



Microsoft 70-219

**Designing a Microsoft Windows 2000
Directory Services Infrastructure**

Study Guide
DEMO Version

Copyright (c) 2003 Chinatag LLC. All rights reserved.

Important Note Please Read Carefully

For demonstration purpose only, this free version Chinatag study guide contains **10** full length questions selected from our full version products which have more than **200** questions each.

This Study guide has been carefully written and compiled by Chinatag certification experts. It is designed to help you learn the concepts behind the questions rather than be a strict memorization tool. Repeated readings will increase your comprehension.

For promotion purposes, all PDF files are **not** encrypted. Feel free to distribute copies among your friends and let them know Chinatag website. Our IT certification products start at only **\$7.99**.

Study Tips

This product will provide you questions and answers along with detailed explanations carefully compiled and written by our experts. Try to understand the concepts behind the questions instead of cramming the questions. Go through the entire document at least twice so that you make sure that you are not missing anything.

Latest Version

We are constantly reviewing our products. New material is added and old material is revised. Free updates are available for 180 days after the purchase. You should check the products page on the <http://www.chinatag.com> website for an update 3-4 days before the scheduled exam date.

Please tell us what you think of our products. We appreciate both positive and critical comments as your feedback helps us improve future versions. Feedback on specific questions should be send to feedback@chinatag.com.

Thanks for purchasing our products and look forward to supplying you with all your Certification training needs.

Good studying!

Technical and Support Team
Chinatag LLC.

TABLE OF CONTENTS

Introduction

1. Introduction to Active Directory and Network Infrastructure

- 1.1 Active Directory Overview
 - 1.1.1 Windows 2000 Active Directory
 - 1.1.2 Active Directory Objects
 - 1.1.3 Active Directory Schema
 - 1.1.4 Active Directory Components
 - 1.1.5 Logical Structures
 - 1.1.5.1 Domains
 - 1.1.5.2 Organizational Units
 - 1.1.5.3 Trees
 - 1.1.5.4 Forests
 - 1.1.6 Physical Structure
 - 1.1.6.1 Sites
 - 1.1.6.2 Domain Controllers
 - 1.1.7 Catalog Services
 - 1.1.7.1 The Global Catalog
 - 1.1.7.2 Global Catalog Roles
- 1.2 Understanding Active Directory Concepts
 - 1.2.1 Replication
 - 1.2.1.1 Information That Is Replicated
 - 1.2.1.2 How Replication Works
 - 1.2.2 Trust Relationships
 - 1.2.3 Group Policy
 - 1.2.4 DNS Namespace
 - 1.2.5 Domain Namespace
 - 1.2.5.1 Root Domain
 - 1.2.5.2 Top-Level Domains
 - 1.2.5.3 Second-Level Domains
 - 1.2.5.4 Host Names
 - 1.2.5.5 Zones
 - 1.2.6 Naming Conventions
 - 1.2.6.1 Distinguished Name
 - 1.2.6.2 Relative Distinguished Name
 - 1.2.6.3 Globally Unique Identifier

2. Introduction to Designing a Directory Services Infrastructure

- 2.1 The Active Directory Infrastructure Design
- 2.2 Design Tools
 - 2.2.1 Assembling a Design Team

2.2.2 Analyzing Business and Technical Environments

2.2.3 Testing Environment

2.3 The Design Process

2.3.1 Stage One—Creating a Forest Plan

2.3.2 Stage Two—Creating a Domain Plan

2.3.3 Stage Three—Creating an Organizational Unit Plan

2.3.4 Stage Four—Creating a Site Topology Plan

2.4 Design Guiding Principles

2.5 Analyzing the Current Business Environment

2.5.1 Analyzing Products and Customers

2.5.2 Analyzing Current Business Structure

2.5.3 Analyzing Current Business Processes

2.5.3.1 Decision-Making Processes

2.5.4 Analyzing Business Strategy Influences

2.5.5 Analyzing the Information Technology Management Organization

2.6 Analyzing the Current Technical Environment

2.6.1 Analyzing the Current Technical Environment

2.6.2 Analyzing the Current Network Architecture

2.6.3 Analyzing the Current Technical Standards

2.6.4 Analyzing the Current DNS Environment

2.6.5 Analyzing the Current Windows NT Domain Architecture

3. Creating a Forest Plan

3.1 Designing a Forest Model

3.1.1 Understanding Forests

3.1.2 Assessing Forest Needs

3.1.3 Determining the Number of Forests

3.1.3.1 Reasons to Use Multiple Forests

3.1.3.2 Implications of Using Multiple Forests

3.2 Designing a Schema Modification Plan

3.2.1 Understanding the Schema

3.2.2 Viewing the Base Schema

3.2.2.1 Viewing Schema Class Objects

3.2.2.2 Viewing Schema Attribute Objects

3.2.3 Schema Admins Group

3.2.4 Creating a Schema Modification Policy

3.2.5 Assessing Schema Needs

3.2.6 Determining Whether to Modify the Schema

3.2.6.1 Reasons to Modify the Schema

3.2.6.2 Automatic Schema Modification

3.2.6.3 Implications of Modifying the Schema

4. Creating a Domain Plan

4.1 Defining Domains

- 4.1.1 Understanding Domains
 - 4.1.1.1 Goals for Defining Domains
 - 4.1.1.2 Defining Domains Based on Geographical Structure
 - 4.1.1.3 Minimizing the Number of Domains
- 4.1.2 Assessing Domain Needs
- 4.1.3 Determining the Number of Domains
- 4.1.4 Reasons to Define Multiple Domains
 - 4.1.4.1 Meeting Security Requirements
 - 4.1.4.2 Meeting Administrative Requirements
 - 4.1.4.3 Optimizing Replication Traffic
 - 4.1.4.4 Retaining Windows NT Domains
- 4.1.5 Implications of Defining Multiple Domains

4.2 Defining a Forest Root Domain

- 4.2.1 Understanding the Forest Root Domain
- 4.2.2 Assessing Forest Root Domain Needs
- 4.2.3 Choosing a Forest Root Domain
 - 4.2.3.1 Reasons for Designating an Existing Domain
 - 4.2.3.2 Reasons for Designating a Dedicated Domain
 - 4.2.3.3 Advantages of Using a Dedicated Domain

4.3 Defining a Domain Hierarchy

- 4.3.1 Understanding Domain Hierarchies
- 4.3.2 Cross-Link Trusts
- 4.3.3 Assessing Domain Hierarchy Needs
- 4.3.4 Determining a Domain Hierarchy
 - 4.3.4.1 Determining the Number of Domain Trees
 - 4.3.4.2 Designating Tree Root Domains
 - 4.3.4.3 Arranging the Subdomain Hierarchy
 - 4.3.4.4 Planning Cross-Link Trusts

4.4 Naming Domains

- 4.4.1 Understanding Domain Names
- 4.4.2 Assessing Domain Naming Needs
- 4.4.3 Choosing Domain Names

4.5 Planning DNS Server Deployment

- 4.5.1 Understanding DNS Servers
- 4.5.2 Zone Replication
 - 4.5.2.1 Standard Zone Replication
 - 4.5.2.2 Active Directory Zone Replication
- 4.5.3 DNS Server Requirements
- 4.5.4 Assessing the DNS Server Environment
- 4.5.5 Determining Placement of DNS Servers
 - 4.5.5.1 Planning Additional Zones

- 4.5.5.2 Determining Existing DNS Services
- 4.5.5.3 Determining the Zone Replication Method

5. Creating an Organizational Unit Plan

- 5.1 Defining OU Structures
 - 5.1.1 Understanding OUs
 - 5.1.2 Defining OUs to Delegate Administration
 - 5.1.2.1 OU Hierarchy Models for Delegation of Administration
 - 5.1.3 Defining OUs to Hide Objects
 - 5.1.4 Defining OUs to Administer Group Policy
 - 5.1.5 Group Policy Inheritance
 - 5.1.6 Guidelines for Defining OU Structures
 - 5.1.7 Defining OU Structures to Delegate Administration
 - 5.1.7.1 Assessing IT Administration Requirements
 - 5.1.7.2 Defining OUs to Delegate Full Control or Control of Object Classes
 - 5.1.8 Defining OU Structures to Hide Objects
 - 5.1.8.1 Assessing the Need to Hide Objects
 - 5.1.8.2 Defining OUs to Hide Objects
 - 5.1.9 Defining OU Structures to Administer Group Policy
 - 5.1.9.1 Assessing the Need to Define OU Structures
 - 5.1.9.2 Defining the OU
- 5.2 Planning User Accounts and Groups
 - 5.2.1 Understanding Users and Groups
 - 5.2.2 User Accounts
 - 5.2.3 Groups
 - 5.2.3.1 Group Types
 - 5.2.3.2 Group Scopes
 - 5.2.3.3 Group Nesting
 - 5.2.3.4 Rules for Group Membership
 - 5.2.4 Naming and Placing User Accounts
 - 5.2.4.1 Assessing User Account Naming Conventions and the OU Structure
 - 5.2.4.2 Determining User Account Naming Conventions
 - 5.2.4.3 Placing User Accounts in the Appropriate OUs
 - 5.2.5 Naming and Defining Groups
 - 5.2.5.1 Assessing Group Naming Conventions and the OU Structure
 - 5.2.5.2 Determining the Group Naming Convention
 - 5.2.5.3 Defining the Appropriate Global and Domain Local Groups
 - 5.2.5.4 Defining the Appropriate Universal Groups

6. Creating a Site Topology Plan

- 6.1 Defining Sites
 - 6.1.1 Understanding Sites
 - 6.1.2 Assessing the Need for Sites
 - 6.1.3 Defining Sites for the Organization

6.2 Placing Domain Controllers in Sites

- 6.2.1 Understanding Domain Controller Placement
- 6.2.2 Naming Domain Controllers and Computers
- 6.2.3 Assessing the Need for Domain Controllers
- 6.2.4 Determining the Location of Domain Controllers
 - 6.2.4.1 Using Active Directory Sizer

6.3 Defining a Replication Strategy

- 6.3.1 Understanding Replication
- 6.3.2 Configuring Site Links
 - 6.3.2.1 Site Link Transitivity
 - 6.3.2.2 Site Link Bridges
- 6.3.3 Bridgehead Servers
- 6.3.4 Assessing Physical Connectivity
- 6.3.5 Planning a Site Link Configuration
- 6.3.6 Planning Site Link Transitivity
- 6.3.7 Planning Preferred Bridgehead Servers

6.4 Placing Global Catalog Servers and Operations Masters

- 6.4.1 Understanding Global Catalog Servers
- 6.4.2 Understanding Operations Masters
- 6.4.3 Determining the Location of Global Catalog Servers
- 6.4.4 Determining the Location of Operations Masters
- 6.4.5 Planning the Operations Master Role Assignments by Domain
- 6.4.6 Planning the Operations Master Roles for the Forest
- 6.4.7 Planning for Growth

7. Creating an Active Directory Implementation Plan

7.1 Planning a Migration from Windows NT 4 Directory Services

- 7.1.1 Understanding Migration
 - 7.1.1.1 Migration Methods
 - 7.1.1.2 Migrating Resource Domains
 - 7.1.1.3 Migration and the Production Environment
 - 7.1.1.4 Migration and Windows 2000 Domain Modes
- 7.1.2 The Active Directory Migration Tool (ADTM)
- 7.1.3 Assessing Migration Goals
- 7.1.4 Determining the Migration Method

7.2 Planning the Migration

- 7.2.1 Planning a Domain Upgrade
- 7.2.2 Planning a Domain Restructure
- 7.2.3 Planning the Consolidation of Resource Domains into OUs

7.3 Migration Strategies for the Single Domain Model

7.4 Multimaster Domain Model Migration Strategy

7.5 Multiple Trust Domain Model Migration Strategy

7.6 Planning Directory Service Synchronization with Active Directory

7.6.1 Understanding Directory Service Synchronization

7.6.2 Synchronizing with Microsoft Exchange Server 5.5

7.6.3 Synchronizing with Novell NetWare Bindery or NDS

7.6.4 Synchronizing with Other LDAP-Compliant Directory Services

7.6.4.1 Analyzing the Current Domain Structure and Exchange Server Site Topology

7.6.4.2 Mapping Exchange Server Sites and Containers to Active Directory Domains and OUs

7.6.4.3 Mapping Exchange Server Attributes to Active Directory Attributes

7.6.4.4 Determining the Location of Active Directory Connectors

7.6.4.5 Defining Connection Agreements

7.6.4.6 Configuring Connection Agreements

7.7 Planning Novell NetWare Bindery or NDS synchronization with Active Directory

7.7.1 Analyzing the Current Novell Network

7.7.2 Choosing One- or Two-Way Synchronization

7.7.2.1 Reasons to choose one-way synchronization

7.7.2.2 Reasons to choose two-way synchronization

7.7.3 Identifying Objects to Synchronize and Planning Synchronization Sessions

7.7.4 Determining Administrative Responsibilities

7.7.5 Planning Pilot Testing and User Education

Designing a Microsoft Windows 2000 Directory Services Infrastructure

Exam Code: 070-219

Certifications:

Microsoft Certified (MCP)

Microsoft Certified Systems Engineer (MCSE)

Design

Prerequisites:

Updating Support Skills from Microsoft Windows NT 4.0 to Microsoft Windows 2000, or Implementing and Administering Microsoft Windows 2000 Directory Services.

About This Study Guide

This Study Guide provides all the information required to pass the Microsoft 070-219 exam – Designing a Microsoft Windows 2000 Directory Services Infrastructure. It however, does not represent a complete reference work but is organized around the specific skills that are tested in the exam. Thus, the information contained in this Study Guide is specific to the 070-219 exam and not only to Designing a Microsoft Windows 2000 Directory Services Infrastructure. It includes the information required to answer questions related to Microsoft Exchange Server, Microsoft SQL Server, and Novell Netscape interoperability that may be asked during the exam. Topics covered in this Study Guide includes Analyzing Business Requirements; Analyzing the existing and planned business models; Analyzing the company model and the geographical scope; Analyzing the existing and planned organizational structures; Analyzing Technical Requirements; Analyzing company size and the distribution of users and resources; Analyzing performance requirements; Analyzing data and system access patterns; Analyzing network roles and responsibilities; Analyzing security considerations; Analyzing the impact of Active Directory on the existing and planned technical environment; Analyzing existing and planned network and systems management; Analyzing the business requirements for client computer desktop management; Designing a Directory Service Architecture; Defining the scope of the Active Directory design; Designing an Active Directory forest and domain structure; Designing an Active Directory naming strategy; Design and plan the structure of organizational units. Considerations include administrative control, existing domain structures, administrative policy, and geographic and company structure; Designing a schema modification policy; Designing an Active Directory implementation plan; Designing the placement of operations masters, global catalog servers, domain controllers, DNS, WINS, and DHCP servers; Designing an Active Directory site topology.

Intended Audience

This Study Guide is targeted specifically at people who wish to take the Microsoft MCSE Design exam 070-219 – Designing a Microsoft Windows 2000 Directory Services Infrastructure. This information in this Study Guide is specific to the exam. It is not a complete reference work. Although our Study Guides are aimed at new comers to the world of IT, the concepts dealt with in this Study Guide are complex and require an understanding of material provided for the MCSA / MCSE exams: 070-210 – Installing, Configuring, and

Administering Microsoft Windows 2000; 070-215 – Installing, Configuring, and Administering Microsoft Windows 2000 and 070-217 - Implementing and Administering a Microsoft Windows 2000 Directory Services Infrastructure.

Note: There is a fair amount of overlap between 070-219 and 070-217. Don't skim over the information that seems familiar. Read over it again to refresh your memory.

How To Use This Study Guide

To benefit from this Study Guide we recommend that you:

- For best results, use this Study Guide in conjunction with the TestKing Study Guides for exam 070-217 and 070-215. These will provide you with valuable background information.
- Study each chapter carefully until you fully understand the information. This will require regular and disciplined work.
- Perform all labs that are included in this Study Guide to gain practical experience, referring back to the text so that you understand the information better. Remember, it is easier to understand how tasks are performed by practicing those tasks rather than trying to memorize each step.
- Be sure that you have studied and understand the entire Study Guide before you take the exam.

Note: Remember to pay special attention to these note boxes as they contain important additional information that is specific to the exam.

Good luck!

1. Introduction to Active Directory and Network Infrastructure

1.1 Active Directory Overview

1.1.1 Windows 2000 Active Directory

A directory stores information related to the network resources to make locating and managing these resources simpler. A **directory service** identifies all resources on a network and makes them accessible to users and applications. Active Directory is the directory service included in Windows 2000 Server. It includes the directory, which stores information about network resources, **and** all the services that make the information available. The information about user data, printers, servers, databases, groups, computers, and security policies stored in the directory, is organized into objects.

1.1.2 Active Directory Objects

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

1.1.3 Active Directory Schema

The Active Directory schema defines objects that may be stored in Active Directory. Because the schema definitions are stored as objects themselves they can be administered in the same manner as the rest of the objects in Active Directory.

The schema contains two types of definition objects: schema class objects and schema attribute objects. Class objects and attribute objects are defined in separate lists within the schema. They are also known as **schema objects** or **metadata**.

A **schema class object** describes the possible Active Directory objects that may be created. It functions as a template for creating new Active Directory objects. Each schema class is a collection of schema attribute objects. When you create a schema class, the schema attributes store the information describing the object.

Schema attribute objects define the schema class objects with which they are associated. Each schema attribute is defined only once and can be used in multiple schema classes.

A set of basic schema classes and attributes is shipped with Windows 2000 Server. Experienced developers and network administrators may dynamically extend the schema by defining new classes and attributes for existing classes. Schemas cannot be deleted, but only deactivated, and are automatically replicated, planning and preparation is needed before the extension.

1.1.4 Active Directory Components

The logical structures of your organization are represented by the following Active Directory components: domains, organizational units, trees, and forests. The physical structure of your organization is represented by the following Active Directory components: sites (physical subnets) and domain controllers. Active Directory completely separates the logical structure from the physical structure.

Active Directory also automatically builds the global catalog on the first domain controller in a forest. This serves as the central repository of selected information about objects.

1.1.5 Logical Structures

In Active Directory, resources are organized in a logical structure reflecting that of your organization. Grouping resources logically allows you to find a resource by its name rather than by its physical location. Because you group resources logically, Active Directory makes the network's physical structure transparent to users.

1.1.5.1 Domains

The core unit of logical structure in Active Directory is the **domain**, which can store millions of objects. Objects in a domain are those vital to the network. These are items the networked community needs, i.e.: printers, documents, e-mail addresses, databases, users, distributed components, and other resources. All network objects are in 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. Theoretically, a domain directory can contain up to 10 million objects, but 1 million objects per domain is a more practical number. 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. None of the security policies and settings can cross from one domain to another. The domain administrator has absolute rights to set policies only within that domain.

1.1.5.2 Organizational Units

An **organizational unit (OU)** is a container used to organize objects into a logical administrative group. This organization typically reflects your organization's functional or business structure. An OU can contain objects such as user accounts, groups, computers, printers, applications, file shares, and other OUs from the same domain. The OU hierarchy in a domain is independent of the OU hierarchy structure of other domains. By adding OUs to other OUs, or **nesting**, administrative control in a hierarchical fashion is provided. OUs provide a means for handling administrative tasks.

1.1.5.3 Trees

A **tree** is a grouping or hierarchical arrangement of one or more Windows 2000 domains that you create by adding one or more child domains to an existing parent domain. Domains in a tree share a contiguous namespace and a hierarchical naming structure. Following Domain Name System (DNS) standards, the domain name of a child domain is the relative name of that child domain appended with the name of the parent domain. All domains within a single tree share a common schema, a formal definition of all object types that you can store in an Active Directory deployment.

All domains within a tree share a common global catalog, which is the central repository of information about objects in a tree. By creating a hierarchy of domains in a tree, you can allow for administration within an OU or within a single domain of a tree. The tree structure easily accommodates organizational changes.

1.1.5.4 Forests

A **forest** is a grouping or hierarchical arrangement of one or more separate, completely independent domain trees. All trees in a forest share a common schema. They 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. Two-way transitive trusts exist between domains and domain trees.

1.1.6 Physical Structure

The physical components of Active Directory are sites and domain controllers.

1.1.6.1 Sites

A site is a combination of one or more IP subnets connected by a highly reliable and fast link to localize network traffic as much as possible. Typically, a site has the same boundaries as a local area network (LAN). An available bandwidth, i.e., the average amount of bandwidth that is available for use after normal network traffic is handled, of 128 Kbps is sufficient for a site.

With Active Directory, sites are not part of the namespace as sites only contain computer and connection objects used to configure replication between sites.

1.1.6.2 Domain Controllers

A domain controller is a computer running Windows Server 2003 that stores a replica of the domain directory (local domain database). The functions of domain controllers are as follows:

- Each domain controller stores and manages a copy of all Active Directory information for that domain.
- Domain controllers automatically replicate directory information in the domain to each other and immediately replicate important updates. **Operations master roles** are special roles assigned to one or more domain controllers in a domain to perform single-master replication.
- Domain controllers detect collisions
- Having more than one domain controller provides fault tolerance
- Domain controllers manage all aspects of users' domain interaction.

You must place domain controllers in sites which reflect your organization's physical structure and optimize replication and authentication.

1.1.7 Catalog Services

1.1.7.1 The Global Catalog

Finding objects outside of the domain and across the enterprise requires a mechanism that allows the domains to act as one entity. A **catalog service** contains selected information about every object in all domains in the directory, which is used to perform searches across an enterprise. The catalog service provided by Active Directory services is called the global catalog.

The **global catalog** is the central repository of information about objects in a tree or forest. By default, it is created automatically on the initial domain controller in the first domain in the forest, the **global catalog server**. Using Active Directory services multimaster replication, the global catalog information is replicated between global catalog servers in other domains. It stores a full replica of all object attributes in the directory for its host domain and a partial replica of all object attributes contained in the directory for every

domain in the forest. The partial replica stores attributes most frequently used in search operations. Attributes are marked or unmarked for replication in the global catalog when they are defined in the Active Directory schema. Object attributes replicated to the global catalog inherit the same permissions as in source domains, ensuring that data in the global catalog is secure.

1.1.7.2 Global Catalog Roles

The global catalog (GC) allows a user to log on to a network by providing **universal group membership information** to a domain controller when a logon process is initiated and it enables finding directory information regardless of which domain in the forest actually contains the data.

When a user logs on to the network, the GC provides universal account group membership information for the account to the domain controller processing the information. The GC server role is held if there is only one domain controller. If there are many domain controllers, one is assigned the GC. If a GC is unavailable, the user can only log on to the local computer unless the site has been configured to **cache universal group membership lookups** when processing user logon attempts.

The GC responds to user and programmatic queries anywhere in the tree or forest with maximum speed and minimum network traffic. A single GC contains information about all objects in all domains in the forest, thus, a query about an object that is not contained in the local domain can be resolved by a GC server in the domain in which the query is initiated.

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

1.2 Understanding Active Directory Concepts

1.2.1 Replication

Replication ensures that changes to a domain controller are reflected in all domain controllers within a domain. Directory information is replicated to domain controllers both within and among sites.

1.2.1.1 Information That Is Replicated

The information stored in the directory is partitioned into three categories. Each of these information categories is referred to as a **directory partition**. These are the units of replication. Information contained in each directory includes: Schema information, Configuration information and Domain data.

Schema and configuration information and domain data is replicated to all domain controllers. All objects in all domains, and selected attributes for all objects in a forest, are replicated to the global catalog.

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
- Selected attributes for all directory objects
- All directory objects and all their properties for the domain in which the global catalog is located

1.2.1.2 How Replication Works

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

- In **Intrasite Replication**, the Windows Server 2003 **knowledge consistency checker (KCC)** generates a topology for replication among domain controllers in the same domain. The topology defines the path for directory updates to pass between domain controllers. The KCC determines which servers replicate with each other and designates certain domain controllers as replication partners. Domain controllers can have more than one replication partner. The KCC then builds connection objects that represent replication connections between the replication partners.

The KCC analyzes the replication topology within a site every 15 minutes. If you add or remove a domain controller from the network or site the KCC reconfigures the topology.

When more than seven domain controllers are added to a site, the KCC creates more connection objects so that if a change occurs at a domain controller, replication partners can ensure that no domain controller is more than three replication hops from another domain controller.

- In **Intersite Replication** between sites, a single KCC generates all connections between the sites. Active Directory uses the network connection information to generate connection objects that provide efficient replication.

1.2.2 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: Implicit two-way transitive trust and an Explicit one-way nontransitive trust.

- **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 relationships among domains in a tree are established and maintained implicitly (automatically). Transitive trust is a feature of the Kerberos authentication protocol, which provides the distributed authentication and authorization in Windows 2000.

Transitive trust between domains eliminates the management of interdomain trust accounts.

- **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. In most cases, you must explicitly (manually) create nontransitive trusts.

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
- A Windows 2000 domain and an MIT Kerberos V5 realm

1.2.3 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, you create group policy objects (GPOs). These are collections of group policy settings. Each Windows 2000 computer has one local GPO and may be subject to any number of nonlocal (Active Directory–based) GPOs. Local GPOs are overridden by nonlocal GPOs. Nonlocal GPOs are linked to Active Directory objects and can be applied to either users or computers. Following the inheritance properties of Active Directory, nonlocal GPOs are applied hierarchically from the least restrictive group (site) to the most restrictive group (OU) and are cumulative.

Because nonlocal GPOs are applied hierarchically, the user or computer's configuration is a result of the GPOs applied to its site, domain, and OU. Group policy settings are applied in the following order:

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

The default order for the application of group policy settings is subject to these exceptions:

- A computer that is a member of a workgroup processes only the local GPO.
- No Override.
- Block Policy Inheritance
- Loopback setting.
- Replace.
- Merge

1.2.4 DNS Namespace

Active Directory, like all directory services, is primarily a namespace. A **namespace** is any bounded area in which a name can be resolved. **Name resolution** is the process of translating a name into some object or information that the name represents. 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 names are user friendly
- DNS names remain more constant than IP addresses
- DNS allows users to connect to local servers using the same naming convention as the Internet.

Because Active Directory uses DNS as its domain naming and location service, Windows 2000 domain names are also DNS names. Windows 2000 Server uses dynamic DNS, which enables clients with dynamically assigned addresses to register directly with a server running the DNS service and update the DNS table dynamically. Dynamic DNS eliminates the need for other Internet naming services, such as Windows Internet Naming Service (WINS), in a homogeneous environment.

1.2.5 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.

The DNS database is indexed by name; therefore, each domain must have a name. As you add domains to the hierarchy, the name of the parent domain is appended to its child domain (called a **subdomain**). A domain's name identifies its position in the hierarchy. The hierarchical structure of the domain namespace consists of a root domain, top-level domains, second-level domains, and host names.

There are two types of namespaces: contiguous and disjointed namespace.

1.2.5.1 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.

1.2.5.2 Top-Level Domains

Top-level domains are arranged by organization type or geographic location. Top-level domains can contain second-level domains and host names.

1.2.5.3 Second-Level Domains

Organizations, such as Network Solutions, Inc., and others, assign and register **second-level domains** to individuals and organizations for the Internet. A second-level name has two name parts: a top-level name and a unique second-level name.

1.2.5.4 Host Names

Host names refer to specific computers on the Internet or a private network. A host name is the leftmost portion of a **fully qualified domain name (FQDN)**, which describes the exact position of a host within the domain hierarchy.

1.2.5.5 Zones

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

Multiple zones in a domain namespace are used to distribute administrative tasks to different groups. A zone must encompass a contiguous domain namespace. The name-to-IP-address mappings for a zone are stored in the zone database file. Each zone is anchored to a specific domain, referred to as the zone's **root domain**. The zone database file does not necessarily contain information for all subdomains of the zone's root domain, only those subdomains within the zone.

A DNS name server stores the zone database file. DNS name servers use these files to handle the DNS name resolution process. Name servers can store data for one zone or multiple zones. A name server is said to



have authority for the domain namespace that the zone encompasses. When a DNS name server receives a DNS query, it responds in one of three ways: by returning the requested name or IP-resolution information, by returning a pointer to another DNS name server, or by indicating that the information is not available. There are three main types of DNS name servers: primary, secondary, and master.

A **primary name server** gets data from the local zone and is the authoritative server for the zone. A **secondary name server** is a backup DNS server and receives data from another name server. A **master name server** is a primary or secondary name server for a zone that is designated to provide updated DNS information to a secondary server.

1.2.6 Naming Conventions

Active Directory uses a variety of naming conventions: distinguished names, relative distinguished names, globally unique identifiers, and user principal names.

1.2.6.1 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.

1.2.6.2 Relative Distinguished Name

Active Directory supports querying by attributes, so you can locate an object even if the exact DN is unknown or has changed. The **relative distinguished name** (RDN) of an object is the part of the name that is an attribute of the object itself. You can have duplicate RDNs for Active Directory objects, but you cannot have two objects with the same RDN in the same OU.

1.2.6.3 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 you move or rename the object. Applications can store the GUID of an object and use the GUID to retrieve that object regardless of its current DN.

A GUID is unique across all domains, meaning that you can move objects from domain to domain and they will still have a unique identifier.

1.2.6.4 User Principal Name

Each user account has a "friendly" name, the **user principal name** (UPN). The UPN comprises a shorthand name for the user account and the DNS name of the tree where the user account object resides.