organization that will not allow outside administrators control over their objects, place those objects in a separate domain. For example, for legal reasons, it might not be prudent for a subdivision of an organization that works on highly sensitive projects to accept domain supervision from a higher level IT group. Remember that all domains in the forest must share the Configuration and Schema containers.

Physical Partitioning

Physical partitioning involves taking the domains you have in a forest and dividing them up into a greater number of smaller domains. Having a greater number of smaller domains allows you to optimize replication traffic by only replicating objects to places where they are most relevant. For example, in a forest containing a single domain, every object in the forest is replicated to every domain controller in the forest. This might lead to objects being replicated to places where they are rarely used, which is an inefficient use of bandwidth. For example, a user that always logs on at a headquarters location does not need their user account replicated to a branch office location. Replication traffic can be avoided by creating a separate domain for the headquarters location and not replicating that domain to the branch office.

Note If you have already deployed Windows NT 4.0 domains, you might be satisfied with your existing physical partitioning. Looking at partitioning again from a clean sheet can help you identify areas for possible domain consolidation. If you have already decided to upgrade your Windows NT 4.0 domains in place and not perform any consolidation, you can skip the partitioning discussion.

To determine if, and how, to partition a forest, you should:

- Draw your network topology.
- Place domain controllers in the network according to availability requirements.
- Partition the forest based on the replication traffic between domain controllers.

Draw Your Network Topology

Begin by drawing a basic network topology diagram. Later in the planning process you will add more detail to this diagram when you plan your site topology. To create the topology diagram:

- Draw collections of sites.

A site is a network of fast, reliable connectivity. A local area network (LAN) or set of LANs connected by a high-speed backbone can be considered a site. Draw each site on your network diagram and indicate the approximate number of users at the site.

- Connect sites with site links.

A site link is a slow or unreliable link that connects two sites. A wide area network (WAN) that connects two fast networks is an example of a site link. It is recommended you treat any link that is slower than LAN speed as a slow link. On the topology diagram, show how each site connects to other sites with site links.

For each site link, record the following:

- Link speed and current usage levels
 - Whether the link is pay-by-usage
 - Whether the link is historically unreliable
 - Whether the link is only intermittently available
- Mark sites with SMTP connectivity only.
- If you have a site that has no physical connection to the rest of your network but can be reached via Simple Mail Transfer Protocol (SMTP) mail, mark that site as having mail-based connectivity only.

Figure 9.3 shows the network topology for the fictitious Reskit company.

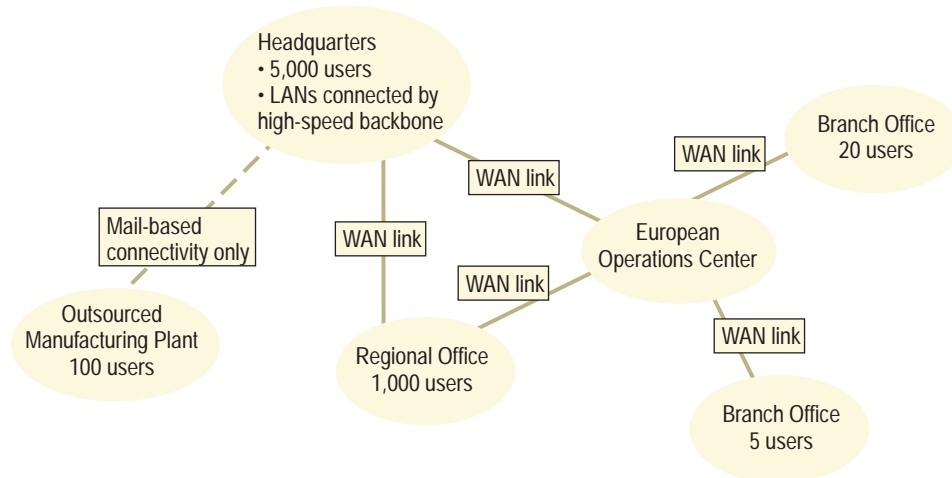


Figure 9.3 Reskit Company Network Topology

Place Domain Controllers

The availability of Active Directory is determined by the availability of domain controllers. Domain controllers must be available so that users can be authenticated. In this step, you will determine where you need to place domain controllers to maintain availability in case of possible network outages.

To place domain controllers, use the following process:

- Select a “home” site and place a domain controller in the site by marking it on the topology diagram.

You can select the home site arbitrarily. For example, use your headquarters location, the site with the largest number of users, or the site with the best overall connectivity to the rest of your network. All the users in the home site will authenticate with this domain controller. Ignore for now the question of what domain is being serviced by that domain controller, and how many replicas of that domain will be necessary in the site.

- For each site that is directly connected to the home site, determine if you need to place a domain controller into that site.

Or, instead of placing a domain controller in that site, determine if users in that site can authenticate over the site link back to the domain controller in the home site. If it is acceptable to you that authentication fails when the site link fails, then you do not need to place a domain controller into the site.

For small branch offices that have client computers but no servers, a domain controller is not necessary. If the link back to the central site fails, users in the office will still be able to log on to their computers using cached credentials. Further authentication is unnecessary because there are no other server-based resources to access in the office—all of the resources are back at the central site.

You should put a domain controller into the site if:

- There are a large number of users in the site, and the site link is slow or near capacity. In this case, you do not want Active Directory client traffic to take up capacity on the link. For more information about network capacity planning and the traffic generated by an Active Directory client, see the Microsoft Windows 2000 Server link on the Web Resources page at <http://windows.microsoft.com/windows2000/reskit/webresources>.
 - The link is historically unreliable. You do not want authentication to fail if the link is down.
 - The link is intermittently available. You do not want authentication to fail at certain times of the day or to rely on a demand-dial link.
 - The site is only accessible using SMTP mail. Users must have a local domain controller for authentication if the site is only accessible via SMTP mail.
- Repeat the previous process to determine where you need to place domain controllers.

Apply the same process to the next adjacent site, until you have visited every site and determined whether or not a local domain controller is necessary.

Note Domain controllers contain security-sensitive information, such as copies of users' secret keys used for domain authentication. Having fewer copies of this information reduces the opportunities for unauthorized access. Domain controllers must be physically secure from unauthorized access. For example, it is recommended that domain controllers be located in a locked room with limited access. Physical access can allow an intruder to obtain copies of encrypted password data to use for an off-line password attack. Stronger security options are available using the Syskey tool. For more information about Syskey, see "Encrypting File System" in the *Distributed Systems Guide*.

When a user logs on, the domain controller servicing the authentication request must be able to communicate with a global catalog server. When you decide to place a domain controller in a site, you need to also consider that domain controller as a global catalog server. As you proceed, keep in mind that global catalog servers generate more replication traffic than regular domain controllers. They contain both a complete copy of one domain and a read-only partial copy of every other domain in the forest.

Figure 9.4 shows the domain controller placement for the Reskit company.

- The first domain controller is placed in the headquarters home location.
- A domain controller is placed in the European operations center because the transoceanic WAN link is already near capacity.
- A domain controller is placed in the regional office because there are too many users for the WAN to carry the authentication traffic.
- Domain controllers are not placed in the branch offices because there are no local servers in the branch offices.
- A domain controller is placed in the manufacturing plant because it can be reached by SMTP mail only.

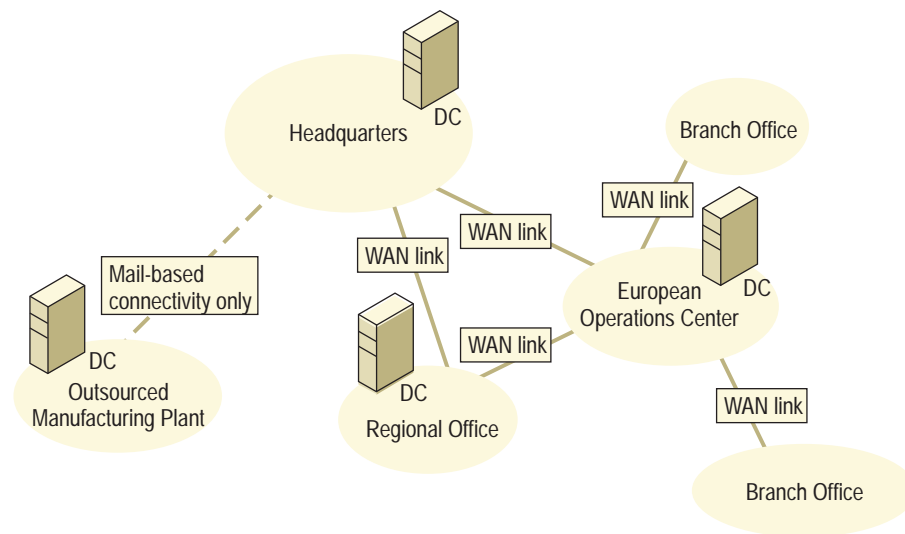


Figure 9.4 Reskit Company Domain Controller Placement

Partition the Forest

Now you will assign a domain to each domain controller, determine if your network can handle the replication traffic, and partition your forest into smaller domains if necessary. While doing this, remember that the objective of partitioning is to put physical copies of directory objects near the users that need those objects. For example, a user's user account object needs to be located on a domain controller that is in the same site as the user.

To partition the forest, perform the following steps for each domain currently in the domain plan:

- For each site that contains a domain controller, decide if the domain is relevant to the users in the site. If appropriate, place a domain controller for the domain into the site.

- Trace the path that replication will follow between domain controllers for the domain. Assume that each domain controller will choose the next nearest domain controller for the same domain as a replication partner, where “nearest” is determined by the most cost-effective path through the network.
- The volume of replication traffic between any two domain controllers for a domain is a factor of how often the objects in the domain change, how many of them change, and how often objects are added and deleted. By splitting a domain into two or more smaller domains, you can decrease the amount of replication traffic that will travel over a particular link. Examine each edge in the replication path and decide if you will permit the replication traffic or split the domain.

Consider these factors when deciding whether or not to replicate a domain between sites, or to split it into two or more smaller domains:

- Consider splitting the domain if a site link in the replication path cannot accommodate the anticipated replication traffic.

The actual capacity of a site link is a function of link speed, daily usage characteristics, reliability, and availability. Consider the following information about a link when deciding whether to create a domain:

- A link operating near capacity might not be able to accommodate replication. Keep in mind that Active Directory replication can be scheduled, so if the link has idle periods during the day, there might be enough actual bandwidth for replication to keep up.
- A link might only be available during certain times of day, lowering its actual bandwidth. Active Directory replication can be scheduled to occur only when the link is available, but the actual bandwidth must be high enough to accommodate replication.

For more information about network capacity planning and Active Directory replication traffic, see the Microsoft Windows 2000 Server link on the Web Resources page at <http://windows.microsoft.com/windows2000/reskit/webresources>.

- Consider splitting the domain if you do not want replication traffic to compete with other, more important traffic on a link.

Interrupting or delaying business-critical traffic might be much more costly than adding an additional domain.

- Consider splitting the domain if replication traffic will cross a pay-by-usage link.

When a link is pay-by-usage, minimizing usage will minimize your costs.

- Create domains for sites that are connected by SMTP mail only.
Active Directory mail-based replication can only occur between domains. Mail-based replication cannot be used between domain controllers of the same domain.

If you do decide to split a large domain into several smaller domains, a good strategy for creating the smaller domains is to base them on geography or geo-political boundaries. For example, create domains that map to countries or continents. Geographical mapping for domains is recommended because network topologies tend to map to geographical locations, and geography tends to change less than any other basis for divisions.

You might want to create a greater number of smaller domains simply to optimize replication traffic on your network. Remember, optimization is a tradeoff against other factors, such as:

- Complexity
As discussed earlier, each additional domain introduces a fixed on-going management overhead.
- Query traffic versus replication traffic
The fewer the number of objects in a domain, the more likely a user in that domain will want to access objects that are in some other domain. If there is no local domain controller for the other domain, the query will cause traffic to leave the site.

Note A model with a single large domain works best with a large roaming user population because every user account will be available in every site that has a domain controller. In this case, a roaming user will never lose the ability to log on if a network outage occurs between the user's current location and home location.

Figure 9.5 shows the physical partitioning for the Reskit company. The domain assignments are as follows:

- The Noam domain for users in North America is assigned to a domain controller in the home site.
- The Avionics domain, which was created for administrative reasons, is assigned to a domain controller in the home site since there are Avionics users at headquarters.
- A new domain, Eu, is assigned to a domain controller in the European operations center because the transoceanic WAN link is near capacity. The link cannot handle the replication traffic for both the North American and European domains combined.

- The Avionics domain is also represented in the European Operations center, since there are Avionics users in Europe.
- A new domain, Seville, is assigned to a domain controller in the regional office in Seville because the WAN link back to the European operations center is carrying business-critical traffic.
- A new domain, Mfg, is assigned to a domain controller in the manufacturing plant because it can only be accessed by SMTP mail.

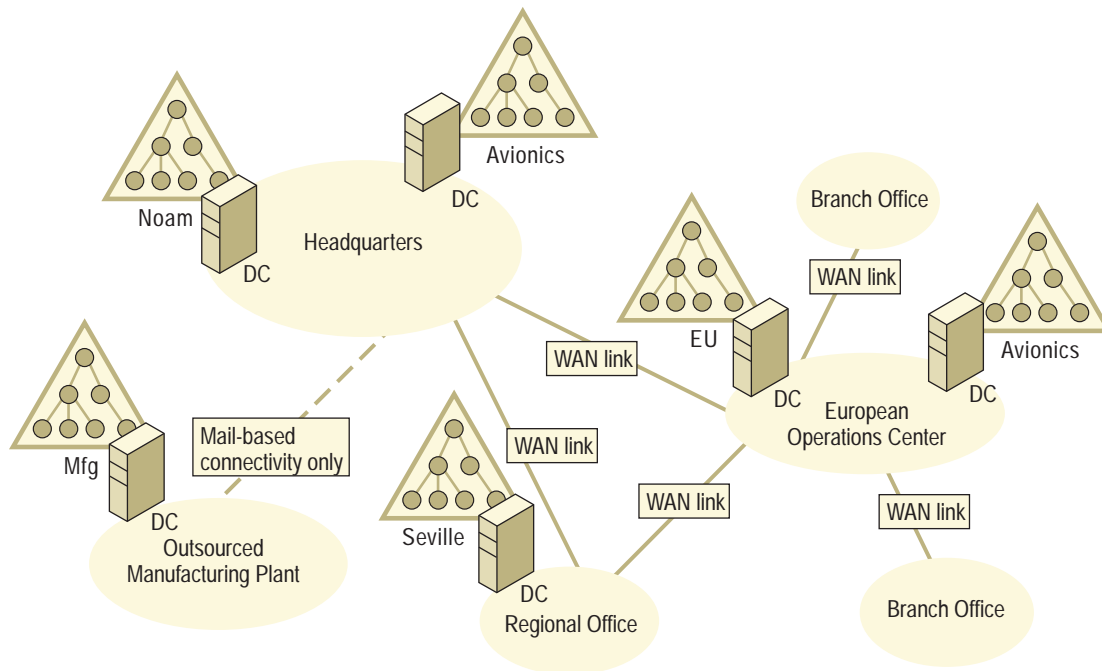


Figure 9.5 Reskit Company Domain Assignment

Incremental Costs for an Additional Domain

Each domain in the forest will introduce some amount of management overhead. When debating whether or not to add a domain to your domain plan, weigh the following costs against the benefits you determined earlier in the chapter.

More Domain Administrators Because domain administrators have full control over a domain, the membership of the domain administrators group for a domain must be closely monitored. Each added domain in a forest incurs this management overhead.

More Domain Controller Hardware In Windows 2000, a domain controller can host only a single domain. Each new domain that you create will require at least one computer, and in most cases will require two computers to meet reliability and availability requirements. Because all Windows 2000 domain controllers can accept and originate changes, you must physically guard them more carefully than you did Windows NT 4.0 backup domain controllers (BDCs), which were read-only computers. Note that the administration delegation within Active Directory domains reduces the requirement for resource domains. Some remote locations that currently must host two domain controllers (a master user domain and a local resource domain) will now only require one domain controller if you choose to consolidate to fewer Active Directory domains.

More Trust Links For a domain controller in one domain to authenticate a user from another domain, it must be able to contact a domain controller within the second domain. This communication represents an added possible point of failure if, for example, the network between the two domain controllers is malfunctioning at the time. The more users and resources located in a single domain, the less an individual domain controller must rely on being able to communicate with other domain controllers to maintain service.

Greater Chance of Having to Move a Security Principal Between Domains

The more domains you have, the greater the chance you have to move security principals, such as users and groups, between two domains. For example, a business reorganization or a job change for a user can create the need to move a user between domains. To end users and administrators, moving a security principal between OUs inside a domain is a trivial and transparent operation. However, moving a security principal between domains is more involved and can impact the end user.

For more information about moving security principals between domains, see “Determining Domain Migration Strategies” in this book.

Group Policy and Access Control Do Not Flow Between Domains

Group Policy and access control applied within a domain do not flow automatically into other domains. If you have policies or delegated administration through access control that is uniform across many domains, they must be applied separately to each domain.

Choosing a Forest Root Domain

After you have determined how many domains you will place in your forest, you need to decide which domain will be the forest root domain. The *forest root domain* is the first domain that you create in a forest. The two forest-wide groups, enterprise administrators and schema administrators, will reside in this domain.

Note If all of the domain controllers for the forest root domain are lost in a catastrophic event, and one or more domain controllers cannot be restored from backup, the enterprise administrators and schema administrators groups will be permanently lost. There is no way to reinstall the forest root domain of a forest.

If your forest contains only one domain, that domain will be the forest root. If your forest contains two or more domains, consider the following two approaches for selecting the forest root domain.

Using an Existing Domain

From the list of domains you have, select a domain that is critical to the operation of your organization and make it the forest root. Because you cannot afford to lose this domain, it will already require the kind of fault tolerance and recoverability that is required for a forest root.

Using a Dedicated Domain

Creating an additional, dedicated domain to serve solely as the forest root carries all the costs of an extra domain, but it has certain benefits that might apply to your organization, such as:

- The domain administrator in the forest root domain will be able to manipulate the membership of the enterprise administrators and schema administrators groups. You might have administrators who require domain administrator privilege for some part of their duties, but you do not want them to manipulate the forest-wide administrators groups. By creating a separate domain, you avoid having to place these administrators into the domain administrators group of the forest root domain.
- Because the domain is small, it can be easily replicated anywhere on your network to provide protection against geographically-centered catastrophes.
- Because the only role the domain has is to serve as the forest root, it never risks becoming obsolete. In the case where you select a domain from your planned list of domains to be the forest root, there is always a chance that particular domain will become obsolete, perhaps due to a change in your organization. However, you will never be able to fully retire such a domain, because it must play the role of forest root.

Assigning DNS Names to Create a Domain Hierarchy

Active Directory domains are named with DNS names. Because DNS is the predominant name system on the Internet, DNS names are globally recognized and have well-known registration authorities. Active Directory clients requesting to log on to the network query DNS to locate domain controllers.

In Windows NT 4.0, the domain locator was based on the network basic input/output system (NetBIOS) Name System (NBNS), and domains were identified with NetBIOS names. The server-based component of NBNS is called the Windows Internet Name Service (WINS) server. NetBIOS names are simple, single-part names, and the NetBIOS namespace cannot be partitioned. In contrast, DNS names are hierarchical, and the DNS namespace can be partitioned along the lines of the hierarchy. As a result, DNS is more scalable than NBNS and can accommodate a larger database spread over a larger network. Internet mail, which leverages DNS in a manner similar to Active Directory, is a good example of how DNS as a locator mechanism can scale to extraordinarily large networks such as the Internet.

Note For interoperability with computers that run earlier versions of Windows, Active Directory domains have NetBIOS names and Active Directory domain controllers register in NBNS and query NBNS when necessary. This allows clients that run earlier versions of Windows to locate Active Directory domain controllers, and allows Active Directory domain controllers and Windows NT 3.51 and Windows NT 4.0 domain controllers to locate each other.

Arranging Domains into Trees

A *tree* is a set of one or more Windows 2000 domains with contiguous names. Figure 9.6 presents a single tree with a contiguous namespace. Because reskit.com does not have a parent domain, it is considered the *tree root domain*. The *child domains* of reskit.com are eu.reskit.com and noam.reskit.com. A grandchild domain of reskit.com is mfg.noam.reskit.com. These domain names are contiguous because each name is only one label different than the name of the domain above it in the domain hierarchy.

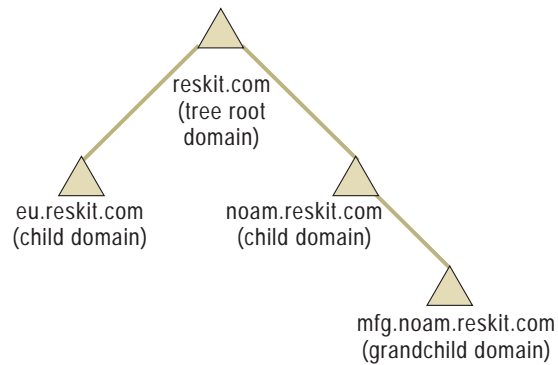


Figure 9.6 Single Tree with Four Domains

A forest can have more than one tree. In a multiple tree forest, the names of the tree root domains are not contiguous, as shown in Figure 9.7. You might have multiple trees in your forest if a division of your organization has its own registered DNS name and runs its own DNS servers.

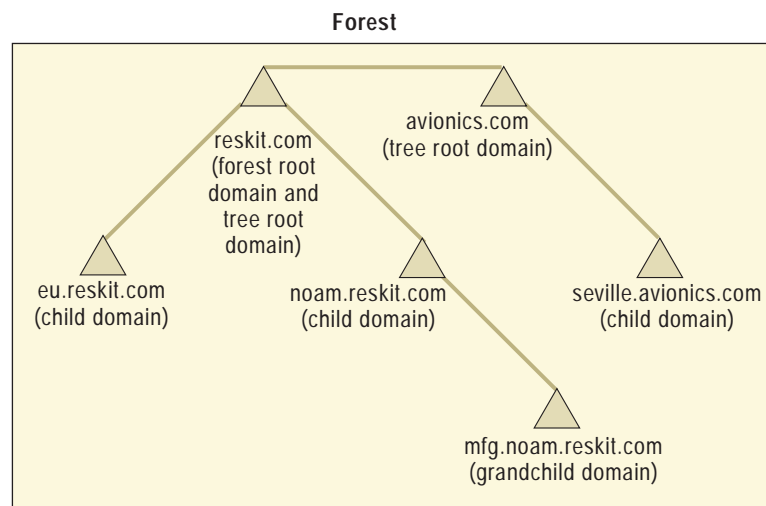


Figure 9.7 Forest with Multiple Trees

The domain hierarchy in a forest determines the transitive trust links that connect each domain. Each domain has a direct trust link with its parent and each of its children. If there are multiple trees in a forest, then the forest root domain is at the top of the trust tree and all other tree roots are children, from a trust perspective. Figure 9.8 depicts a transitive trust relationship between two trees.

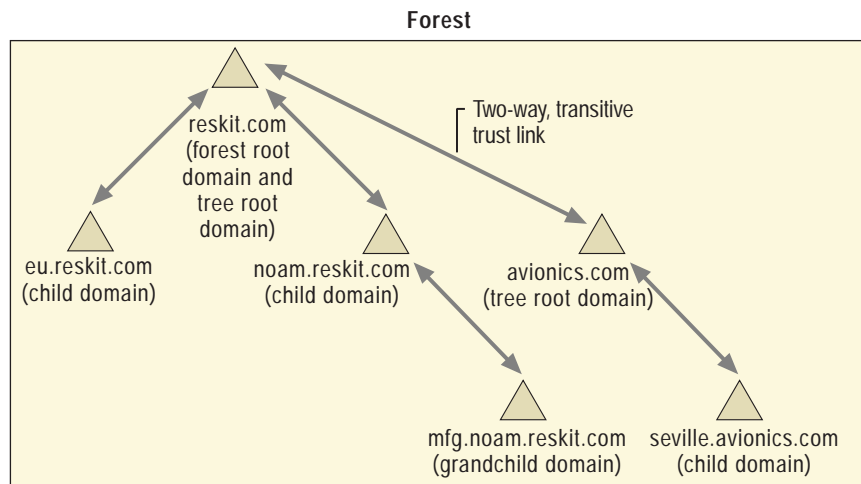


Figure 9.8 Transitive Trust Relationship Between Trees

The parent-child relationship is a naming and trust relationship only. Administrators in a parent domain are not automatically administrators of a child domain. Policies set in a parent domain do not automatically apply to child domains.

Domain Naming Recommendations

To create the domain hierarchy in a forest, assign a DNS name to the first domain, and then for every subsequent domain decide if it is a child of an existing domain or if it is a new tree root. Based on that evaluation, assign names accordingly. Some recommendations for naming domains are as follows:

Use names relative to a registered Internet DNS name.

Names registered on the Internet are globally unique. If you have one or more registered Internet names, use those names as suffixes in your Active Directory domain names.

Use Internet standard characters.

Internet standard characters for DNS host names are defined in Request for Comments (RFC) 1123 as A–Z, a–z, 0–9, and the hyphen (-). Using only Internet standard characters ensures that your Active Directory will comply with standards-based software. To support the upgrade of earlier Windows-based domains to Windows 2000 domains that have nonstandard names, Microsoft clients and the Windows 2000 DNS service will support almost any Unicode character in a name.

Never use the same name twice.

Never give the same name to two different domains, even if those domains are on unconnected networks with different DNS namespaces. For example, if the Reskit company decides to name a domain on the intranet `reskit.com`, it should not also create a domain on the Internet called `reskit.com`. If a `reskit.com` client connects to both the intranet and Internet simultaneously, it would select the domain that answered first during the locator search. To the client, this selection would appear random, and there is no guarantee that the client will select the “correct” domain. An example of such a configuration is a client that has established a virtual private network connection to the intranet over the Internet.

Use names that are distinct.

Some proxy client software, such as the proxy client built into Microsoft® Internet Explorer or the Winsock Proxy client, use the name of a host to determine if that host is on the Internet. Most software of this type provides, at minimum, a way of excluding names with certain suffixes as being local names, instead of assuming that they are on the Internet.

If the Reskit company wants to call an Active Directory domain on their intranet `reskit.com`, they would have to enter `reskit.com` in the exclusion list of their proxy client software. This would prevent clients on the Reskit intranet from seeing a host on the Internet called `www.reskit.com`, unless they provide an identical site on the intranet.

To avoid having this problem, the Reskit company could use a registered name that does not have a presence on the Internet, such as `reskit-int01.com`, or establish a company policy that states names ending in a specific suffix of `reskit.com`, for example `corp.reskit.com`, would never appear on the Internet. In both cases, it is easy to configure proxy client exclusion lists so that they can determine which names are on the intranet and which are on the Internet.

There are many different techniques for accessing the Internet from a private intranet. Before using any name, ensure that it can be properly resolved by clients on your intranet within your specific Internet access strategy.

Use the fewest number of trees possible.

There are some advantages to minimizing the number of trees in your forest. The following advantages could apply in your environment:

- After you have been given control over a particular DNS name, you own all names that are subordinate to that name. The smaller the number of trees, the smaller the number of DNS names that you must take ownership of in your organization.

- There are fewer names to enter in the proxy client-excluded suffixes list.
- LDAP client computers that are not Microsoft clients might not use the global catalog when searching the directory. Instead, to perform directory-wide searches, these clients will use deep searches. A deep search covers all of the objects in a particular subtree. The fewer the number of trees in a forest, the fewer deep searches you will have to perform to cover the entire forest.

Make the first part of the DNS name the same as the NetBIOS name.

It is possible to assign a domain a DNS name and NetBIOS name that are entirely unrelated. For example, the DNS name of a domain could be sales.reskit.com, but the NetBIOS name could be “Marketing.” Keep in mind that pre-Windows 2000 computers and non-Active Directory-aware software will display and accept NetBIOS names; whereas, Windows 2000 computers and Active Directory-aware software will display and accept DNS names. This can potentially lead to confusion on the part of your end users and administrators.

You should only use unmatched NetBIOS and DNS names if:

- You want to migrate to a new naming convention on your network.
- You are upgrading a NetBIOS name that contains nonstandard characters but you want the DNS name to have all standard characters.

Review names internationally.

Names that have a benign or useful meaning in one language can sometimes be derogatory or offensive in another language. DNS is a global namespace; be sure to review your names globally within your organization.

Note If you have multiple localized versions of Windows running on your network, all computers, including Windows 2000 Professional and all versions of Windows 2000 Server, must use only Internet-standard characters in both their DNS and NetBIOS names. If you use characters other than those described above, only computers with the same locale setting will be able to communicate with each other.

Use names that are short enough to remember.

Length should not be a significant deciding factor when choosing names. Users typically interact with the global catalog and are not concerned with domain names. Typically, only administrators are exposed to domain names. Administrative tools almost always present a list of domains to choose from, and the number of cases where an administrator has to type a full name will be the exception, not the rule. In general, if you can remember all the components of a name then it is not too long.

Domain Names and Computer Names

Windows 2000 computers that are joined to a domain will, by default, assign themselves a DNS name that is made up of the host name of the computer and the DNS name of the domain the computer has joined. For example, in Figure 9.9 if the computer account for Server 1 is located in eu.reskit.com, the computer will name itself by default server1.eu.reskit.com. However, it is possible to use any arbitrary DNS suffix instead of the Active Directory domain name. For this reason, it is not necessary to name your Active Directory domains to fit a DNS structure that is already deployed in your organization. Your Active Directory domains can use any name, and your computers can retain their existing names.

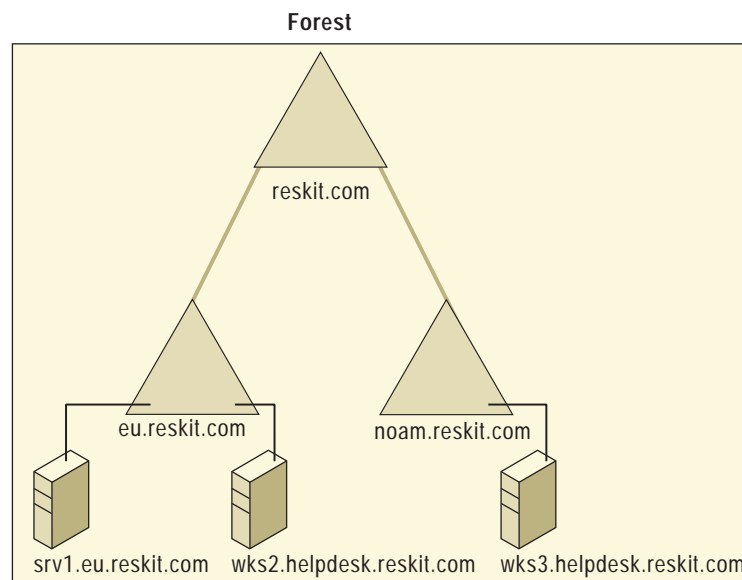


Figure 9.9 Member Computers with Default and Nondefault Names

For more information about computer naming, see “Windows 2000 DNS” in the *TCP/IP Core Networking Guide*.

Planning DNS Server Deployment

To plan DNS server deployment for support of your Active Directory domains, you must identify the DNS servers that will be authoritative for your domain names, and ensure that they meet the requirements of the domain controller locator system.

Authority and Delegation in DNS

The Domain Name System is a hierarchical, distributed database. The database itself consists of resource records, which primarily consist of a DNS name, a record type, and data values that are associated with that record type. For example, the most common records in the DNS database are Address (A) records, where the name of an Address record is the name of a computer, and the data in the record is the TCP/IP address of that computer.

Like Active Directory, the DNS database is divided into partitions that enable the database to scale efficiently even on very large networks. A partition of the DNS database is called a zone. A zone contains the records for a contiguous set of DNS names. A DNS server that loads a zone is said to be authoritative for the names in that zone.

A zone begins at a specified name and ends at a delegation point. A delegation point indicates where one zone ends and another zone begins. For example, there is a registration authority on the Internet that is responsible for the zone called “com.” Inside this zone are thousands of delegation points to other zones, for example, reskit.com. The data in a delegation point indicates which servers are authoritative for the delegated zone. Figure 9.10 shows the relationship among DNS servers, zones, and delegations.

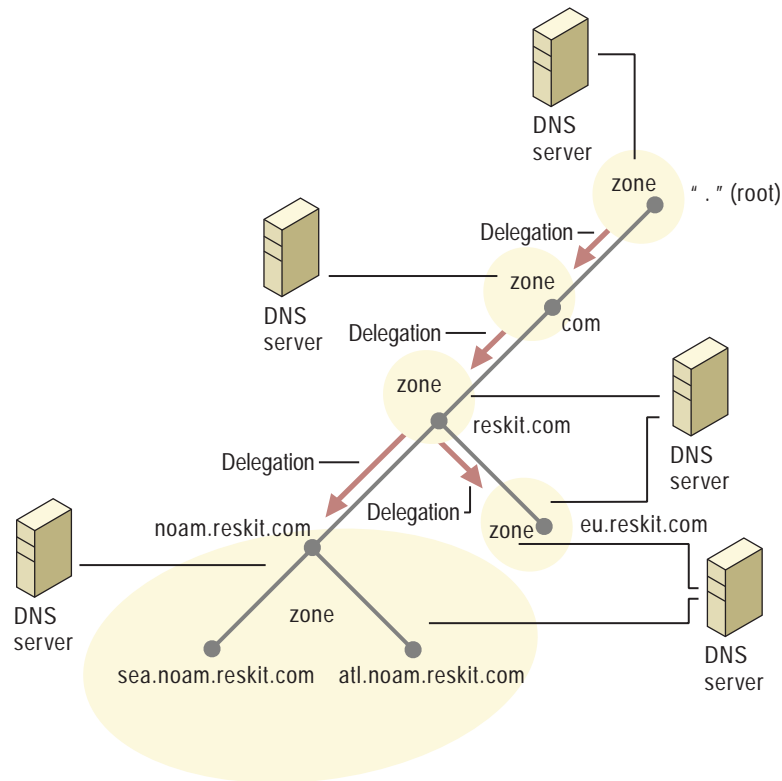


Figure 9.10 Servers, Zones, and Delegations in DNS

Domain Controller Locator System

Domain controllers register a set of records in DNS. These records are collectively called the locator records. When a client requires a particular service from a domain, it sends a query for a specific name and type of record to the nearest DNS server. The answer is a list of domain controllers that can satisfy the request.

The names of the locator records for each domain end in *<DNS-domain-name>* and *<DNS-forest-name>*. The DNS servers that are authoritative for each *<DNS-domain-name>* are authoritative for the locator records.

Note Windows 2000 does not require reverse lookup zones to be configured. Reverse lookup zones might be necessary for other applications, or for administrative convenience.

DNS Server Requirements

If you do not already have DNS servers running on your network, it is recommended that you deploy the DNS service that is provided with Windows 2000 Server. If you have existing DNS servers, then the servers that are authoritative for the locator records must meet the following requirements to support Active Directory:

- Must support the Service Location resource record.

The DNS servers that are authoritative for the locator records must support the *Service Location (SRV)* resource record type. For more information about the SRV record, see “Introduction to DNS” in the *TCP/IP Core Networking Guide*.

- Should support the DNS dynamic update protocol.

The DNS servers that are authoritative for the locator records and are the primary master servers for those zones should support the DNS dynamic update protocol as defined in RFC 2136.

The DNS service provided with Windows 2000 Server meets both these requirements and also offers two important additional features:

- Active Directory integration

Using this feature, the Windows 2000 DNS service stores zone data in the directory. This makes DNS replication create multiple masters, and it allows any DNS server to accept updates for a directory service–integrated zone. Using Active Directory integration also reduces the need to maintain a separate DNS zone transfer replication topology.

- Secure dynamic update

Secure dynamic update is integrated with Windows security. It allows an administrator to precisely control which computers can update which names, and it prevents unauthorized computers from obtaining existing names from DNS.

The remaining DNS servers on your network that are not authoritative for the locator records do not need to meet these requirements. Servers that are not authoritative are generally able to answer SRV record queries even if they do not explicitly support that record type.

Locate Authoritative Servers

For each DNS name you choose, consult your DNS management team and find out if the DNS server supports the listed requirements. If you find one that does not, there are three basic courses of action that you can take:

Upgrade the server to a version that supports the requirements.

If the authoritative servers are running the Windows NT 4.0 DNS service, simply upgrade those servers to Windows 2000. For other DNS server implementations, consult the vendor's documentation to find out which version supports the features necessary to support Active Directory.

If the authoritative DNS servers are not under your control, and you cannot persuade the owners of those servers to upgrade, you can use one of the other options.

Migrate the zone to Windows 2000 DNS.

You can migrate the zone from the authoritative servers to Windows 2000 DNS instead of upgrading those servers to a version that supports Active Directory requirements. Migrating a zone to Windows 2000 DNS is a straightforward process. Introduce one or more Windows 2000 DNS servers as secondary servers for the zone. After you are comfortable with the performance and manageability of the servers, convert the zone on one of the servers to be the primary copy, and rearrange the DNS zone transfer topology as necessary.

Delegate the name to a DNS server that meets the requirements.

If upgrading and migrating authoritative servers are not suitable options, you can change the authoritative servers by delegating the domain name to Windows 2000 DNS servers. How this is done depends on the relationship of the domain name to the existing zone structure.

- If the domain name is not the same as the name of the root of a zone, the name can be delegated directly to Windows 2000 DNS servers. For example, if the name of the domain is noam.reskit.com and the zone that contains this name is reskit.com, delegate noam.reskit.com to a Windows 2000 DNS server.
- If the domain name is the same as the name of the root of a zone, you cannot delegate the name directly to a Windows 2000 DNS server. Instead, delegate each of the subdomains used by the locator records to a Windows 2000 DNS server. Those subdomains are: *_msdcs.<DNS-domain-name>*, *_sites.<DNS-domain-name>*, *_tcp.<DNS-domain-name>*, and *_udp.<DNS-domain-name>*. If you do this, you will have to register the *<DNS-domain-name>* address (A) records by hand. For more information about this topic, see "Windows 2000 DNS" in the *TCP/IP Core Networking Guide*.

Optimizing Authentication with Shortcut Trust Relationships

When a user requests access to a network resource, a domain controller from the user's domain must communicate with a domain controller from the resource's domain. If the two domains are not in a parent-child relationship, the user's domain controller must also communicate with a domain controller from each domain in the trust tree between the user's domain and the resource's domain. Depending on the network location of the domain controllers for each domain, each extra authentication hop between the two domains can increase the chance of a possible failure, or increase the likelihood of authentication traffic having to cross a slow link. To reduce the amount of communication necessary for such interactions, you can connect any two domains with a *shortcut trust* relationship.

For example, if you have multiple trees in a forest, you might want to connect the group of tree roots in a complete mesh of trust. Remember that in the default arrangement, all tree roots are considered children of the forest root from a trust perspective. That means authentication traffic between any two domains in different trees must pass through the forest root. Creating a complete mesh of trust allows any two tree root domains to communicate with each other directly.

Figure 9.11 shows a complete mesh of trust created between four tree root domains.

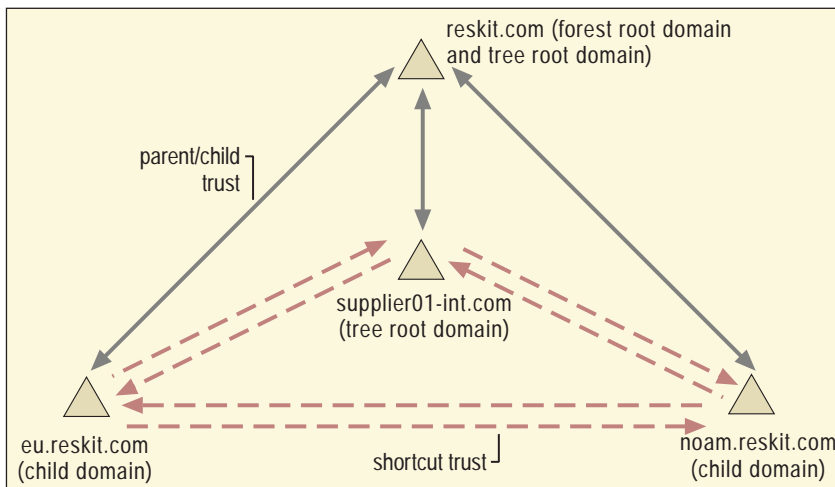


Figure 9.11 Complete Mesh of Trust Between Four Domains

Changing the Domain Plan After Deployment

Domain hierarchies are not easy to restructure after they have been created. For this reason, it is best not to create domains that are based on a temporary or short-lived organizational structure. For example, creating a domain that maps to a particular business unit in your organization might create work for you if that business unit is split up, disbanded, or merged with another unit during a corporate reorganization. However, there are cases where organization-based partitioning is appropriate. Geopolitical boundaries provide a relatively stable template for partitioning, but only if the organization does not frequently move across those boundaries. Consider a domain plan for an army, where the army has different divisions spread across a number of bases. It might be common for divisions to move between bases. If the forest were partitioned according to geographic location, administrators would have to move large numbers of user accounts between domains when a division moved between bases. If the forest were partitioned according to divisions, administrators would only have to move domain controllers between bases. In this case, organization-based partitioning is more appropriate than geographic partitioning.

Adding New Domains and Removing Existing Domains

It is easy to add new domains to a forest; however you cannot move existing Windows 2000 Active Directory domains between forests.



Critical Decision Point After a tree root domain has been established, you cannot add a domain with a higher level name to the forest. You cannot create a parent of an existing domain; you can only create a child. For example, if the first domain in a tree is called eu.reskit.com, you cannot later add a parent domain called reskit.com.

Demoting all of the domain controllers for a domain to the member server or standalone role will remove a domain from a forest and delete all of the information that was stored in the domain. A domain can only be removed from the forest if it has no child domains.

Merging and Splitting Domains

Windows 2000 does not provide the ability to split a domain into two domains or to merge two domains into one domain in a single operation.



Critical Decision Point It is important that you design your domain plan to require a minimum amount of partitioning changes as your organization evolves.

It is possible to split a domain by adding an empty domain to a forest and then move objects into that domain from other domains. In the same way, it is possible to merge one domain with another domain by moving all of the objects from the source domain into the target domain. As mentioned previously, moving security principals between domains can impact end users. For more information about moving objects between domains, see “Determining Domain Migration Strategies” in this book.

Renaming Domains

Windows 2000 does not provide the ability to rename a domain in-place. Because the name of a domain is also representative of its position in a tree hierarchy, it is also true that a domain cannot be moved within a forest.



Critical Decision Point When selecting names for your domains, choose names that you believe will continue to be meaningful as your organization evolves.

The alternative to in-place renaming is to create a new domain in the forest with the desired new name, and then move all the objects from the old domain into the new domain.

Creating an Organizational Unit Plan

An Organizational Unit (OU) is the container you use to create structure within a domain. The following characteristics of OUs are important to consider when creating structure in a domain.

OUs can be nested. An OU can contain child OUs, enabling you to create a hierarchical tree structure inside a domain.

OUs can be used to delegate administration and control access to directory objects.

When you use a combination of OU nesting and access control lists, you can delegate the administration of objects in the directory in a very granular manner. For example, you could grant a group of Help desk technicians the right to reset passwords for a specific set of users, but not the right to create users or modify any other attribute of a user object.

OUs are not security principals. You cannot make OUs members of security groups, nor can you grant users permission to a resource because they reside in a particular OU. Because OUs are used for delegation of administration, the parent OU of a user object indicates who manages the user object, but it does not indicate the resources a user can access.

Group Policy can be associated with an OU. Group Policy enables you to define desktop configurations for users and computers. You can associate Group Policy with sites, domains, and OUs. Defining Group Policy on an OU basis allows you to use different policies within the same domain. For more information about Group Policy, see “Applying Change and Configuration Management” and “Defining Client Administration and Configuration Standards” in this book.

Users will not navigate the OU structure. It is not necessary to design an OU structure that will appeal to end users. Although it is possible for users to navigate the OU structure of a domain, it is not the most efficient way for a user to discover resources. The most efficient way to find resources in the directory is by querying the global catalog.

OU Structure and Business Structure

At first, the phrase “organizational unit structure” might start you thinking about creating a structure that mirrors your business organization and its various divisions, departments, and projects. It is possible to create such a structure, but it might prove difficult and expensive to manage. OUs are for delegating administration, so the structure you create is most likely a reflection of your administrative model. The administrative model of your organization might not map exactly to your business organization.

For example, consider the business-oriented structure shown in Figure 9.12. OUs have been created for the Home Electronics (Electronics OU), Medical Systems (Medical OU), and Automotive (Automotive OU) divisions, where the users on the Automotive teams are in the Automotive OU, and so on.

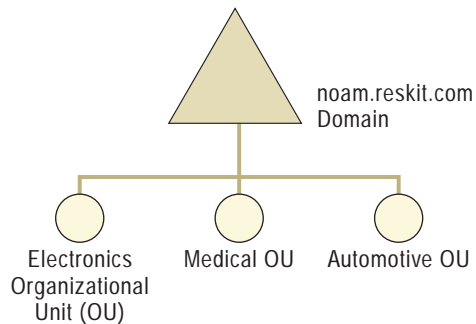


Figure 9.12 OU Structure Aligned with Business Structure

Assume that the company in this example uses a centralized administration model. A single group of administrators manages all of the users across the company, regardless of business division. During the day-to-day operation of the company, many things can happen. If a person transfers between the Home Electronics and Automotive divisions, an administrator has to move that person’s user account from the Electronics OU to the Automotive OU. If the number of transfers is high, this could amount to a significant amount of work for the administration group. But what is actually being accomplished?

For the same company, now consider an OU structure that consists of a single OU that contains all user accounts. If a user transfers between divisions, no additional work to move the object is created for an administrator. Whenever you create structure, make sure that it serves a meaningful purpose. Structure without justification will always create unnecessary work.

You might want to mirror your business structure in your OU structure to make it easy to generate lists of users based on business unit. Using OUs is just one way of doing this. Your business structure might more closely reflect the way resource access is granted to your users. For example, users on a particular project might be granted access to a specific set of file servers, or users in a particular division might be granted access to a particular Web site. Because resource access is granted using security groups, you might find that your business organizational structure is best represented in security group structures instead of OUs.

OU Planning Process

The steps to creating an OU structure plan for a domain are:

- Create OUs to delegate administration.
- Create OUs to hide objects.
- Create OUs for Group Policy.
- Understand the impact of changing OU structures after deployment.

It is important to follow the steps in the order presented. You will find that an OU structure designed purely for delegation of administration is shaped differently than an OU structure designed purely for Group Policy. Because there are multiple ways of applying Group Policy, but only one way to delegate administration, you should create OUs for delegation of administration first.

Your OU structure can become complex very quickly. Note the specific reason for creating an OU each time you add one to the plan. This will help you make sure that every OU has a purpose, and it will help the readers of your plan to understand the reasoning behind the structure.

When creating the OU plan for each domain, consult the following groups in your administrative organizations:

- Current domain administrators who are responsible for user accounts, security groups, and computers accounts.
- Current resource domain owners and administrators.

Creating OUs to Delegate Administration

In versions of Windows NT previous to Windows 2000, delegation of administration within a domain was limited to the use of built-in local groups, such as the Account Administrators group. These groups had predefined capabilities, and in some cases the capabilities did not fit the needs of a particular situation. As a result, there were situations where administrators in an organization needed high levels of administrative access, such as Domain Administrators rights.

In Windows 2000, delegation of administration is more powerful and flexible. This flexibility is achieved through a combination of organizational units, per-attribute access control, and access control inheritance. Administration can be delegated arbitrarily by granting a set of users the ability to create specific classes of objects, or modify specific attributes on specific classes of objects.

For example, your human resources department can be granted the ability to create user objects in a particular OU, but nowhere else. Helpdesk technicians can be granted the ability to reset the passwords of users in that OU, but not the ability to create users. Other directory administrators can be granted the ability to modify the address book attributes of a user object, but not be allowed to create users or reset passwords. Delegating administration in your organization has several benefits. Delegating specific rights enables you to minimize the number of users who must have high levels of access. Accidents or mistakes made by an administrator with restricted capability will only have an impact within their area of responsibility. Previously, in your organization it might have been necessary for groups other than IT to submit change requests to high-level administrators, who would make these changes on their behalf. With delegation of administration, you can push responsibility down to the individual groups in your organization and eliminate the overhead of sending requests to high-level administrative groups.

Modifying Access Control Lists

To delegate administration, grant a group specific rights over an OU. To do this, you need to modify the access control list (ACL) of the OU. The access control entries (ACEs) in the ACL of an object determine who can access that object and what kind of access they have. When an object is created in the directory, a default ACL is applied to it. The default ACL is described in the schema definition of the object class. ACEs can be inherited by child objects of a container object. If any of the child objects are also containers, the ACEs are applied to the children of those containers as well. With inheritance, you can apply a delegated right to an entire subtree of OUs instead of a single OU. You can also block ACE inheritance on an object to prevent ACEs from a parent container from applying to that object or any child objects. Inheritable ACEs only apply within a domain and do not flow down to child domains. To delegate control over a set of objects in an OU subtree, you edit the ACL on the OU. The easiest way to do this is by using the Delegation of Control wizard in the Microsoft Management Control (MMC) snap-in for Active Directory Users and Groups. To view the ACL on an object or to perform advanced editing of an ACL, use the **Security** tab on the property page for the object.

Always reference groups in ACLs, not individual users. Managing the membership of a group is simpler than managing an ACL on an OU. When users change roles, it is much easier to discover and change their group memberships than to check the ACLs on every OU. Where possible, delegate to local groups instead of global or universal groups. Unlike global groups, local groups can have members from any trusted domain, making them better suited for granting resource permissions. Unlike universal groups, local group membership is not replicated to the global catalog, making local groups less resource intensive.

Deciding What OUs to Create

The OU structure that you create will depend entirely on how administration is delegated within your organization. Three ways to delegate administration are:

- By physical location. For example, administration for objects in Europe can be handled by an autonomous set of administrators.
- By business unit. For example, administration of objects belonging to the Avionics division can be handled by an autonomous set of administrators.
- By role or task. This division is according to the type of object being managed. For example, a set of administrators might be responsible only for computer account objects.

These three dimensions are frequently combined. For example, as shown in Figure 9.13, there can be an administrative group that is responsible for computer account objects in Atlanta for the Automotive business unit.

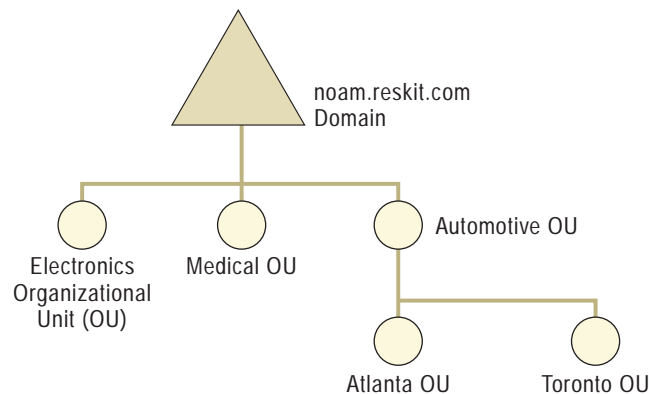


Figure 9.13 Two-Tiered Delegation

Whether or not the Atlanta OU is the child of the Automotive OU depends on whether the Automotive administrators delegate authority to the Atlanta administrators, or vice versa. It is also possible that the Atlanta administrators are completely autonomous from the Automotive administrators; therefore, the two OUs would be peers.

Note Some organizations have geographically distributed administrative groups in order to support 24-hour operation. The combined normal work hours of all administrative groups gives the organization 24-hour coverage. In this situation, the scope of each administrative group is not specific to location, because the administrators must be able to assist users all around the world. Even though this scenario has administrators distributed over many locations, it is not an example of location-based delegation.

Delegation Procedures

Starting with the default structure inside a domain, create an OU structure using the following primary steps:

- Create the top layers of OUs by delegating full control.
- Create the bottom layers of OUs to delegate per-object class control.

Delegating Full Control

To begin, only domain administrators have full control over all objects. Ideally, domain administrators should only be responsible for:

- Creating the initial OU structure.
- Repairing mistakes.

Domain administrators not only have full control by default, they also have the right to take ownership of any object in the domain. Using this right, domain administrators can gain full control over any object in the domain, regardless of the permissions that have been set on the object.

- Creating additional domain controllers.

Only members of the domain administrators group can create additional domain controllers for a domain.

Because domain administrators can have limited and specific duties, the membership of the group can be kept small and controlled.

If you have units in your organization that need to be allowed to determine their own OU structure and their own administrative model, use the following steps:

- Create an OU for each unit.
- Create a local group for each unit representing the highest level administrators in that unit.
- Assign the corresponding group full control over its OU.
- If the unit is allowed to set their its membership, place the unit's administrators group into the OU. If the unit is not allowed to set its own administrator membership, leave the group outside of the OU.

Example of Delegating Full Control

The Automotive unit of the Reskit company is the result of a merger of two companies, where the Automotive unit retained a fully autonomous IT group. In this situation, the Automotive unit gets its own OU from the root of the domain. Because they are also allowed to define the membership of their administrators group, the group is placed into the Automotive OU. If the Automotive unit itself had completely autonomous operations in Atlanta and Toronto, the Automotive administrators might again create OUs and delegate full control. As shown in Figure 9.14, the Automotive administrators have retained the ability to set the membership of the Atlanta and Toronto administrators groups.

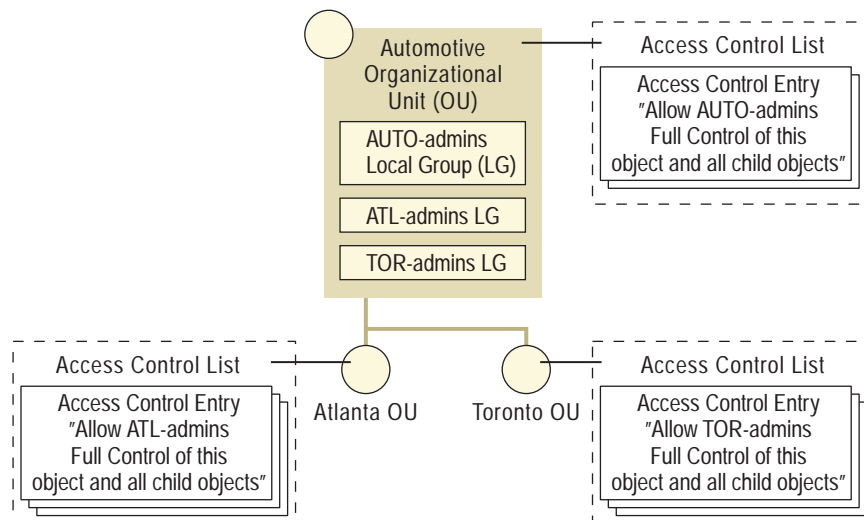


Figure 9.14 Delegating Full Control

If you do not have any units in your organization that need full control, domain administrators will determine the remainder of the OU structure.

Delegating Per-Object Class Control

Groups with full control can decide if additional OUs are necessary to delegate more restrictive control. A simple way to do this is to consider each object class that will be created in the directory, and determine if management of that object class is delegated further in the organization. Although the schema defines many different kinds of object classes, it is only necessary to consider the object classes that your administrators will create in the Active Directory. At minimum, you should consider:

- User account objects
- Computer account objects
- Group objects
- Organizational unit objects

As you examine each object class, separately consider:

- Which groups should be granted full control over objects of a particular class? Groups with full control can create and delete objects of the specified class and modify any attribute on objects of the specified class.
- Which groups should be allowed to create objects of a particular class? By default, users have full control over objects that they create.
- Which groups should only be allowed to modify specific attributes of existing objects of a particular class?

In each case that you decide to delegate control, you will:

- Create a local group that will be allowed to perform the specific function.
- Grant that group the specific right on the highest OU possible.

Note To move an object between two OUs, the administrator performing the move must have the ability to create the object in the destination container and to delete the object from the source container. For these reasons, you might want to create a separate group for administrators that can move objects, and grant them the necessary rights on a common parent OU.

The list of objects to be considered can grow as you deploy more Active Directory–aware applications. However, some applications will create objects in the directory that do not require hands-on management. For example, print servers running Windows 2000 automatically publish print queues in the directory. Because the print server takes care of the management of the print queue object, it is not necessary to delegate management to a special administrators group.

By modifying the ACL on the default Computers container, you can delegate the ability to create computer account objects to all users, with no administrative attention required. Computer accounts would be created when users join a computer to a domain in the default Computers container.

Example of Delegating Per-Object Class Control

The Atlanta location for the Automotive unit of the Reskit company is home to two Windows NT 4.0 resource domains, Powertrain and Chassis. Part of the Windows 2000 migration will involve consolidating those two domains into the noam.reskit.com domain.

The Powertrain and Chassis administrators use the domain today to:

- Create computer accounts for team members.
- Share file system space on Windows NT 4.0 backup domain controllers (BDCs), where access to the file system and shares are controlled by local group membership.

Using delegation of administration, it is simple to replace resource domains with OUs. In this case, groups are created to administer each kind of object, and they are granted full control in a project-specific OU. Project-specific OUs are necessary to prevent Powertrain administrators from being able to manipulate Chassis objects, and vice versa. Figure 9.15 illustrates this concept.

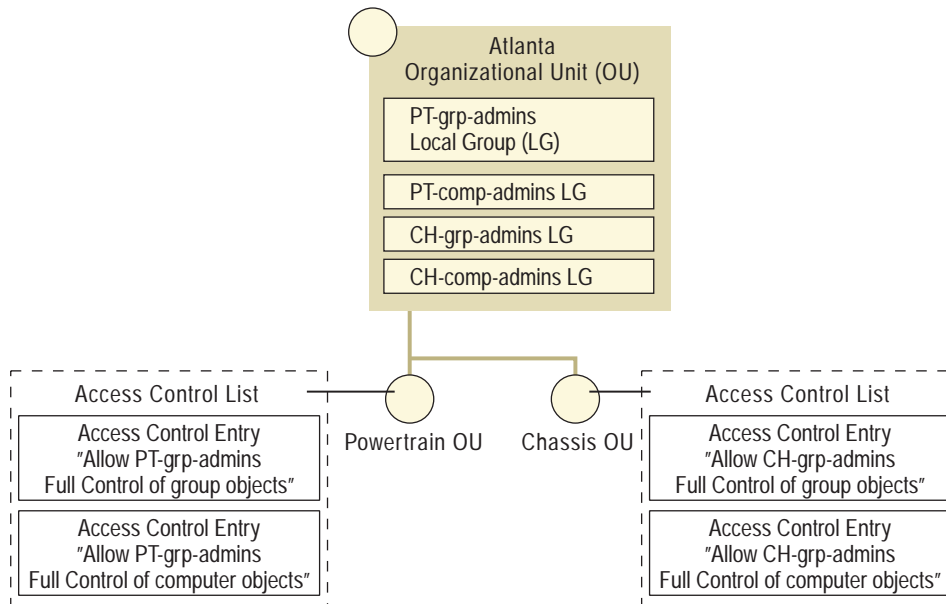


Figure 9.15 Replacing Resource Domains

Creating OUs to Hide Objects

Even if a user does not have the right to read the attributes of an object, that user can still see that the object exists by enumerating the contents of that object's parent container. The easiest and most efficient way to hide an object or set of objects is to create an OU for those objects and limit the set of users who have the List Contents right for that OU.

- **To create an OU to hide objects**
 1. Create the OU where you will hide objects.
 2. Click the **Security** tab on the property sheet on the OU.
 3. Remove all existing permissions from the OU.
 4. In the **Advanced** dialog box, clear the **Inherit permissions from parent** check box.
 5. Identify the groups that you want to have full control on the OU. Using the **Security** tab on the property sheet, grant those groups full control.
 6. Identify the groups that should have generic read access on the OU and its contents. Using the **Security** tab on the property sheet, grant those groups read access.
 7. Identify any other groups that might need specific access, such as the right to create or delete a particular class of objects, on the OU. Using the **Security** tab on the property sheet, grant those groups the specific access.
 8. Move the objects you want to hide into the OU.

Only users who can modify the ACL on an OU will be able to hide objects in this manner.

Creating OUs for Group Policy

In Windows NT 4.0, you can use the System Policy Editor to define user and computer configurations for all of the users and computers in a domain. With Windows 2000, you use Group Policy to define user and computer configurations, and associate those policies with sites, domains or OUs. Whether or not you will need to create additional OUs to support the application of Group Policy depends on the policies you create and the implementation options you select. For more information about Group Policy, see "Applying Change and Configuration Management" and "Defining Client Administration and Configuration Standards" in this book.

Changing the OU Plan After Deployment

Creating new OUs, moving OU subtrees within a domain, moving objects between OUs in the same domain, and deleting OUs are simple tasks.

Moving an object or subtree of objects will change the parent container of those objects. ACEs that were inherited from the old parent will no longer apply, and there might be new inherited ACEs from the new parent. To avoid unexpected changes in access, evaluate in advance what the changes will be and determine whether those changes will have any impact on the users that currently access and manage those objects.

Moving a user object, computer object, or a subtree containing user or computer objects can change the Group Policy that is applied to those objects. To avoid unexpected changes in client configurations, evaluate the changes in Group Policy and ensure that they are acceptable for end users.

Creating a Site Topology Plan

An Active Directory site topology is a logical representation of a physical network. Site topology is defined on a per-forest basis. Active Directory clients and servers use the site topology of a forest to route query and replication traffic efficiently. A site topology also helps you decide where to place domain controllers on your network. Keep the following key concepts in mind when designing your site topology:

A site is a set of networks with fast, reliable connectivity.

A site is defined as a set of IP subnets connected by fast, reliable connectivity. As a rule of thumb, networks with LAN speed or better are considered fast networks.

A site link is a low-bandwidth or unreliable network that connects two or more sites.

Site links are used to model the amount of available bandwidth between two sites. As a general rule, any two networks connected by a link that is slower than LAN speed is considered to be connected by a site link. A fast link that is near capacity has a low effective bandwidth, and can also be considered a site link. Site links have four parameters:

- Cost

The cost value of a site link helps the replication system determine when to use the link when compared to other links. Cost values will determine the paths that replication will take through your network.

- Replication schedule

A site link has an associated schedule that indicates at what times of day the link is available to carry replication traffic.

- Replication interval
The replication interval indicates how often the system polls domain controllers on the other side of the site link for replication changes.
- Transport
The transport that is used for replication.

Client computers first try to communicate with servers located in the same site as the client.

When a user turns on a client computer, the computer sends a message to a randomly selected domain controller of the domain in which the client is a member. The domain controller determines the site in which the client is located based on its IP address, and returns the name of the site to the client. The client caches this information and uses it the next time it is looking for a replicated server in the site.

Active Directory replication uses the site topology to generate replication connections.

The knowledge consistency checker (KCC) is a built-in process that creates and maintains replication connections between domain controllers. Site topology information is used to guide the creation of these connections. Intra-site replication is tuned to minimize replication latency, and inter-site replication is tuned to minimize bandwidth usage. Table 9.1 shows the differences between intra-site and inter-site replication.

Table 9.1 Intra-site vs. Inter-site Replication

Intra-site replication	Inter-site replication
Replication traffic is not compressed to save processor time.	Replication traffic is compressed to save bandwidth.
Replication partners notify each other when changes need to be replicated, to reduce replication latency.	Replication partners do not notify each other when changes need to be replicated, to save bandwidth.
Replication partners poll each other for changes on a periodic basis.	Replication partners poll each other for changes on a specified polling interval, during scheduled periods only.
Replication uses the remote procedure call (RPC) transport.	Replication uses the TCP/IP or SMTP transport.

(continued)

Table 9.1 Intra-site vs. Inter-site Replication (*continued*)

Intra-site replication	Inter-site replication
Replication connections can be created between any two domain controllers located in the same site.	Replication connections are only created between bridgehead servers.
The KCC creates connections with multiple domain controllers to reduce replication latency.	One domain controller from each domain in a site is designated by the KCC as a bridgehead server. The bridgehead server handles all inter-site replication for that domain.
	The KCC creates connections between bridgehead servers using the lowest cost route, according to site link cost. The KCC will only create connections over a higher cost route if all of the domain controllers in lower cost routes are unreachable.

Site topology information is stored in the Configuration container.

Sites, site links, and subnets are all stored in the configuration container, which is replicated to every domain controller in the forest. Every domain controller in the forest has complete knowledge of the site topology. A change to the site topology causes replication to every domain controller in the forest.

Note Site topology is separate and unrelated to domain hierarchy. A site can contain many domains, and a domain can appear in many sites.

Site Topology Planning Process

To create a site topology for a forest, use the following process:

- Define sites and site links using your physical network topology as a starting point.
- Place servers into sites.
- Understand how changes to your site topology after deployment will impact end users.

When creating the site topology plan, you will most likely need to consult:

- Teams that manage and monitor the TCP/IP implementation on your network.
- Domain administrators for each domain in the forest.

For more information about sites or any of the topics discussed in this section, see the *Distributed Systems Guide*.

Defining Sites and Site Links

To create the site topology for a forest, you will take the physical topology of your network and create a more general topology based on available bandwidth and network reliability.

If you performed the physical partitioning exercise when you created your domain plan, you can use the site topology and domain controller placement plan that you created as a starting point for your site topology. If you skipped the physical partitioning exercise earlier in this chapter, it is recommended that you see “Determining the Number of Domains in Each Forest” and create a basic site topology now.

When creating your site topology, it is useful to have a complete map of the physical topology of your network. That map should include the list of physical subnets on your network, the media type and speed of each network, and the interconnections between each network.

Creating Sites

To begin, create a list of sites on your network.

- Create a site for each LAN, or set of LANs, that are connected by a high speed backbone, and assign the site a name. Connectivity within the site must be reliable and always available.
- Create a site for each location that does not have direct connectivity to the rest of your network and is only reachable via SMTP mail.
- Determine which sites will not have local domain controllers, and merge those sites with other, nearby sites. Sites help route client-to-domain controller and domain controller-to-domain controller traffic efficiently. Without a domain controller in a site, there is no replication traffic into the site to be controlled.

For each site you add to the plan, record the set of IP subnets that comprise the site. You will need this information later when you create the sites in the directory.

Note Site names are used in the records that are registered in DNS by the domain locator, so they must be legal DNS names. It is recommended that you only use the standard characters A–Z, a–z, 0–9, and the hyphen (–) in site names.

Remember, clients will attempt to communicate with domain controllers in the same site as the client before trying to communicate with domain controllers in any other site. Any time bandwidth between a set of networks is plentiful enough that you do not care whether a client on one network communicates with a server on a different network, then consider those networks all to be in one site.

If a client is on a subnet that is not defined in the directory, it is not considered part of a site, and it selects randomly from all domain controllers for a particular domain. You might encounter situations where not all subnets are defined in the directory, such as when new subnets are being added to your network. To associate these clients with a site, create the two default subnets shown in Table 9.2 and then associate them with a site.

Table 9.2 Default Subnets

Subnet ID	Mask	Description
128.0.0.0	192.0.0.0	Captures all clients on class B networks not yet defined in the directory.
192.0.0.0	224.0.0.0	Captures all clients on class C networks not yet defined in the directory.

There is no default subnet for clients on a class A network.

Any time two networks are separated by links that are heavily used during parts of the day and are idle during other parts of the day, put those networks into separate sites.

You can use the ability to schedule replication between sites to prevent replication traffic from competing with other traffic during high usage hours.

If your entire network consists of fast, reliable connectivity, the entire network can be considered a single site.

Connecting Sites with Site Links

Next, connect sites with site links to reflect the physical connectivity of your network.

Assign each site link a name.

Site links are transitive, so if site A is connected to site B, and site B is connected to site C, then the KCC assumes that domain controllers in site A can communicate with domain controllers in site C. You only need to create a site link between site A and site C if there is in fact a distinct network connection between those two sites.

For each site link you create, record the following information:

- Replication schedule

Replication polling only occurs during the scheduled period or periods over a seven-day interval. The default schedule on a link allows replication polling to happen throughout the seven-day interval.

- Replication interval

Replication polling occurs at the specified interval when the schedule allows replication. The default polling interval is three hours.

- Replication transport
If the site is only reachable via SMTP, select the SMTP transport. Otherwise, select the TCP/IP transport.
- Link cost
Assign a cost value to each site link to reflect the available bandwidth or cost of bandwidth as compared to other site links.

A backbone network that connects many sites can be represented by a single site link that connects many sites, instead of creating a mesh of links between sites. This is a useful way to reduce the number of site links that need to be created and managed if many links have the same characteristics. Figure 9.16 illustrates how a frame relay network that connects four offices can be represented as a single link, instead of a mesh of six individual links.

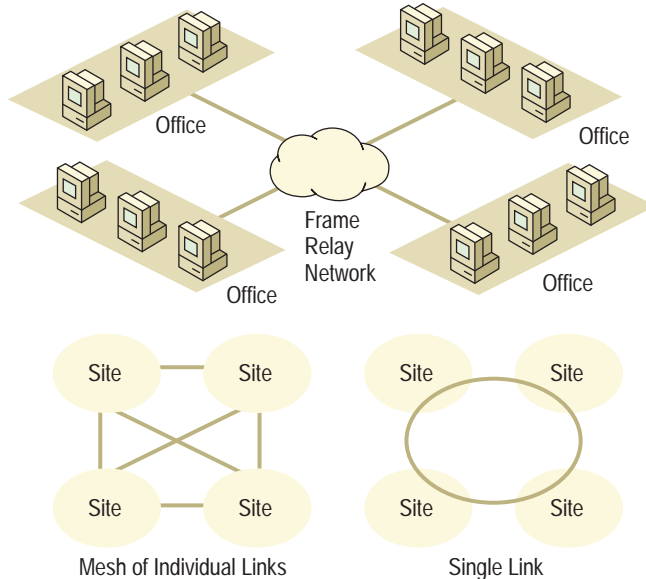


Figure 9.16 Single Link or Mesh of Links

Note The replication schedule determines when a domain controller polls replication partners for changes. If a replication cycle is underway when the scheduled window closes, replication continues until the current cycle is complete.

Figure 9.17 shows the site topology for the Reskit company. The site naming convention uses a combination of region code, the code of the nearest airport, and an identifying number. Site link names include the names of the connected sites.

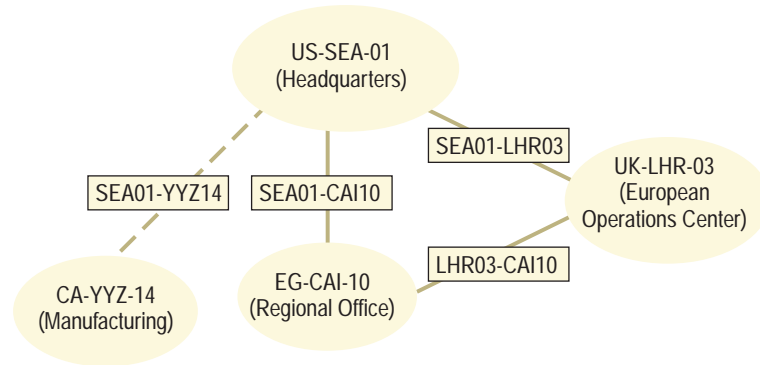


Figure 9.17 Reskit Company Site Topology

Table 9.3 shows the parameters for each site link in the Reskit site topology.

Table 9.3 Site Link Parameters for Reskit Site Topology

Site Link	Transport	Cost	Polling Interval	Schedule
SEA01-YYZ14	SMTP	100	30 mins	0500 to 0900 UTC daily
SEA01-CAI10	IP	100	30 mins	2000 to 0400 UTC daily
SEA01-LHR03	IP	25	1 hr	(always)
LHR03-CAI10	IP	50	15 mins	2000 to 0400 UTC daily

Replication is scheduled to occur only during off-hours for the link between the manufacturing plant and headquarters. Replication is also scheduled for off-hours only between the regional office and other sites. Since the link cost between the regional office and the operations center is lower than the cost between the regional office and headquarters, the KCC attempts to make connections with bridgeheads in the operations center before making connections with bridgeheads in headquarters. The schedule for the link between headquarters and the operations center is wide open, but uses a longer polling interval to reduce traffic.

Placing Servers into Sites

The location of servers on your site topology has a direct effect on the availability of Active Directory. During the physical partitioning exercise of the domain plan, you created a basic plan for domain controller placement. By placing servers onto the site topology, you will complete the details of this plan.

Placing Additional Domain Controllers

During the partitioning exercise, you decided which sites would have domain controllers for each domain, but you did not decide on the number of domain controllers that would be placed in each site for each domain. The number of domain controllers you will create for a particular domain is driven by two factors: fault tolerance requirements and load distribution requirements.

For each domain, use the following guidelines to determine if more domain controllers are necessary:

Always create at least two domain controllers. Even for small domains with small user populations, create at least two domain controllers so that there is no single point of failure for the domain.

For each site that contains a single domain controller, decide if you trust the WAN for failover.

If the single domain controller fails, clients in the site can be serviced by other domain controllers for that domain that are located in other sites. If network connectivity is unreliable or intermittently available, you might not want to trust the network to handle failover. In that case, place a second domain controller for that domain into the site.

Place additional domain controllers for a domain into a site to handle the client workload.

The number of clients that a particular server can handle depends on the workload characteristics and the hardware configuration of the server. Clients randomly select from the available domain controllers in a site to distribute client load evenly.

Placing Global Catalog Servers

The availability of global catalog servers is crucial to the operation of the directory. For example, a global catalog server must be available when processing a user log on request for a native-mode domain, or when a user logs on with a user principal name.

Note When processing a log on request for a user in a native-mode domain, a domain controller sends a query to a global catalog server to determine the user's universal group memberships. Since groups can be explicitly denied access to a resource, complete knowledge of a user's group memberships are necessary to enforce access control correctly. If a domain controller of a native-mode domain cannot contact a global catalog server when a user wants to log on, the domain controller refuses the log on request.

As a general rule, designate at least one domain controller in each site as a global catalog server.

Use the same failover and load distribution rules that you used for individual domain controllers to determine whether additional global catalog servers are necessary in each site.

Note In a single domain environment, global catalog servers are not required to process a user log on request. However, you should still designate global catalog servers using the suggested process. Clients still seek global catalog servers for search operations. Also, having global catalog servers already in place allows the system to adapt gracefully if you add more domains later.

Placing DNS Servers

The availability of DNS directly affects the availability of Active Directory. Clients rely on DNS to be able to find a domain controller, and domain controllers rely on DNS to find other domain controllers. Even if you already have DNS servers deployed on your network today, you might need to adjust the number and placement of servers to meet the needs of your Active Directory clients and domain controllers. As a general rule, place at least one DNS server in every site. The DNS servers in the site should be authoritative for the locator records of the domains in the site, so that clients do not need to query DNS servers off-site to locate domain controllers that are in a site. Domain controllers will also periodically verify that the entries on the primary master server for each locator record are correct.

A simple configuration that satisfies all requirements is to use Active Directory–integrated DNS, store the locator records for a domain within the domain itself, and run the Windows 2000 DNS service on one or more domain controllers for each site where those domain controllers appear.

Distributing the Forest Wide Locator Records

Each domain controller in the forest registers two sets of locator records: a set of domain-specific records that end in *<DNS-domain-name>*, and a set of forest-wide records that end in *_msdcs.<DNS-forest-name>*. The forest-wide records are interesting to clients and domain controllers from all parts of the forest. For example, the global catalog locator records, and the records used by the replication system to locate replication partners, are included in the forest-wide records.

For any two domain controllers to replicate between each other, including two domain controllers from the same domain, they must be able to look up forest-wide locator records. In order for a newly created domain controller to participate in replication, it must be able to register its forest-wide records in DNS, and other domain controllers must be able to look up these records. For this reason, it is important to make the forest-wide locator records available to every DNS server in every site.

To do this, create a separate zone called *_msdcs.<DNS-forest-name>*, and replicate that zone to every DNS server. If you are using the simple Active Directory-integrated configuration, you can place the primary copy of this zone in the forest root domain along with the *<DNS-forest-name>* zone. You can then replicate the zone to DNS servers outside the domain using standard DNS replication.

Generally, it is not sufficient to replicate the zone to only one DNS server per site. If a DNS server does not have a local copy of the *_msdcs.<DNS-forest-name>* zone, it must use DNS recursion to look up a name in that zone. For a DNS server to perform recursion, it contacts a DNS server that is authoritative for the root of the namespace (a DNS root server) and proceeds down the delegations in DNS until it finds the record in question. If there is no DNS root server in a site, and the links between that site and other sites are down, a DNS server cannot perform recursion. Thus, it will not be able to find any DNS servers that are authoritative for *_msdcs.<DNS-forest-name>*, even if those DNS servers are in the same site.

DNS Client Configuration

Clients and domain controllers should be configured with at least two DNS server IP addresses: a preferred local server, and an alternate server. The alternate server can be in the local site, or it can be remote if you trust your network to handle the failover.

Changing the Site Topology After Deployment

A forest site topology is very flexible and easily changed after your initial deployment. As your physical network evolves, remember to evaluate and tune your site topology. As changes to your network increase or decrease bandwidth or reliability, remember to create or remove sites and site links, and be sure to tune site link parameters to balance replication latency against bandwidth usage.

Before making a change to your site topology, anticipate the impact of the change on availability, replication latency and replication bandwidth, and how that might affect end users. Because site topology is stored in the Configuration container, changes will replicate to every domain controller in the forest. Frequent changes to the site topology will cause a large amount of replication traffic, so changes should be rolled up into fewer, larger changes instead of many, smaller changes. Depending on your replication topology and schedule, site topology changes can take a long time to reach every domain controller in the forest.

Planning Task List for Designing the Active Directory Structure

Use Table 9.4 as a checklist to be sure you have performed all the primary tasks necessary to design your Active Directory structure.

Table 9.4 Active Directory Planning Task List

Task	Location in Chapter
Determine the number of forests.	Creating a Forest Plan
Create a change control policy for each forest.	Creating a Forest Plan
Determine the number of domains in each forest.	Creating a Domain Plan
Choose a forest root domain.	Creating a Domain Plan
Assign a DNS name to each domain.	Creating a Domain Plan
Plan DNS server deployment.	Creating a Domain Plan
Optimize authentication with shortcut trust relationships.	Creating a Domain Plan
Create OUs to delegate administration.	Creating an Organizational Unit Plan
Create OUs to hide objects.	Creating an Organizational Unit Plan
Create OUs for Group Policy.	Creating an Organizational Unit Plan
Define sites and site links.	Creating a Site Topology Plan
Place servers into sites.	Creating a Site Topology Plan

