Windows 2000 Directory Services Infrastructure

Study Guide
For Exam
70-219

Table of Contents

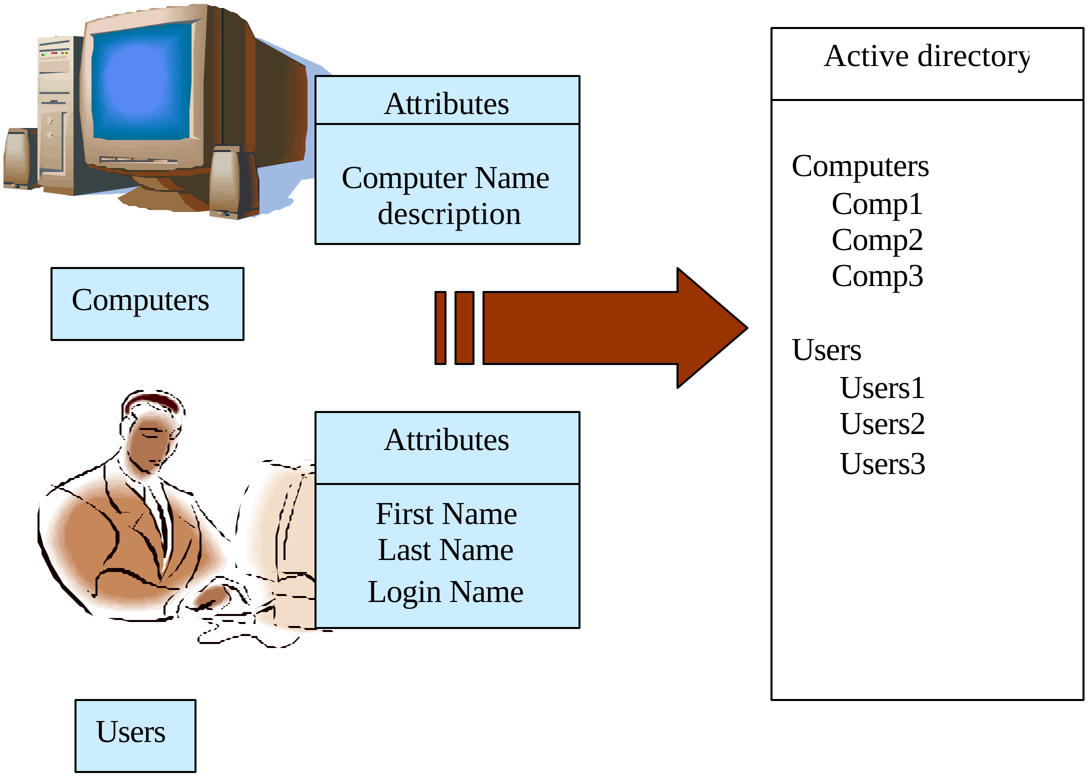
Introduction to Active Directory

Active Directory provides a single point of network resource management, allowing you to add, remove, and relocate users and resources easily.

A directory stores information related to the network resources to facilitate locating and managing of these resources. A *directory service* is a network service that identifies all resources on a network and makes them accessible to users and applications.

The information about user data, printers, servers, databases, groups, computers, and security policies stored in the directory, is organized into objects.

- Objects : An *object* is a distinct named set of attributes that represents a network resource. Object *attributes* are characteristics of objects in the directory. Some objects, known as *containers,* can contain other objects.



- Schema : The Active Directory schema defines objects that can be stored in Active Directory. The schema is a list of definitions that determines the kinds of objects and the types of information about these objects that are stored in Active Directory.

The schema contains two types of definition objects: schema class objects and schema attribute objects.

1. *Schema class objects* : It describes the possible Active Directory objects that can be created. A schema class object functions as a template for creating new Active Directory objects.

2. *Schema attribute objects*: It defines the schema class objects with which Schema attribute objects are associated. Each schema attribute is defined only once and can be used in multiple schema classes.

Note - schemas cannot be deleted, but only deactivated, and a schema is automatically replicated.

- Components : Active Directory uses components to build a directory structure

  Directory structure is divided into two types

  1. ***Logical structure*** : It is represented by the following Active Directory components:

     I. **Domains** – A *domain* can store millions of objects. objects are items the networked community needs to do its job eg. printers, documents, e-mail addresses, databases, users, distributed components, and other resources. All network objects exist within a domain, and each domain stores information only about the objects it contains. Active Directory is made up of one or more domains. A domain can span more than one physical location.

     Characteristics of domain

     - All network objects exist within a domain, and each domain stores information only about the objects that it contains. Theoretically, a domain directory can contain up to 10 million objects.
     - A domain is a security boundary. Access control lists (ACLs) control access to domain objects. ACLs contain the permissions associated with objects that control which users can gain access to an object and what type of access users can gain to the objects.

     II. **Organizational Units** - An *organizational unit* (OU) is a container used to organize objects within a domain into a logical administrative group. OUs provide a means for handling administrative tasks, such as the administration of users and resources, as they are the smallest scope to which administrative authority can be delegated.

III. **Trees** - A *tree* is a grouping or hierarchical arrangement of one or more Windows 2000 domains, created by adding one or more child domains to an existing parent domain.

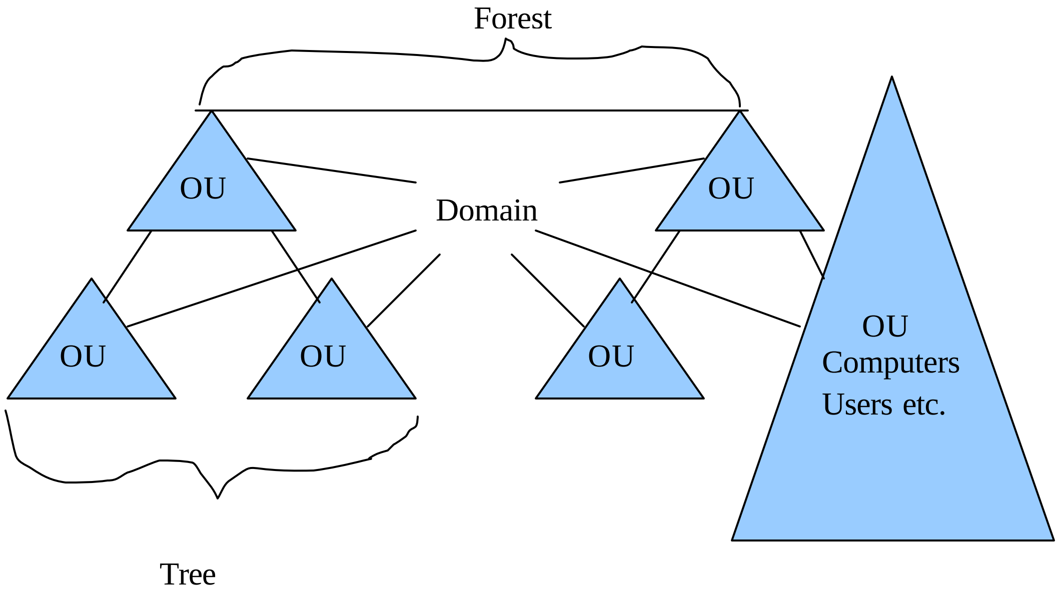
Characteristics of trees are

- It follows Domain Name System (DNS) standards
- All domains within a single tree share a common schema
- All domains within a single tree share a common global catalog

IV. **Forests** - A *forest* is a grouping or hierarchical arrangement of one or more separate, completely independent domain trees.

Characteristics of Forests are

- All trees in a forest share a common schema.
- Trees in a forest have different naming structures, according to their domains.
- All domains in a forest share a common global catalog.
- Domains in a forest operate independently, but the forest enables communication across the entire organization.
- Implicit two-way transitive trusts exist between domains and domain trees.



Logical structure of resources

2. ***Physical structure***: is represented by the following Active Directory components:

I. **Sites** - A *site* is a combination of one or more Internet Protocol (IP) subnets connected by a highly reliable and fast link to localize as much network traffic as possible. Sites contain only computer objects and connection objects used to configure replication between sites. A single domain can span one or multiple geographical sites, and a single site can include user accounts and computers belonging to multiple domains.

II. **Domain Controllers** - A *domain controller* is a computer running Windows 2000 Server that stores a replica of the domain directory (local domain database). Because a domain can contain one or more domain controllers, each domain controller in a domain has a complete replica of the domain's portion of the directory.

Characteristics of domain controllers are

- Each domain controller stores a complete copy of all Active Directory information for that domain, manages changes to that information, and replicates those changes to other domain controllers in the same domain.
- Domain controllers in a domain automatically replicate all objects in the domain to each other.
- Domain controllers immediately replicate certain important updates, such as the disabling of a user account.
- Active Directory uses multimaster replication, in which no one domain controller is the master domain controller.
- Domain controllers detect collisions, which can occur when an attribute is modified on a domain controller before a change to the same attribute on another domain controller in the same domain is completely propagated.
- Having more than one domain controller in a domain provides fault tolerance.
- Domain controllers manage all aspects of users' domain interaction, such as locating Active Directory objects and validating user logon attempts.

There are two domain modes: mixed mode and native mode.

Mixed mode – This mode allows a Windows 2000 domain controller to interact with any domain controllers in the domain that is running previous versions of Windows NT.

Native mode – This mode does not allow any domain controllers in the domain to run previous versions of Windows NT.

- ***Global Catalog***: The *global catalog* is the central repository of information about objects in a tree or forest. By default, a global catalog is created automatically on the initial domain controller in the first domain in the forest, known as the *global catalog server*.

Global Catalog Roles

- It enables network logon by providing universal group membership information to a domain controller when a logon process is initiated.
- It enables finding directory information regardless of which domain in the forest actually contains the data.

A *query* is a specific request made by a user to the global catalog in order to retrieve, modify, or delete Active Directory data.

Windows 2000 Active Directory Concepts

Replication

Replication is a concept which ensures that changes to a domain controller are reflected in all domain controllers within a domain.

A domain controller stores and replicates

- The schema information for the domain tree or forest
- The configuration information for all domains in the domain tree or forest
- All directory objects and properties for its domain

A global catalog stores and replicates

- The schema information for a forest
- The configuration information for all domains in a forest
- Selected attributes for all directory objects in the forest (replicated between global catalog servers only)
- All directory objects and all their properties for the domain in which the global catalog is located

Active Directory replicates information in two ways:

Intrasite (within a site) - A Windows 2000 service known as the Knowledge Consistency Checker (KCC) automatically generates a topology for replication among domain controllers in the same domain using a ring structure. The topology defines the path for directory updates to flow from one domain controller to another until all domain controllers in the site receive the directory updates. The KCC analyzes the replication topology within a site every 15 minutes to ensure that it still works and is efficient.

Intersite (between sites) - To ensure replication between sites, sites must be manually connected by creating *site links*. Site links represent network connections and allow replication to occur. Active Directory uses the network connection information to generate connection objects that provide efficient replication and fault tolerance.

Note - When operating in native mode, Windows 2000 domain controllers does not replicate with pre–Windows 2000 domain controllers.

Trust Relationships

A *trust relationship* is a link between two domains in which the trusting domain honors the logon authentication of the trusted domain. Active Directory supports two forms of trust relationships:

1. **Implicit two-way transitive trust**. A relationship between parent and child domains within a tree and between the top-level domains in a forest. This is the default Trust Relationship. Transitive trust is a feature of the Kerberos authentication protocol, which provides the distributed authentication and authorization in Windows 2000.
2. **Explicit one-way nontransitive trust**. A relationship between domains that are not part of the same tree. A nontransitive trust is bounded by the two domains in the trust relationship and does not flow to any other domains in the forest. Explicit one-way nontransitive trusts are the only form of trust possible between
   - A Windows 2000 domain and a Windows NT domain
   - A Windows 2000 domain in one forest and a Windows 2000 domain in another forest

Group Policy

Group policies are collections of user and computer configuration settings that can be linked to computers, sites, domains, and OUs to specify the behavior of users' desktops. To create a specific desktop configuration for a particular group of users, group policy objects (GPOs) must be created. GPOs are collections of group policy settings. Each Windows 2000 computer has one local GPO and any number of nonlocal (Active Directory–based) GPOs. Local GPOs are overridden by nonlocal GPOs. Nonlocal GPOs are linked to Active Directory objects (sites, domains, or OUs) and can be applied to either users or computers.

Note - Following the inheritance properties of Active Directory, nonlocal GPOs are applied hierarchically.

Group policy settings are applied in the following order:

1. Local GPO. Each Windows 2000 computer has exactly one GPO stored locally.
2. Site GPOs. Any GPO that has been linked to the site are applied next. GPO application is synchronous; the administrator specifies the order of GPOs linked to a site.
3. Domain GPOs. Multiple domain-linked GPOs are applied synchronously; the administrator specifies the order of GPOs linked to a domain.
4. OU GPOs. GPOs linked to the OU highest in the Active Directory hierarchy are applied first, followed by GPOs linked to its child OU, and so on. Finally, the GPOs linked to the OU that contains the user or computer are applied.

There exists some exceptional cases in which default ordering of group policy is overridden

1. A computer that is a member of a workgroup processes only the local GPO.
2. No Override - Any GPO linked to a site, domain, or OU (not the local GPO) can be set to No Override with respect to that site, domain, or OU, so that none of its policy settings can be overwritten.
3. Block Policy Inheritance - At any site, domain, or OU, group policy inheritance can be selectively marked as Block Policy Inheritance. However, GPO links set to No Override are always applied and cannot be blocked.
4. Loopback setting - Loopback provides alternatives to the default method of obtaining the ordered list of GPOs whose user configuration settings affect a user. Loopback can be Not Configured, Enabled, or Disabled as can any other group policy setting. In the Enabled state, loopback can be set to Replace or Merge mode.
    - Replace. The computer's GPOs replace the user GPOs
    - Merge. The GPO list obtained for the local computer at startup is appended to the GPO list obtained for the user at logon.

DNS Namespace

The Active Directory namespace is based on the DNS naming scheme, which allows for interoperability with Internet technologies. Private networks use DNS extensively to resolve computer names and to locate computers within their local networks and the Internet.

DNS provides the following benefits:

- DNS names are user friendly, which means they are easier to remember than IP addresses.
- DNS names remain more constant than IP addresses. An IP address for a server can change, but the server name remains the same.

- DNS allows users to connect to local servers using the same naming convention as the Internet.

Domain Namespace

The *domain namespace* is the naming scheme that provides the hierarchical structure for the DNS database. Each node represents a partition of the DNS database. These nodes are referred to as *domains.*

There are two types of namespaces:

- **Contiguous namespace**. The name of the child object in an object hierarchy always contains the name of the parent domain. A tree is a contiguous namespace.
- **Disjointed namespace**. The names of a parent object and a child of the same parent object are not directly related to one another. A forest is a disjointed namespace.

Root Domain

The *root domain* is at the top of the hierarchy and is represented as a period (.). The Internet root domain is managed by several organizations, including Network Solutions, Inc.

Top-Level Domains

Top-level domains are arranged by organization type or geographic location. Table given below provides some examples of top-level domain names.

Examples of Top-Level Domains

Top-level domain	**Description**
gov	Government organizations
com	Commercial organizations
edu	Educational institutions
org	Noncommercial organizations
net	Commercial sites or networks

Zones

A *zone* is a database containing resource records for a portion of a DNS namespace. Zones provide a way to partition the domain namespace into manageable sections.

Naming Conventions

Every object in Active Directory is identified by a name. Active Directory uses a variety of naming conventions:

1. **Distinguished Name** - Every object in Active Directory has a *distinguished name* (DN) that uniquely identifies the object and contains sufficient information for a client to retrieve the object from the directory.

The DN includes the name of the domain that holds the object, as well as the complete path through the container hierarchy to the object.

e.g.

```
/DC=COM/DC=abc/OU=dev/CN=Users/CN=Firstname Lastname
```

Attribute	**Description**
DC	Domain component name
OU	Organizational unit name
CN	Common name

2. **Relative Distinguished Name** - The *relative distinguished name* (RDN) of an object is the part of the name that is an attribute of the object itself for identification of an object even if the exact DN is unknown or has changed.

Globally Unique Identifier

A *globally unique identifier* (GUID) is a 128-bit number that is guaranteed to be unique within the enterprise. GUIDs are assigned to objects when the objects are created. The GUID never changes, even if object is moved or renamed. Applications can store the GUID of an object and use the GUID to retrieve that object regardless of its current DN.

Designing a Directory Services Infrastructure

An *Active Directory infrastructure design* is a plan which is created to represent organization's network infrastructure. This plan can be used to determine how Active

Directory will be configured to store information about objects in network and make the information available to users and network administrators.

Design Tools

Design team

To ensure that all aspects of organization are addressed in Active Directory implementation, following members must be choosen

- **Infrastructure designers** - the infrastructure designers panel should contain system administrators, network administrators, and members of the information technology management organization.
- **Staff representatives** - The *staff representatives* panel consists of personnel throughout the organization who are responsible for carrying out daily operations. The panel should contain an exemplary staff member from each business unit or department within the organization.
- **Management representatives** - The *management representatives* panel consists of management level personnel who are responsible for approving business decisions within the organization. Management representatives must have the authority and ability to approve and support design decisions made by infrastructure designers at *each stage* of the design development process.

Analyzing Business Environment

- **Products and customers** - Products are sets of tangible or intangible attributes assembled to provide benefits to a customer and can include goods, services, places, persons, and ideas. Customers are the entities that purchase products.
- **Business structure** - A *business structure* represents the daily operating structure of an organization. To determine the current business structure of organization, it must be understandable, how that company conducts daily operations both administratively and geographically.
- **Business processes** - A *business process* is a series of steps that must be taken to achieve a desired result within the organization. To determine the current business processes active in an organization, following processes must be analyzed in an organization,
  - Information flow
  - Communication flow
  - Decision-making processes
- **Factors that influence company strategies** - A *business strategy* is the long-range plan for defining and achieving the objectives set up by an organization.
- **Information technology (IT) management organization** - In an organization, IT management refers to the management of the computing environment, usually performed by the IT, IS (information services) or MIS (management information services) departments.

Analyzing the Technical Environment

- **Network architecture** – includes the
  - Location of points on the network
  - Number of users at each location
  - Network type used at each location
  - Location, link speed, and percentage of average available bandwidth of remote network links
  - TCP/IP subnets at each location
  - Speed of local network links
  - Location of domain controllers
  - List of servers at each location and the services that run on them
  - Location of firewalls in the network
- **Hardware** - hardware inventory should include the name of each device and the manufacturer's name and model number. Depending on the device type, different type of information is included like processor type, memory, or disk capacity. The types of devices should be audio or sound cards, computers, cameras or digital cameras, CD-R/RW, controller cards, DVD, input devices, modems, monitors, networking, printers, scanners, smart card readers, storage, TV tuners, uninterruptible power supply (UPS), USB/ 1394, video, and any other devices that are installed.
- **Software** - software inventory should include the name of the product, the version number, the manufacturer's name, and the language (for example, English or French) used in the software. Depending on the software, more information about the software may be included such as whether it's a service pack or patch release. The categories of software products will depend on the individual needs of the company, but in general they include arts & entertainment, commerce, connectivity and communications, cross-platform tools/integration, data processing, data warehousing, multimedia, network infrastructure, operating systems, system management, user interface enhancements and accessibility, utilities and servers, and workflow and conferencing.
- **Technical standards** - Technical standards usually include
  - Standard hardware configurations for desktops, servers, and other devices
  - Standard software configurations for user desktops
  - Naming conventions for users, groups, devices, and domains
  - Network performance standards
  - Security standards
- **DNS environment** (if applicable) - The minimum requirement for a DNS service to be compatible with Active Directory is for the service to support service resource records (SRV RRs, and dynamic update.

  Note – If DNS environment is BIND (Berkeley Internet Name Domain) version 8.1.2 or laterand Windows NT 4 DNS supports SRV RRs and dynamic update and are compatible with Active Directory DNS requirements, but cant be used as as the data storage and replication engine.

- **Windows NT domain architecture** (if applicable) - In Windows NT, users and servers can be grouped into domains for administrative purposes. In Windows 2000, organizational units (OUs) have been introduced to handle administration, while domains still provide administration but hold OUs and many more objects than in Windows NT. The purpose of analyzing the current Windows NT domain architecture employed in organization is to understand the workings of the present domain structure so that conversion of each of the domains into an Active Directory domain, tree, and forest structure can be done.

Design Process

the Active Directory infrastructure design process consists of four stages:

1. Creating a Forest Plan
2. Creating a Domain Plan
3. Creating an Organizational Unit Plan
4. Creating a Site Topology Plan

Creating a Forest Plan

To design the forest model for organization, the following tasks must be performed:

1) **Assess the organization's forest needs** - To design a forest model the business and technical environment analysis documents must be analysed by design team:

    - Business Structures. Assess the current administrative structure of an organization.
    - IT Management Organization. Assess current structure and administration practices in an organization's IT management organization.
    - Technical Standards. Assess current administrative and security standards.

2) **Determine the number of forests for an organization** - Windows 2000 domains in a forest share a single schema, configuration container, and global catalog and are linked by two-way transitive trusts. So only one forest is enough for an organization. Ideally, the use of multiple forests should be temporary, and reserved for situations such as a merger, acquisition, or partnership where two or more organizations must be joined.

    Multiple Forests can be used in following situations

- Network administration is separated into autonomous groups that do not trust each other.
- Business units are politically separated into autonomous groups.
- Business units must be separately maintained.
- There is a need to isolate the schema, configuration container, or global catalog.
- There is a need to limit the scope of the trust relationship between domains or domain trees.

Adding a forest increases administrative and usability costs like

- Schema. Each forest has its own schema. The contents and administration group memberships for each schema must be maintained separately even if they are similar.
- Configuration container. Each forest has its own configuration container. The contents and administration group memberships for each configuration container must be maintained separately even if they are similar.
- Trusts. An explicit one-way nontransitive trust is the only trust relationship permitted between domains in different forests.
- Replication. Replication of objects between forests is manual and requires the development of new administrative policies and procedures.
- Merging forests or moving domains. Forests cannot be merged in a one-step operation;various steps are
    - clone security principals,
    - migrate objects,
    - decommission domain controllers,
    - downgrade them to member servers,
    - and add each to the new forest domain.
- Moving objects. Although objects can be moved between forests, The ClonePrincipal tool must be used to clone security principals in the new forest, or the LDAP Data Interchange Format (LDIFDE.EXE) command-line tool to move other objects.
- Smart card logon. Default UPNs must be maintained for smart cards to be able to log on across forests.
- Additional domains. Each forest must contain at least one domain. Additional domains increase hardware and administrative costs.

Schema Modification

The schema is stored in the schema table as part of the NTDS.DIT file. There are two types of objects in the schema: schema class (classSchema) objects and schema attribute (attributeSchema) objects.

A basic set of schema classes and attributes, often called the *base schema* or base *directory information tree* (DIT), is shipped with Windows 2000 Server. There are nearly 200 schema class objects and more than 900 schema attribute objects provided in the base schema.

If the base schema doesn't meet the needs of an organization, The organization must consider modifying the schema or creating additional schema class and/or attribute objects; this process is called *extending* the schema. Because schema that is added cannot be deleted, but only deactivated, and a schema is automatically replicated, si it must be planned and prepared carefully before extending the schema. Inconsistencies in the schema brought about by modifications can cause problems that may impair or disable Active Directory.

To view the base schema in Windows 2000,

- install all of the Windows 2000 administration tools,
- install the Active Directory Schema snap-in
- add the snap-in to Microsoft Management Console (MMC) by using the Add/Remove Snap-in dialog box accessible from the Console menu

The Schema Admins predefined universal group is the only group authorized to make changes to the Active Directory schema.

Designing a Schema Modification Plan

To plan schema modifications following steps must be performed

1. Create a schema modification policy - A *schema modification policy* is a written plan you create to administer schema modifications that affect the entire forest. It outlines who has control of the schema and how modifications are administered.

   To create a schema modification policy

   - List the entity (division, department) that controls the Schema Admins predefined universal group.
   - List the members of the Schema Admins predefined universal group.
   - Appoint a schema modification approval committee. List the members.
   - List the steps required for initiating a schema modification.
   - List the steps required for testing a schema modification.
   - List the steps required for implementing a schema modification.

2. Assess the organization's schema needs – in an organization changes currently planned to address growth and flexibility needs and any other changes that would help meet the ideal design specifications for the organization would require schema modifications.

3. Determine whether to modify the schema - Active Directory schema contains hundreds of the most common object classes and attributes that users of a server system require, the need to change the schema is rare. However, some organizations may require object classes or attributes not anticipated in the default schema.

   The following modifications can be made to the schema:

   - Create a new class
   - Modify an existing class
   - Deactivate classes
   - Create a new attribute
   - Modify an existing attribute
   - Deactivate attributes

Automatic Schema Modification

The schema will be modified automatically if you choose to install a directory-enabled application. A *directory-enabled application* is software that has the capability to read Active Directory objects (and their attributes) or has the capability to create schema class or attribute objects. These capabilities allow the application to integrate directly with Active Directory, combining services and reducing the total cost of ownership and network overhead.

Modifying the Schema may result in certain inconsistencies like

- Existing object instances. If a schema attribute object is added to or removed from a schema class object, any existing instances of the class object become invalid because they no longer match the class definition.
- Replication. Schema modiafication can cause temporary inconsistencies in the schema that will result in replication failure if an instance of a newly created class object is replicated to a domain controller before the newly created class.
- Network traffic. The schema modification and then choosing to replicate attributes to the global catalog can negatively affect network performance during replication. Replicating attributes to the global catalog causes all global catalogs to replicate all objects, not just the modified schema attributes, and significantly increases network traffic.

Creating a Domain Plan

In Active Directory services, a domain is a partition of a forest, or a partial database. Each domain has a unique name and provides access to centralized user accounts and group accounts maintained by the domain administrator. Active Directory is made up of one or more domains, each of which can span one or more sites.

There are two goals which should be kept in mind when defining the domains for an organization:

- Define domains based on the organization's geographical structure –

    - Business Structures. Assess current administrative and geographical structure to determine possible domain locations.
    - Network Architecture. Assess current network architecture to determine possible domain locations.
    - Technical Standards. Assess current administrative and security standards to determine need for domains.
    - Hardware & Software . Assess the hardware devices and software that are not compatible with Windows 2000.
    - Windows NT Domain Architecture. Assess current domain structure to determine ways to consolidate domains.
    - Forest model. Assess the number of forests planned for the organization to determine domain locations.

- Minimize the number of domains - Whenever possible, it's best to limit infrastructure design to one domain that is administered through organizational units (OUs). Adding domains to the forest increases management and hardware costs.

    The principles for defining multiple domains in Windows NT than in Windows2000 are different which are

    - Security Accounts Manager (SAM) size limitations. In Windows NT, the SAM database had a limitation of about 40,000 objects per domain. In Windows 2000, each domain can contain more than one million objects, so it is no longer necessary to define a new domain just to handle more objects.
    - Primary Domain Controller (PDC) availability requirements. In Windows NT, only one computer in a domain, the PDC, could accept updates to the domain database. In Windows 2000, all domain controllers accept updates, eliminating the need to define new domains just to provide fault tolerance.
    - Limited delegation of administration within domains. In Windows NT, domains were the smallest units of administrative delegation. In Windows 2000, OUs allow you to partition domains to delegate administration, eliminating the need to define domains just for delegation.

These are the reasons to consider using multiple domains:

- To meet security requirements
- To meet administrative requirements
- To optimize replication traffic
- To retain Windows NT domains

Forest Root Domain

A *forest root domain* is the first domain which is created in an Active Directory forest. The forest root domain must be centrally managed by an IT organization that is responsible for making domain hierarchy, naming, and policy decisions. The Enterprise Admins and Schema Admins predefined universal groups reside only in this domain. Administrators in this domain are those who are key to the network design.

To define a forest root domain, the following tasks must be taken into account:

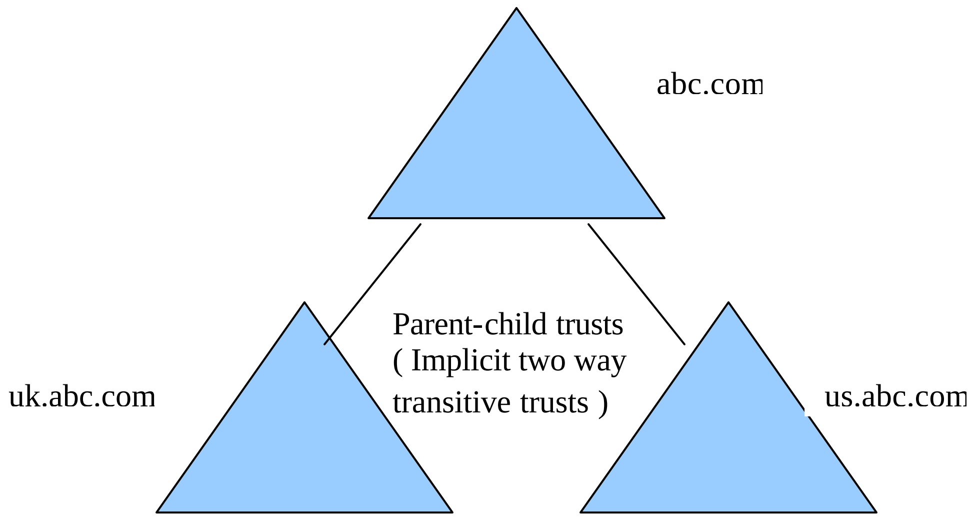
1. Assess the domains defined for the organization and its IT management organization - To define an organization's forest root domain, the following documents must be consulted

    - Business Structures. Assess current administrative structure to locate the IT management organization.
    - Network Architecture. Assess current network architecture and the domains that have been defined.
    - IT Management Organization. Assess current IT management organization structure and analyze how the IT management organization handles decisions and changes to determine the location of the forest root.

2. Choose a forest root domain for an organization – forest root domain may be an
    1. an existing domain as the forest root domain
    2. an additional, dedicated domain to serve as the forest root domain

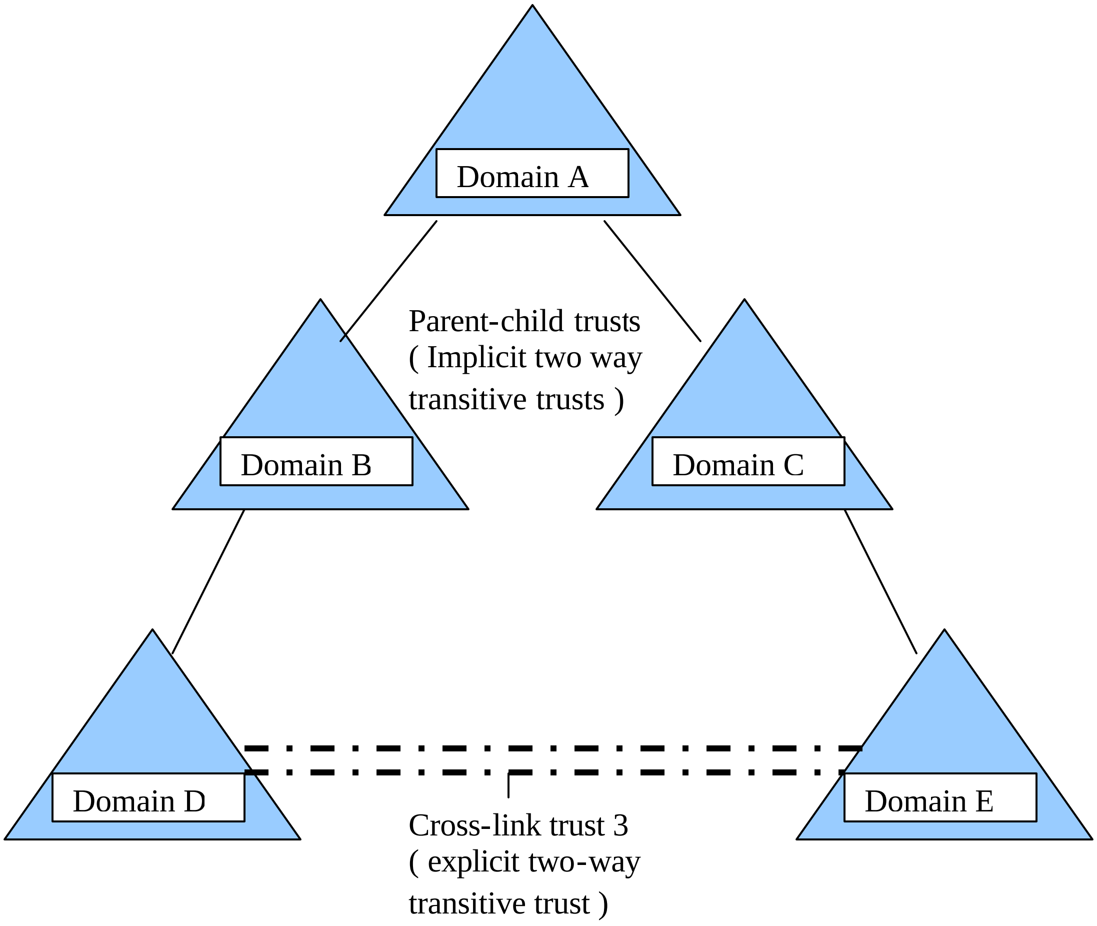
Defining a Domain Hierarchy

A domain hierarchy is a tree structure of parent and child domains. The way in which domains are arranged in a hierarchy determines the trust relationships between domains. Windows 2000 creates parent-child trusts between parent and child domains in a forest or tree hierarchy. *Parent-child trusts* are implicit, two-way transitive trusts that are created automatically when a domain is added to the hierarchy.



Parent-child trusts

Windows 2000 provides a means for improving query response performance with cross-link trusts. A *cross-link trust* is a two-way transitive trust that you explicitly create between two Windows 2000 domains that are logically distant from each other in a forest or tree hierarchy in order to optimize the interdomain authentication process. Cross-link trusts are also known as *shortcut trusts*. Cross-link trusts can be created only between Windows 2000 domains in the same forest.



To define a hierarchy of domains for an organization, following tasks must be performed:

1. Assess the organization's domain hierarchy needs – can be compiled by using the following documents
   - Information Flow. Analyze which domains need access to resources in other domains.
   - Network Architecture. Assess current network architecture, including the domains defined and the location of the forest root domain.
   - DNS Environment. Assess current domain structure to determine existing DNS names that may require separate tree structures.
2. Determine the domain hierarchy - To determine a domain hierarchy, the information about the organization must be analyzed to determine
   1. the number of domain trees,
   2. designate tree root domains,

3. arrange the subdomain hierarchy,
4. and plan any cross-link trusts.

Deciding Domain Names

In Windows 2000 and Active Directory, a *domain name* is a name given to a collection of networked computers that share a common directory. Active Directory uses the Domain Name System (DNS) as its domain naming and location service, which allows for interoperability with Internet technologies. Therefore, Windows 2000 domain names are also DNS names. When requesting logon to the network, Active Directory clients query their DNS servers to locate domain controllers.

In DNS, names are arranged in a hierarchy and can be partitioned according to the hierarchy. The hierarchy allows parent-child relationships where the name of the child domain is designated by the name of the parent domain preceded by a label.

It is nearly impossible to change domain names, so the domain name selection are very important. The following are guidelines for naming domains:

- Use of Internet standard characters - Internet standard characters are defined as: A-Z, a-z, 0-9, and the hyphen (-). Although Windows 2000 DNS supports the use of almost any Unicode character in a name, by using only Internet standard characters, Active Directory domain names will be compatible with other versions of DNS.
- Differentiate between internal and external namespaces - Most organizations have an Internet presence, so they should use different names for the internal and external root domains to clearly delineate public resources from private resources and prevent unauthorized users from accessing resources on the internal network
- Never use the same domain name twice.
- Use only registered domain names. Register all second-level domain names, whether they are internal or external namespaces, with the InterNIC or other authorized naming authority.
- Use short, distinct, meaningful names. Use domain names that are easy to use and are representative of an organization's identity.
- Use names that have been reviewed internationally. Review domain names to ensure that they are not derogatory or offensive in another language.
- Use names that will remain static. Use generic names rather than specific ones.
- Use the International Standards Organization (ISO) standards for names that include countries and U.S. states. The ISO defines two-letter country codes and U.S. state codes, as presented in ISO 3166.

DNS Server Deployment

A *DNS server* is a computer that resolves names to IP addresses and IP addresses to names for host devices contained within a portion of the namespace. When a client queries a DNS server for a name or IP address, the server performs one of the following actions: provides the name or IP address, refers the client to another DNS server, or indicates that it cannot fulfill the request. DNS servers are also known as *DNS name servers*.

DNS servers use information stored about zones to handle name resolution. Each DNS server can store information for no zones, one zone, or multiple zones. A *zone* is a contiguous portion of the DNS namespace that is administered separately by a DNS server. Each zone contains a *zone database file*, a text file containing resource records for the zone. *Resource records* are records that contain information used to process client queries. There are many different types of resource records. When a zone is created, DNS automatically adds two resource records: the Start of Authority (SOA) and the Name Server (NS) records. Table given below describes these resource record types, along with the most frequently used resource records.

Resource record type	**Description**
Host (A)	Lists the host name to IP address mappings for a forward lookup zone.
Alias (CNAME)	Creates an alias, or alternate name, for the specified host name. Canonical Name (CNAME) record can be used to define more than one name to point to a single IP address.
Host Information (HINFO)	Identifies the CPU and operating system used by the host. Use this record as a low-cost resource-tracking tool.
Mail Exchanger (MX)	Identifies which mail exchanger to contact for a specified domain and in what order to use each mail host.
Name Server (NS)	Lists the name servers that are assigned to a particular domain.
Pointer (PTR)	Points to another part of the domain namespace. For example, in a reverse lookup zone, it lists the IP-address-to-name mapping.
Service (SRV)	Identifies which servers are hosting specific services.
Start of Authority (SOA)	Identifies which name server is the authoritative source of information for data within this domain. The first record in the zone database file must be the SOA record.

Zone Replication

Zone replication is the synchronization of DNS data between DNS servers within a given zone. Replicating zones provides the following benefits:

- Fault tolerance. If a DNS server fails, clients can still direct queries to other DNS servers.
- Query load distribution. Query loads can be balanced among DNS servers.
- WAN traffic reduction. DNS servers can be added in remote locations to eliminate the need for clients to send queries across slow links.

There are two methods for replicating zones

1. Standard zone replication - In standard zone replication, primary and secondary zones and primary and secondary DNS servers handle zone replication. A *primary zone* is the master copy of a zone stored in a standard text file on a primary DNS server. A *primary DNS server* is the authoritative server for a primary zone. A *secondary DNS server* is a backup DNS server that receives the zone database files from the primary server in a zone transfer. *Zone transfer* is the process by which DNS servers interact to maintain and synchronize authoritative name data. A zone can have multiple secondary servers, and a secondary server can serve more than one zone.

   There are three types of zone transfers

   - full zone transfers - In a *full zone transfer (AXFR query)*, the primary DNS server transmits the entire zone database file for the primary zone to the secondary DNS server
   - incremental zone transfers - an *incremental zone transfer (IXFR query)*, the servers keep track of and transfer only incremental resource record changes between each version of the zone database file.
   - transfers that use the DNS Notify process - In the DNS Notify process, the primary server, rather than the secondary server, initiates the zone transfer.

2. Active Directory zone replication - In Active Directory zone replication, Active Directory—integrated zones and domain controllers handle zone replication. Each domain controller functions as a primary DNS server, using Active Directory to store and replicate primary zone files. Active Directory zone replication provides the following advantages over standard zone replication:
   - Replication planning is simplified. Because DNS resource records are part of Active Directory and are replicated to each domain controller, it is no longer necessary to maintain zone database files or use zone transfer.
   - Replication is multimaster. Updates to zones are allowed at every DNS server/domain controller, rather than just the primary DNS server.

- Efficiency. Because Active Directory zone replication is processed at the property level, it generates less replication traffic than standard zone replication.
- Detailed delegation of administration. Administration for directory-integrated zone data can be delegated for users for each resource record.

Defining organizational unit (OU) Structures

An *organizational unit (OU)* is a container used to organize objects within one domain into logical administrative groups. An OU can contain objects such as user accounts, groups, computers, printers, applications, file shares, and other OUs from the same domain. There are three reasons for defining an OU:

- To delegate administration - *Delegating administration* is the assignment of IT management responsibility for a portion of the namespace, such as an OU, to an administrator, a user, or a group of administrators or users. An *access control list* (ACL) is the mechanism for limiting access to certain items of information or certain controls based on users' identity and their membership in various groups. *Access control entries* (ACEs) in each ACL determine which users or groups can access the OU and what type of access they have. Because ACEs are inherited by child OUs in an OU hierarchy by default, permissions can be applied to an entire tree of Ous, In order to prevent permissions from being inherited by child objects, the Allow Inheritable Permissions From Parent To Propagate To This Object check box must be cleared on the Security tab of the Properties dialog box for the OU.
- To hide objects – some organization may require that certain domain objects be hidden from certain users. Although a user may not have the permission to read an object's attributes, the user can still see that the object exists by viewing the contents of the object's parent container. Objects can be hidden in a domain by creating an OU for the objects and limiting the set of users who have List Contents permission for that OU.
- To administer group policy - *group policies* are collections of user and computer configuration settings that can be linked to computers, sites, domains, and OUs to specify the behavior of users' desktops. To create a specific desktop configuration for a particular group of users, *group policy objects* (GPOs) must be created, which are collections of group policy settings. By linking GPOs to OUs, GPOs can be applied to either users or computers in the OU.

  Group policy settings are processed in the following order:

  1. Local GPO
  2. Site GPOs
  3. Domain GPOs
  4. OU GPOs

Group Policy Inheritance

In general, group policy is passed down from parent to child containers. Group policy is inherited in the following ways:

- If a policy is configured for a parent OU, and the same policy *is not* already configured for its child OUs, the child OUs, which contain the user and computer objects, inherit the parent's policy setting.
- If a policy is configured for a parent OU, and the same policy *is* configured for a child OU, the child OU's group policy setting overrides the setting inherited from the parent OU.
- Policies are inherited as long as they are compatible.
- If a policy configured for a parent OU is incompatible with the same policy configured for a child OU, the child OU does not inherit the policy setting from the parent OU. Only the setting configured for the child OU is applied.
- If any of the policy settings of a parent OU are disabled, the child OU inherits them as disabled.
- If any of the policy settings of a parent OU are not configured, the child OU does not inherit them.

OUs can be added to other OUs to form a hierarchical structure

User Accounts and Groups

Groups contain objects such as user accounts, contacts, computers, and other groups. However, groups are defined mainly to assign permissions to users or to restrict user access to various objects in the domain.

User Accounts

A domain *user account* provides a user with the ability to log onto the domain to gain access to network resources. Each person who regularly uses the network should have a unique user account.

Groups

A *group* is a collection of user accounts. Groups simplify administration by allowing system administrator to assign permissions to a group of users rather than having to assign permissions to each individual user account.

Windows 2000 includes two group types: *security* and *distribution*. Windows 2000 uses only security groups, which you use to assign permissions to gain access to resources.

The scope of a group determines where in the network that group will be accessible to assign permissions to the group. The three group scopes are global, domain local, and universal.

Global security groups are most often used to organize users who share similar network access requirements. A global group has the following characteristics:

- Limited membership. Only global group members can be added.
- Access to resources in any domain. Global group is used to assign permissions to gain access to resources that are located in any domain in the domain tree or forest.

Domain local security groups are most often used to assign permissions to resources. A domain local group has the following characteristics:

- Open membership. Members from any domain can be added.
- Access to resources in one domain. Domain local group is used to assign permissions to gain access only to resources located in the same domain where you create the domain local group.

Universal security groups are most often used to assign permissions to related resources in multiple domains. A universal security group has the following characteristics:

- Open membership. Members from any domain can be added.
- Access to resources in any domain. Universal group can be added to assign permissions to gain access to resources that are located in any domain.
- Available only in native mode. Universal security groups are not available in mixed mode.

Group Scope Membership Rules

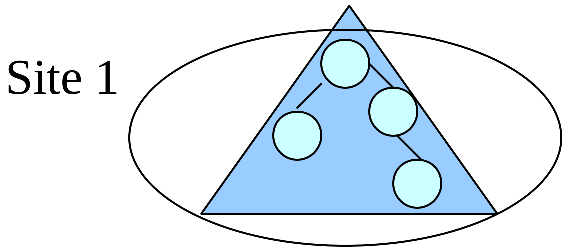
Group scope	**In native mode, group can contain**	**In mixed mode, group can contain**
Global	User accounts and global groups from the same domain	Users from the same domain
Domain local	User accounts, universal groups, and global groups from any domain; domain local groups from the same domain	User accounts and global groups from any domain
Universal	User accounts, other universal groups, and global groups from any domain	Not applicable; universal groups cannot be created in mixed mode

Creating a Site Topology Plan

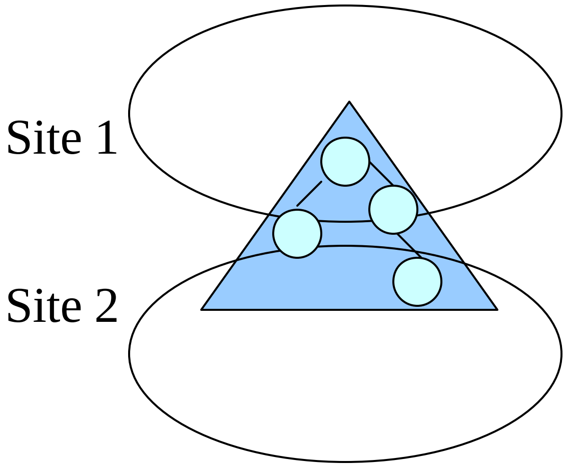
Sites

a *site* is a set of Internet Protocol (IP) subnets connected by a highly reliable and fast link (usually a LAN). The main purpose of a site is to physically group computers to optimize network traffic. Sites act to confine authentication and replication traffic to only the devices within a site. Because network traffic is prevented from unnecessarily crossing slow WAN links, traffic is limited. Sites have two main roles:

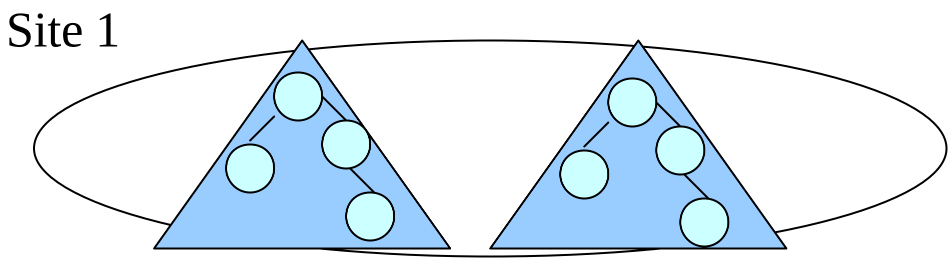
- To determine the nearest domain controller during workstation logon
- To optimize the replication of data between sites



A Single Domain with a single site



A Single Domain with multiple sites



Multiple domains with a single site

Relationship of site and domain structures

To define sites

1. Determine the site(s) needed to encompass each
   - LAN or set of LANs that are connected by a high-speed backbone
   - Location that does not have direct connectivity to the rest of the network and is only reachable using SMTP mail
2. Create a site diagram:
   - Use an oval to represent each site.
   - Name each site using a valid DNS name.
   - List the set of IP subnets that constitute each site.

Placement of Domain Controllers in Sites

Domain controller is a computer running Windows 2000 Server that authenticates user logons and maintains the security policy and the master database for a domain. Because the availability of Active Directory depends on the availability of domain controllers, a domain controller must always be available so that the users can be authenticated.

For optimum network response time and application availability, place at least

- One domain controller in each site

  A domain controller in each site provides users with a local computer that can service query requests for their domain over LAN connections.

- Two domain controllers in each domain

  By placing at least two domain controllers in each domain, you provide redundancy and reduce the load on the existing domain controller in a domain.

To determine the number of domain controllers you need, you may want to use Active Directory Sizer, a tool for estimating the hardware required for deploying Active Directory based on your organization's profile, domain information, and site topology.

Replication Strategy

Replication is the process of copying data from a data store or file system to multiple computers to synchronize the data. In Windows 2000, each domain controller maintains a replica of all Active Directory objects contained in the domain to which it belongs. Replication ensures that changes made to a replica on one domain controller are synchronized to replicas on all other domain controllers within the domain.

The following actions trigger replication between domain controllers:

- Creating an object
- Modifying an object
- Moving an object
- Deleting an object

Active Directory replicates information in two ways: *intrasite* (within a site) and *intersite* (between sites).

	Intrasite replication	**Intersite replication**
Compression	To save CPU time, replication data is not compressed.	To save WAN bandwidth, replication data greater than 50 KB is compressed.
Replication model	To reduce replication latency, replication partners notify each other when changes need to be replicated and then pull the information for processing.	To save WAN bandwidth, replication partners do not notify each other when changes need to be replicated.
Replication frequency	Replication partners poll each other periodically.	Replication partners poll each other at specified intervals, only during scheduled periods. If updates are necessary, operations are scheduled to pull the information for processing.
Transport protocols	Remote procedure call (RPC)	IP or SMTP

By default, all site links are transitive, which simply means that if sites A and B are linked and sites B and C are linked, then site A and site C are transitively linked. Site link transitivity is enabled or disabled by selecting the Bridge All Site Links check box in the Properties dialog box for either the IP or the SMTP intersite transport. By default, site link transitivity is enabled for each transport.

A *site link bridge* connects two or more site links in a transport where transitivity has been disabled in order to create a transitive and logical link between two sites that do not have an explicit site link.

A *bridgehead server* is a single domain controller in a site, the contact point, used for replication between sites. The KCC automatically creates connection objects between bridgehead servers. When a bridgehead server receives replication updates from another site, it replicates the data to the other domain controllers within its site.

An organization's replication strategy determines when and how information is replicated. To define a replication strategy, Following tasks must be completed:

1. Assess the physical connectivity of the organization's network.
2. Plan a site link configuration for each network connection.
3. Plan site link transitivity disabling (optional).
4. Plan preferred bridgehead servers (optional).

Global Catalog Servers and Operations Masters

A *global catalog server* is a Windows 2000 domain controller that holds a copy of the global catalog for the forest. A global catalog server must be available when a user logs on to a Windows 2000 native-mode domain or logs on with a user principal name because in native mode a domain controller must send a query to a global catalog server to determine the user's membership in universal groups.

Operations master roles are special roles assigned to one or more domain controllers in an Active Directory domain to allow the domain controllers to perform single-master replication for specific operations. Active Directory supports multimaster replication of the database between all domain controllers in the domain.

Every Active Directory forest must have the following roles:

- Schema master

  The schema master domain controller controls all updates and modifications to the schema. At any time, there can be only one schema master in the entire forest.

- Domain naming master

  The domain controller holding the domain naming master role controls the addition or removal of domains in the forest. At any time, there can be only one domain naming master in the entire forest.

Every domain in the forest must have the following roles:

- Relative ID master

  The relative ID master allocates sequences of relative IDs to each of the various domain controllers in its domain. Whenever a domain controller creates a user, group, or computer object, it assigns the object a unique security ID. The security ID consists of a domain security ID (that is the same for all security IDs created in the domain) and a relative ID that is unique for each security ID created in the domain. At any time, only one domain controller can act as the relative ID master in each domain in the forest.

- Primary domain controller (PDC) emulator

If the domain contains computers operating without Windows 2000 client software or if it contains Windows NT backup domain controllers (BDCs), the PDC emulator acts as a Windows NT primary domain controller. It processes password changes from clients and replicates updates to the BDCs. In a Windows 2000 domain operating in native mode, the PDC emulator receives preferential replication of password changes performed by other domain controllers in the domain. If a password was recently changed, that change takes time to replicate to every domain controller in the domain. If a logon authentication fails at another domain controller due to a bad password, that domain controller will forward the authentication request to the PDC emulator before rejecting the logon attempt. At any time, only one domain controller can act as the PDC emulator in each domain in the forest.

- Infrastructure master

  The infrastructure master is responsible for updating the security identifiers and distinguished names in cross-domain object references whenever the name of an object is renamed or changed. At any time, only one domain controller can act as the infrastructure master in each domain.

To place domain global catalog servers and operations masters, The following tasks must be performed.

1. Locate domain controllers.
2. Determine the location of global catalog servers for the organization.
3. Determine the location of operations masters for the organization.

Active Directory Implementation

Active directory implementation can be performed in gollowing ways

1. ***Implementing from scrach*** – No special requirement to perform this, as previous defined documents can be useful to perform this type of implementation.

2. ***Migration from Windows NT to Windows 2000 platform*** –

   The supported migration paths from Windows NT to Windows 2000 Server.

Server role in Windows NT Server 3.51 or 4	**Server role in Windows 2000 Server**
Primary domain controller (PDC)	Domain controller
Backup domain controller (BDC)	Domain controller or member server
Member server	Member server
Standalone server	Member server or standalone server

There are two methods for migrating to Windows 2000 Server:

- **Domain upgrade** - A *domain upgrade* is the process of installing an existing Windows NT domain structure and its users and groups intact into the Windows 2000 DNS-based domain hierarchy, This method also retains most Windows NT system settings, preferences, and program installations.
- **Domain restructure** - A domain restructure migrates the existing Windows NT environment into a "pristine" Windows 2000 forest using a nondestructive copy. A *pristine forest* is an ideal Windows 2000 forest that is isolated from the Windows NT production environment and that operates in native mode. Domain accounts exist in both Windows NT and Windows 2000, and the Windows NT environment is retained until it is ready to be decommissioned. This method requires more administrative overhead, resources, and time.

The Active Directory Migration Tool (ADMT) is provided to migrate existing Windows NT 4 and earlier domains into Windows 2000.

To plan a Windows NT 4 directory services migration to Windows 2000 Active Directory, following tasks must be performed:

- Assess the organization's goals for migration.
- Determine the migration method(s).
- Plan the migration steps.
- Plan the consolidation of resource domains into OUs, if applicable.

3. ***Synchronization of other directory services with Active Directory*** - Active Directory is designed to extend Windows 2000 interoperability and allows you to synchronize directory information with other directory services. Active Directory is able to synchronize with

- **Microsoft Exchange Server 5.5 directory service**- To set up synchronization between Active Directory and Exchange Server 5.5, Install the Active Directory Connector (ADC). The installation files for

ADC are located on the Windows 2000 Server CD in the Valueadd\Msft\Mgmt\ADC folder. After installation, the Active Directory Connector Management tool is used to set up *connection agreements*, which define how synchronization will occur.

- **Novell NetWare Bindery or Novell Directory Services (NDS)** - To enable users of Novell directory services to implement synchronization, Microsoft developed Microsoft Directory Synchronization Services (MSDSS), which is included with Services for NetWare version 5 (SFNW5). MSDSS is managed by using a Microsoft Management Console (MMC) snap-in, and supports Active Directory synchronization with the following Novell directory services:
    1. NDS for Novell NetWare 4, 4.1, 4.11, 4.2, 5, 5 with NDS 8, and 5.1
    2. Bindery for Novell NetWare 3.1, 3.11, 3.12, and 3.2, as well as NetWare 4.*x* configured in bindery emulation mode.

- **Other LDAP-compliant directory services** - Some organizations have sophisticated directory management needs, including the need to synchronize more than two directory services, For these organizations, Microsoft Metadirectory Services (MMS) is available through a service engagement with trained providers. MMS allows the integration of identity and directory from multiple repositories with Active Directory. This allows organizations to manage diverse information and reduces the cost of directory management. MMS allows the integration of information from platforms such as Microsoft Windows 2000, Microsoft Active Directory, Microsoft Windows NT, Microsoft Exchange, Lotus Notes, Domino, cc: Mail, Novell NDS, Bindery, GroupWise, Netscape Directory and MetaDirectory Server, ISOCOR MetaConnect and X.500, various ODBC/SQL databases, and other systems.