# Table of Contents

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction to Active Directory</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Active Directory Architecture</td>
<td>31</td>
</tr>
<tr>
<td>3</td>
<td>Domains, Forests and Trusts</td>
<td>65</td>
</tr>
<tr>
<td>4</td>
<td>Sites and Replication</td>
<td>89</td>
</tr>
<tr>
<td>5</td>
<td>User Groups and Computer Accounts</td>
<td>117</td>
</tr>
<tr>
<td>6</td>
<td>Organizational Units</td>
<td>163</td>
</tr>
<tr>
<td>7</td>
<td>Creating and Managing Group Policies</td>
<td>175</td>
</tr>
<tr>
<td>8</td>
<td>Disaster Recovery</td>
<td>205</td>
</tr>
</tbody>
</table>
TAB

Insert Tab # 1 Here
Intro to Active Directory
The Microsoft Server System contains several operational infrastructure technologies. Among these system servers are Windows Server 2003, ISA server, MOM Server, SQL 2000, MAC Server, and SMS Server. All of these servers utilize the Active Directory service hosted from the Windows 2000 Server.

Windows Server NT 4.0 – The Windows NT server family was the predecessor to Microsoft’s modern server platform and served as a functional contributor to several other consumer level Microsoft products. The original NT servers were used at corporate levels but lacked enterprise scalability and flexible administrative controls. Windows NT 4.0 included several versions to address these issues. Eventually the NT family was phased out by Server 2000 however; there still exists businesses using legacy systems still use Windows NT. Support of the NT family ended on January 1, 2005.

Windows Server 2000 – Is a primary operating system and functional platform for all Microsoft server products that succeeds Microsoft Server NT 4.0. Server 2000 is backwards compatible with most Server NT 4.0 components and was designed to be fully scalable with Enterprise infrastructures.


**Internet Security Application (ISA) Server** - ISA Server is an application-layer firewall, virtual private network, and Web cache server delivering Microsoft class security with simplified management.

**Microsoft Operations Manager (MOM) Server** - MOM Server is an event and performance management tool for Windows Server System

**Microsoft Application Center (MAC) Server** – MAC Server is Microsoft's deployment and management tool for high-availability Web applications. MAC makes managing groups of servers as simple as managing a single computer.

**SMS Server** – SMS provides a solution for software updates and configuration changes from non-Microsoft vendors.

**Software Update Services (SUS) Server** – SUS is designed to deploy multiple updates to client operating systems and/or software at designated intervals.

The “Active” Directory Concept

A domain is a security boundary.

The **Microsoft Server Systems** employs several key services that are depended upon by the other servers within the enterprise operational infrastructure. Active Directory, LDAP, DNS, and SQL are commonly used services that are required by other services within the system. To use the Microsoft Windows 2003 Server operating system with maximum effectiveness you must first understand what the Active Directory service is.

**Store.** Active Directory is the Windows 2000 Server directory service. It hierarchically stores information about network objects and makes this information available to administrators, users, and applications.

**Structure.** In AD, the network and its objects are organized by object containers: domains, trees, forests, trust relationships, organizational units, and sites. Each of these objects containers are issued varied levels of administrative control.

**Inter-communicate.** Active Directory is based on standard directory access protocols. It can operate in conjunction with other directory services that use these same open protocols.
Microsoft’s Active Directory can be condensed into three very high level concepts:

- A namespace declaration that is integrated with the Internet's Domain Name System (DNS)
- Software that makes a server into a domain controller
- A mid-point service that enables other directory dependant applications to function properly
- A dynamic entry is an object in the directory, which has an associated time-to-live (TTL) value. The TTL for an entry is set when the entry is created. The time-to-live is automatically decremented, and when it expires, the dynamic entry disappears. The client can cause a dynamic entry to remain longer than its current remaining life by refreshing (modifying) its TTL value. Clients that store dynamic data in the directory must periodically refresh that data to prevent it from disappearing.
Open Standards

- **LDAP**
  - Low-Level API to Active Directory
- **X.500**
  - Active Directory Structure
  - Not fully standard-compliant
- **DNS**
  - Resource Location
  - Extensions (e.g. “Dynamic DNS”)
- **Kerberos**
  - Authentication

- A directory is a set of objects with attributes organized in a logical and hierarchical manner. A simple example is the telephone directory, which consists of a list of names (of either persons or organizations) organized alphabetically, with each name having an address and phone number associated with it.

- **X.500** is a series of computer networking standards covering electronic directory services. The X.500 series was developed by ITU-T, formerly known as CCITT. The directory services were developed in order to support the requirements of X.400 electronic mail exchange and name lookup. ISO was a partner in developing the standards, incorporating them into the Open Systems Interconnection suite of protocols. ISO/IEC 9594 is the corresponding ISO identification.

- The Domain Name System (DNS) is a hierarchical naming system for computers, services, or any resource connected to the Internet or a private network. It associates various information with domain names assigned to each of the participants. Most importantly, it translates domain names meaningful to humans into the numerical (binary) identifiers associated with networking equipment for the purpose of locating and addressing these devices worldwide. An often-used analogy to explain the Domain Name System is that it serves as the "phone book" for the Internet by translating
human-friendly computer hostnames into IP addresses. For example, www.example.com translates to 208.77.188.166

- Kerberos uses as its basis the symmetric Needham-Schroeder protocol. It makes use of a trusted third party, termed a key distribution center (KDC), which consists of two logically separate parts: an Authentication Server (AS) and a Ticket Granting Server (TGS). Kerberos works on the basis of "tickets" which serve to prove the identity of users
Kerberos - AD Security

• Legacy Internet-Standard
• Commonly used with Unix
• Secure Encrypted Authentication
• Authentication Standard for Server 2000 and 2003
• Microsoft Kerberos is not fully compatible to other Kerberos Implementations

• The current version of Kerberos is v5, which was developed in 1993. This is the version on which Microsoft’s implementation in Windows 2000/XP/Server 2003 is based. Windows 2000 and Server 2003 native mode domains use Kerberos by default. Domains that must authenticate NT systems along with the newer operating systems must use NT LAN Manager (NTLM) authentication
What is the Schema?

• Definition of…
  – Object-Types (Classes)
  – Attributes
  – Data-Types (Syntaxes)

• Extensible objects

• Similar to a Database’s Schema

• ONE consistent Schema inside a single Forest

• The Microsoft Active Directory schema contains formal definitions of every object class that can be created in an Active Directory forest. The schema also contains formal definitions of every attribute that can exist in an Active Directory object. This section provides the reference for each schema object and provides a brief explanation of the attributes, classes, and other objects that comprise the Active Directory schema

• Attribute: Data items used to describe the objects that are represented by the classes that are defined in the schema. Attributes are defined in the schema separately from the classes; this allows a single attribute definition to be applied to many classes. For example, Description is an attribute that can be applied to any class in the schema. The Description attribute is defined once in the schema, assuring consistency, rather than having a different definition for Description of a user and Description of a printer

• A formal description of a discrete, identifiable type of object stored in the directory service. For example, User, Print-Queue, and Group are all classes. Furthermore, there are 3 distinct categories of classes: Structural Classes, Abstract Classes, and Auxiliary Classes

• A unit of data storage in the directory service. Directory service objects is not to be confused with COM objects or other object-oriented system objects,
which have an executable component and run-time behavior. Directory service objects consist only of data. A directory service object is defined by a Class-Schema object and a group of Attribute-Schema objects referenced by the Class-Schema object.
Active Directory uses classes to create **objects** that represent unique constructs. To create an object, Active Directory references a set list of predefined classes. A **class** is an attributes list that defines the qualities of an object type. Some object types are user accounts, user groups, computer accounts, computer groups, printer accounts, printer groups, organizational units, group policies, forest, trees, and domains. A class is said to be extensible when its original attributes list is modified with additional parameters. All objects of the same type share or “inherit” from the same class.

A class can be thought of as a standard requirements list that each object in a specific type must adhere. Each object created from a class is inherently different from other objects created from any other class because no two classes have the exact same attributes list. This solves the problem that Active Directory might have in identifying the difference between two object types such as a user account and a user group, but it does not uniquely identify one user account versus all other user accounts that may exist.

To facilitate this, all objects must provide a set of uniquely identifying parameters at the time an object is instantiated from a class. The parameters provided are pieces of data assigned to each required attribute within the object. When all of the required attributes have been provided with data that as a whole uniquely identifies the object as a matchless entry the object may be created. Not all
attributes associated with an object are required attributes. Only required attributes are used to determine if an entry is truly unique.

Account Types:
   User
   Computer
   Printer
   Group Policy (Not an actual account object, GPOs are object dependant upon Organizational Units.)
Active Directory Objects

User
A user object is an object that is a security principal in the directory. A user can log on to the network with these credentials and access permissions can be granted to users.

Contact
A contact object is an account that does not have any security permissions. You cannot log on to the network as a contact. Contacts are typically used to represent external users for the purpose of e-mail.

Computer
An object that represents a computer on the network, for Windows NT-based workstations and servers this is the machine account.

Organizational Unit
Organizational units are used as containers to organize directory objects such as users, groups, and computers in much the same way that folders are used to organize files on your hard disk.

Group
Groups can have users, computers, and other groups. Groups simplify the management of large numbers of objects.
**Shared Folder**
A shared folder is a network share that has been published in the directory.

**Shared printer**
A shared printer is a network printer that has been published in the directory.
Group Types

- Domain Local Groups
- Global Groups
- Universal Groups
- Special Identity Groups

**Domain Local Groups**
Domain Local Groups are groups whose group policy restrictions and membership availability are limited to action within their own domain.

**Global Groups**
Global Groups are groups whose group policy restrictions and membership availability are available to all trusted domains throughout the forest.

**Universal Groups**
Universal Groups are groups whose group policy restrictions and membership availability are available to all trusted and untrusted domains throughout the forest. This is an enterprise level (or forest level) credential.

**Special Identity Groups**
Special Identity Groups is a specific group that literally includes every account available to Active Directory. All accounts are a part of the “EVERYONE” group by default.
Built-in Groups:

- **Domain Local Groups**
  - Account Operators
  - Administrators
  - Backup Operators
  - Guests
  - Print Operators
  - Replicator
  - Server Operators
  - Users

- **Special identity groups:**
  - Everyone

- **Global Groups**
  - Cert Publishers
  - Domain Admins
  - Domain Computers
  - Domain Controllers
  - Domain Guests
  - Domain users
  - Enterprise Admins
  - Group Policy Admins
  - Schema Admins

Built-in groups are groups that are created automatically with each new domain controller setup.
Objects may be copied, modified, deleted, or moved from one logical location to another.

Active Directory dynamically updates its databases to in order to maintain an accurate record of where all objects within a given forest or domain reside.

Account objects are typically nested inside of group objects. Group objects in turn are nested inside of organizational units (OUs). Organizational Units carry specific functional instructions imposed upon them by the domain in which they reside.
What are Organizational Units?

- Implements a Structure inside a Domain
- Can be nested as needed
- Can not be assigned any rights
- Typically used for Administrative purposes
- Uses GPOs to assign System Policies

1. A particularly useful type of directory object contained within domains is the organizational unit. Organizational units are Active Directory containers into which you can place users, groups, computers, and other organizational units. An organizational unit cannot contain objects from other domains.

An organizational unit is the smallest scope or unit to which you can assign Group Policy settings or delegate administrative authority. Using organizational units, you can create containers within a domain that represent the hierarchical, logical structures within your organization. You can then manage the configuration and use of accounts and resources based on your organizational model. For more information about Group Policy settings

2. A nested OU is one that has another OU as its parent. In the case of nested OUs, GPOs associated with parent OUs are processed prior to GPOs associated with child OUs. In this processing order, sites are applied first but have the least precedence. OUs are processed last and have the highest precedence.
Domain Name Server (DNS)

DNS is:
- a hierarchical naming system for computers, services
- serves as a dynamic yet persistent listing key to compare unique service names or resource location names to specific IP/Port address combinations
- allows machine preferred number sequences to be more easily read and written by human language
- may be administered with global or local scopes
- is used in cooperation with additional protocols and resource services

DNS is also referred to as:
- Domain Name Space
- Domain Name Service
- Domain Name System
- Name Resolution Server

Domain Names:
- consists of two or more parts
- each part (domain) is separated by a single dot
- each extending domain specifies a more specific path residing within the location of the previous domain
Domains are administrative entities. The purpose and expected use of domains is to divide the name management required of a central administration and assign it to sub-administrations. There are no geographical, topological, or technological constraints on a domain. The hosts in a domain need not have common hardware or software, nor even common protocols. Most of the requirements and limitations on domains are designed to ensure responsible administration.

The domain system is a tree-structured global name space that has a few top-level domains. The top-level domains are subdivided into second level domains. The second level domains may be subdivided into third level domains, and so on.

The administration of a domain requires controlling the assignment of names within that domain and providing access to the names and name related information (such as addresses) to users both inside and outside the domain.

The right to use a domain name is delegated by domain name registrars which are accredited by the Internet Corporation for Assigned Names and Numbers (ICANN), the organization charged with overseeing the name and number systems of the Internet. In addition to ICANN, each top-level domain (TLD) is maintained and serviced technically by a sponsoring organization, the TLD Registry. The registry is responsible for maintaining the database of names registered within the TLDs they administer. The registry receives registration information from each domain name registrar authorized to assign names in the corresponding TLD and publishes the information using a special service, the 'whois' protocol.

ICANN publishes a complete list of TLD registries and domain name registrars in the world. One can obtain information about the registrant of a domain name by looking in the WHOIS database held by many domain registries.

RFC 1033 Domain administrators operations guide - You may only put data in a domain server that you are authorized to administer.

RFC 1032 Domain administrators’ guide - Name assignments within a domain are controlled by the DA, who should verify that names are unique within his domain and that they conform to standard naming conventions.

RFC 920 Specified original TLDs (Top-Level Domains):
A multi-organization may be a top-level domain if it is large, and is composed of other organizations; particularly if the multi-organization cannot be easily classified into one of the categories and is international in scope. There are very few top-level domains; each of these may have many second level domains. An initial set of top-level names has been identified. Each of these has an administrator and an agent.

.ARPA = The ARPA-Internet *** TEMPORARY ***
Administrator: DARPA
Agent: The Network Information Center
Mailbox: HOSTMASTER@SRI-NIC.ARPA

.GOV = Government
Administrator: DARPA
Agent: The Network Information Center
Mailbox: HOSTMASTER@SRI-NIC.ARPA

.EDU = Education
Administrator: DARPA
Agent: The Network Information Center
Mailbox: HOSTMASTER@SRI-NIC.ARPA

.COM = Commercial
Administrator: DARPA
Agent: The Network Information Center
Mailbox: HOSTMASTER@SRI-NIC.ARPA

.MIL = Military
Administrator: DDN-PMO
Agent: The Network Information Center
Mailbox: HOSTMASTER@SRI-NIC.ARPA
Active Directory Benefits

- Active Directory Benefits
  - Integration with DNS
  - Flexible querying
  - Extensibility
  - Policy-based administration
  - Scalability
  - Information Replication
  - Information security
  - Interoperability

Integration with DNS
Active Directory uses the Domain Name System to locate objects and resources.

Flexible querying
Active Directory objects can be located by using the Search command from the Start menu, using My Network Places, or by using the Active Directory Users and Computers snap-in.

Extensibility
Object class can be modified or newly created and added to the AD schema.

Policy-based administration
Groups, objects, or account objects can be assigned rights and restrictions by being placed under an OU controlled by GPOs (Group Policy Objects).

Scalability
Active Directory includes one or more domains, each with one or more domain controllers, enabling you to scale the directory to meet any network requirements. Multiple domains can be combined into a domain tree and multiple domain trees can be combined into a forest. In the simplest structure, a single-domain network is simultaneously a single tree and a single forest. In the
most complex structure, resources can be shared and managed across sub-networks, WAN links, forest, and sub-domains.

**Information Replication**
Active Directory uses multi-master replication, which lets you update the directory at any domain controller. Deploying multiple domain controllers in one domain provides fault tolerance and load balancing. If one domain controller within a domain slows, stops, or fails, other domain controllers within the same domain can provide necessary directory access, since they contain the same directory data.

**Information security**
Active Directory centralizes authentication processes. Access control provides granular control over objects and object properties.

**Interoperability**
Active Directory is based on standard directory access protocols that are compatible with other non-Microsoft directory services.
Modes & Functional Levels

- Domain Functional Levels
  - Windows 2000 Mixed Mode (default)
  - Windows 2000 Native Mode
  - Windows Server 2003 Interim Mode
  - Windows Server 2003 Mode

- Forest Functional Levels
  - Windows 2000 (Default)
  - Windows Server 2003 Interim
  - Windows Server 2003

There are four domain functional modes in Active Directory:
1. Mixed Mode
2. Native Mode
3. Interim Mode
4. Server 2003 Mode

Functional levels are an extension of the mixed/native mode concept introduced in Windows 2000, which are used to activate Active Directory features new to Server 2003. There are domain functional levels and forest functional levels. Promoting a domain controller to a higher level is an irreversible task.

**DOMAIN FUNCTIONAL LEVELS**

**Windows 2000 Mixed**
Mixed Mode allows Windows NT 4.0 backup domain controllers, Windows Server 2000, and Server 2003 to function simultaneously. Windows 2000 mixed mode is the default function level because it supports all types of domain controllers including non-Microsoft directory controllers.

**Windows 2000 Native**
Native Mode allows Windows 2000 and 2003 servers to function simultaneously, but not Windows NT Servers.
**Windows Server 2003 Interim**
Interim Mode supports domain controllers upgrading from NT machines to Windows Server 2003 and does not support Windows Server 2000. This level occurs when a Microsoft Windows NT 4.0-based *primary domain controller* to Windows Server 2003. Interim mode is important where you have NT 4.0 groups with more than 5000 members because Microsoft Windows 2000 does not allow you to create groups with more than 5000 users.

**Windows Server 2003**
In Server 2003 Mode, all domain controllers on the network must be operating with Windows Server 2003 domain controllers. At this level, other clients and servers may join the domain, but may not be domain controllers.

**FOREST FUNCTIONAL LEVELS**
As when changing the domain functional level, there are requirements for moving to a higher forest functional level. For example, all domains in the forest must be running Windows Server 2003 functional level before you can move up to Windows Server 2003 forest function level.

**Windows 2000**
This mode supports Windows NT, Windows 2000 Server, and Windows Server 2003 domain controllers. It offers essentially the same features as Windows 2000 Server. Consequently, if all domain replication partners are running Windows Server 2003, this mode enables a handful of improvements to global catalog replication.

**Windows Server 2003 Interim**
This mode is available only when migrating from Windows NT Server to Windows Server 2003.

**Windows Server 2003**
This mode offers several additional features not available in the other two modes including replication improvements, forest trusts, domain rename, defunct schema objects, linked value replication, dynamic auxiliary classes, and InetOrgPerson objectClass change. This mode requires that all domain controllers be running Windows Server 2003 in Windows Server 2003 domain mode.
Domain Functional Levels

**Windows 2000 mixed (Default)**

- **Domain Controllers Supported:**
  - Microsoft Windows NT 4.0
  - Windows 2000
  - Windows Server 2003

- **Activated features:**
  - Local and Global groups, Global Catalog support

**Windows Server 2003 Interim**

- **Domain Controllers Supported:**
  - Microsoft Windows NT 4.0
  - Windows Server 2003

- **Activated features:**
  - No domain-wide features
  - All domains in a forest raised to this level
  - Used when upgrade NT domain controllers

**Windows 2000 native**

- **Domain Controllers Supported:**
  - Windows 2000
  - Windows Server 2003

- **Activated features:**
  - Group nesting, Universal groups, SID history
  - Converting groups between security groups and distribution groups

**Windows Server 2003**

- **Domain Controllers Supported:**
  - Windows Server 2003
  - Windows Server 2008

- **Activated features:**
  - Rename domain controller
  - Logon timestamp updated and replicated
  - User password support on ‘InetOrgPerson’
  - Constrained delegation

Requirements for Installing AD

- A computer running Windows Server 2003
- Minimum disk space of 250 MB and a partition formatted with NTFS
- Administrative privileges for creating a domain
- TCP/IP that is installed and configured to use DNS
- An authoritative DNS server that supports SRV resource records
Installation Requirements

- **Installation Methods**
  - Remote Installation
  - Installation across a network
  - Local Installation

- **Installation Checklist:**
  - Confirm that all hardware is HCL* compatible
  - The current user is logged on to the local computer as a member of the Administrators group on the local computer.
  - A previous installation or removal of Active Directory has not taken place without the computer having first been restarted.
  - At least one logical disk drive is formatted with the NTFS v5 file system.
  - Verify Unique Names
  - Verify That TCP/IP Is Installed
  - Verify That DNS Client Is Configured
  - Get and Configure Sites

Active Directory is a featured service of Windows Server 2003.

There are three installation methods for Windows Server 2003 Active Directory:
1. Remote Installation
2. Installation across a network
3. Local Installation

Remote installation of Active Directory is accomplished through a process called Remote Installation Service (RIS), whereby a Windows 2003 installation package is “pushed” out to network systems. RIS simplifies installation to multiple servers but complicates configuration of server key services such as DNS, DHCP, and Active Directory. Additionally the client must be made compatible for RIS by loading a special network boot client disk. Microsoft TechNet (http://technet.microsoft.com) is a good resource for additional information regarding this and other Windows Server 2003 topics.

Windows Server 2003 can be installed across a network by mapping a local drive on the client machine to a network share. Setup and installation occurs almost the same over a network as it would if the installation CD were present locally. The installation process must be started manually. All data files needed to continue installation and setup are automatically copied through the mapped
drive before the installation process requires a reboot. If launching the installation process from a Microsoft operating system predating Windows 2003, the “Copy all Installation Files from the Setup CD” option should be selected.

INSTALLATION CHECKLIST:
- Confirm that all hardware is HCL* compatible
- The current user is logged on to the local computer as a member of the Administrators group on the local computer.
- A previous installation or removal of Active Directory has not taken place without the computer having first been restarted.
- At least one logical disk drive is formatted with the NTFS v5 file system.
- Verify Unique Names
- Verify That TCP/IP Is Installed
- Verify That DNS Client Is Configured
- Get and Configure Sites

*The hardware compatibility list (or HCL) provides a representative list of hardware peripherals that are compatible with the Microsoft Windows operating systems (e.g. Windows 98, Windows NT 4.0, Windows 2000, Windows XP, and Windows 2003). Microsoft has validated the compatibility of the hardware peripherals on an x86-based Windows hardware reference platform.
Pre-installation Checklist

1. Confirm that all hardware is HCL* compatible
2. The current user is logged on to the local computer as a member of the Administrators group on the local computer.
3. The computer is running Windows 2000 Server or higher.
4. A previous installation or removal of Active Directory has not taken place without the computer having first been restarted.
5. The Active Directory Installation Wizard is not currently running.
6. An installation or removal operation of Active Directory is not already in progress.
7. At least one logical disk drive is formatted with the NTFS v5 file system.
8. Verify Unique Names
9. Verify That TCP/IP Is Installed
10. Verify That DNS Client Is Configured
11. Confirm and Validate the DNS Domain Name
12. Get and Validate the NetBIOS Name
13. Enter Administrative Password
14. Get Credentials for the User
15. Get and Verify File Paths
16. Configure Site

HCL* - Hardware compatibility list as provided by Microsoft for supported hardware and peripherals.
TAB

Insert Tab # 2 Here
Active Directory Architecture
Active Directory Architecture

- Hierarchical
- Base object
- Domain
Each and every domain in Active Directory requires at least one dedicated domain controller. Normally this domain controller will also serve as the DNS server and the Operations Master.

**Active Directory DNS services:**
Active Directory requires the use of a DNS server for name resolution. AD calls objects by their hierarchical names and therefore must have IP address translations available at all time in order to reach a desired network resource.

**Active Directory Operations Master services:**
Active Directory requires the use of Operations Master services for situations were multi-master replication is impractical to perform. Single-master operation requires the designation of an Operations Master to serve as a central control point for these types of changes.
Multiple domain controllers may be present in a single domain for redundancy purposes.
In a single domain forest or in a newly declared forest, the first domain controller created becomes the Forest Root, Tree Root, and Domain Controller.
No trust relationship has been specified in this diagram. However, we can assume that all domains trust the forest root.
Trees within a forest do not share a common namespace, but they do share a common forest schema.
In this example all of the domains are in the same forest and therefore share a common Global Catalog. The domain controllers in each site may freely replicate whenever an update is made. Site 1 and Site 2 are in a two-way trust which means that if updates to objects that belong to a domain in the other site will take effect as soon as the replication schedule allows.

When is replication scheduled between sites? It depends on what the administrator setup. A site can be made to replicate across a specified site link at any hour in the day and for any number of times during that ‘open’ replication period. If the window for replication between sites is ‘closed’ then Active Directory will hold all updates for the site until the next window becomes available.

The exchange of object data across sites’ 1 and 2 is explicitly labeled as two-way trust. Updates made on Site 1 about an object that resides in a domain in Site 2 will be replicated to Site 2. Likewise, updates made on Site 2 about an object that resides in a domain in Site 1 will be replicated to Site 1. Of course these updates will wait until the next replication scheduled.

The relationship between Site 3 and Site 2, and Site 3 and Site 1 is more complex than the previous explanation. Site 3 and Site 2 share a one-way trust. Site 3 trust the updates supplied by Site 2, but Site 2 does not trust the updates.
provided by Site 3. As a result of this one-way trust, any updates to objects local to domains in Site 1, 2, or 3 will be made in Site 3 and contained in Site 3.

If a domain controller in Site 3 attempts to add a new user to its own domain and therefore the forest's Global Catalog, all domain controllers in Site 3 will be made aware of the new user. However, none of the domain controllers in the other sites will be made aware that the new user exists. If the domain controllers are not aware that the new user exists, because their GCs have not been updated, then the user will not be able to login to a computer local to any domain outside of the originating site.

Simply put if a new user “John Franks” is added to a domain controller in site 3, but then tries to logon to a computer that is physically located in a subnet range in either site’s 1, 2, or 4, he will be unsuccessful. Conversely, updates made to Site 4 will not be allowed to pass directly to Site 2 or Site 3. Instead, Site 2 and Site 3 will be made aware of changes in Site 4 because of Site 1 acting as a conduit. Site 2 does not trust the updates from Site 4 and Site 3 does not have a site link with any other site except Site 2. Therefore since Site 1 is in full trust with both Site 2 and Site 4, all sites (Site 1, Site 2, Site 3, and obviously Site 4) will be made aware of updates to the Global Catalog by a domain controller that exists in Site 4.

From a security perspective the one-way trust between Site 2 and Site 3 makes sense. What is most logical purpose for the trust relationship / site-link configuration between sites 1, 2, and 4?
From a security perspective the one-way trust between Site 2 and Site 3 makes sense. What is most logical purpose for the trust relationship / site-link configuration between sites 1, 2, and 4?

**ANSWER** – Site 4 most likely has this strange trust relationship / site-link configuration with Site 1 and Site 2 because of physical limitations within the network infrastructure.
Active Directory consists of two structures: Logical and Physical

The **logical structure** defines the resource limitations of each object in Active Directory. It operates primarily without regard to network configuration. The logical structure is what prevents one object from gaining access to an area or set of controls that the object has no permissions to access.

The **physical structure** assures optimum network utilization by limiting the operational functionality of Active Directory in accordance to network limitations. Without the physical structure, Active Directory would treat all resource servers as if they were of equal network cost.
Active Directory Architecture (2)

- Provides consistent information to resource members
  - Names
  - Descriptions
  - Locations
  - Access
  - Management
  - Authentication

- Uses an LDAP (database) protocol

- Identification Keys
  - Object GUIDs
  - User Principal Name (UPNs)
  - SAM Account Name
  - Relative ID (RID)

Active Directory maintains a regularly updated list of network resource locations and AD objects. This provides a central point of query that other services may use to facilitate networked resource sharing. To accomplish the centralized versioning, Active Directory uses the LDAP protocol to query a database of active objects. Each server, client, and object is assigned a uniquely identifying key that LDAP may use to track their location.

**Object GUIDs**
Active Directory objects are the entities that make up a network. An object is a distinct, named set of attributes that represents something concrete, such as a user, a printer, or an application. When you create an Active Directory object, Active Directory assigns it a globally unique identifier (GUID) which is a character string that is unique throughout the forest.

**User Principle Name (UPN)**
The User Principal Name (UPN) provides a naming style for users to log on to Active Directory. The style of the UPN is based on Internet standard RFC 822, which is sometimes referred to as an “email address”. The default UPN suffix is the forest DNS name, which is the DNS name of the first domain in the first tree of the forest. A UPN suffix can be changed by an administrator. The UPN prefix is a custom character string, often the same as a user’s email address that is unique and not shared by another UPN prefix within the forest. Logging into to a
resource controlled by Active Directory using a UPN credential is considered a forest level authentication and grants the user access to all participating domains throughout the forest. There are obvious security implications for granting users this level of access in addition to their normal domain level access. For this reason, UPN’s are not required for user accounts. UPN's are useful, however, when a user is expected to travel between domains or across forests.

**Security Account Manager (SAM)**
A Security Account Manager (SAM) account name is required for compatibility with Windows NT 3.x and Windows NT 4.0 domains. The Windows 2000 user interface refers to the SAM account name as the "User logon name (pre-Windows 2000)." SAM account names are sometimes referred to as flat names because SAM account names do not use hierarchical naming. For this reason SAM names must be unique in each domain.

**Relative ID (RID)**
The Relative ID is a unique character string issued by the RID Master. When a new security principle is created, it is assigned a unique SID that is a combination of the domain’s SID and a RID. The domain-level **RID Master** role is assigned to the domain controller responsible for ensuring that each domain controller is provided with a unique pool of Relative IDs.
**LDAP**

- **The LDAP database**
  - Each client computer uses Lightweight Directory Access Protocol
  - Distinguished Names (DN) to represent Active Directory objects
  - LDAP can be queried by DN (primary key) and by other attributes
  - LDAP provides DNs and RDNs for objects
  - Two naming conventions:
    - LDAP DN Names (URL Names)
    - LDAP-based Active Directory Names (Canonical Names)

- **Same Name in Two Formats:**
  - LDAP DN Name: `cn=JSmith, ou=Garrison, ou=Infantry, dc=dcTheatreName, divisionName.dc=mil`
  - Canonical Name: `theatreName:divisionName.mil/Infantry/Garrison/JSmith`

<table>
<thead>
<tr>
<th><strong>LDAP DN/RDN Naming Convention</strong></th>
<th>=</th>
<th><strong>Active Directory Naming Convention</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>cn</code> = common name</td>
<td>=</td>
<td><code>cn</code> = common name</td>
</tr>
<tr>
<td><code>ou</code> = organizational unit</td>
<td>=</td>
<td><code>ou</code> = organizational unit</td>
</tr>
<tr>
<td><code>o</code> = organization</td>
<td>=</td>
<td><code>dc</code> = domain component</td>
</tr>
<tr>
<td><code>c</code> = country</td>
<td>=</td>
<td>(not supported)</td>
</tr>
</tbody>
</table>

**Distinguished Names & RDNs (1)**

- **Distinguished Names (DN)**
  - Uniquely identifies an AD object
  - DN = “Relative Path” + “Common Name”
  - Must be unique within an AD Forest

- **Relative Distinguished Names (RDN)**
  - Names the container of an AD object
  - Also based on DNS

- **Example:**
  - LDAP DN Name: `cn=William Gates, ou=staff, ou=NCIOC, dc=335sig, dc=army, dc=mil`
  - Canonical Name: `subDomain:domain.mil/organizationalUnit/organizationalUnit/commonName`

- **Which is the DN?**
  - Wrong order: `335sig.army.mil/NCIOC/staff/William Gates`
  - Missing OU: `335sig.army.mil/NCIOC/William Gates`
  - Relative path: `335sig.army.mil/NCIOC/staff/
  - Relative Path + Common Name: `335sig.army.mil/staff/NCIOC/William Gates`
Distinguished Names & RDNs (2)

Distinguished names identify an object's domain and path

Relative Distinguished names identify an object's domain

Every domain uses at least 2 domain components (DC); Not to be confused with Domain Controllers

- army.mil
- cecom
- ITFSB
- contractors
- Sinn Akaul

DN: CN=Sinn Akaul, OU=cecom, OU=ITFSB, OU=contractors, DC=army, DC=mil

RDN: OU=contractors
Logical Structures

- Objects
- Organizational Units
- Domains
- Domain Trees
  - Parent Domains
  - Child Domains
  - Tree Root
- Forests
  - Forest Root
- Global Catalog

The Logical Structure contains and maintains directory objects, organizational units, domains, trees, forests, and the global catalog.

The parent domain / child domain relationships are hierarchical relationships maintained in the Active Directory by placing one domain as a sub-domain as another. When adding or modifying domains it is possible to specify which tree in the forest the domain will belong to, new trees can be created. When a new tree is created it is the first domain present in that tree and becomes the tree root.

The first domain in a forest is by default the forest root. New domains can be added to existing forest or an entirely new forest may be created at the same time that the domain is constructed. In the scenario that the domain starts a new forest, that domain independently becomes a new domain, domain controller, forest, forest root, tree, and tree root. This ability to create separate, but similar forests at will allows enterprise administrators to maintain entirely separate networks with independent administration partitioned away from the main enterprise. Active Directory treats each forest independently and equal. Separate forest must be re-configured post-creation in order to communicate authentication credentials.

Global catalog is a distributed data repository that contains a searchable, partial representation of every object in every domain in a multi-domain Active Directory.
forest. The global catalog is stored on domain controllers that have been designated as global catalog servers and is distributed through multi-master replication. Searches that are directed to the global catalog are faster because they do not involve referrals to different domain controllers.

In addition to configuration and schema directory partition replicas, every domain controller in a Windows 2000 Server or Windows Server 2003 forest stores a full, writable replica of a single domain directory partition. Therefore, a domain controller can locate only the objects in its domain. Locating an object in a different domain would require the user or application to provide the domain of the requested object.

The global catalog provides the ability to locate objects from any domain without having to know the domain name. A global catalog server is a domain controller that, in addition to its full, writable domain directory partition replica, also stores a partial, read-only replica of all other domain directory partitions in the forest. The additional domain directory partitions are partial because only a limited set of attributes is included for each object. By including only the attributes that are most used for searching, every object in every domain in even the largest forest can be represented in the database of a single global catalog server.
Physical Structure

- Physical Structure
  - Optimizes network traffic
  - Consists of two physical structures:
    - Domain Controllers
    - Active Directory Sites

An Active Directory **Site** is an object that represents “two or more subnets connected by a high-speed, trusted, and reliable link”. Another definition of an AD Site is an administratively defined replication partition. The second definition makes a lot of sense primarily because when Active Directory notices that one or more domain controllers is connected to the other domain controllers in a forest via a "site" link, it will instinctively limit the number of global catalog replications that it will send out across the site link.

A **site link** consist of a the IP address of two or more gateway points for a physical network that is connected by a link that is assumed to be unreliable, slow, and/or untrusted.

**WAN** network links are presumed to be unreliable, slow, and untrusted. However, a **WAN** link is often the only or best means of connection between two portions of a network. Active Directory should be made aware of **WAN links** in its network infrastructure through proper architecture planning to limit the exposure of secure resources and to conserve network bandwidth. By default Active Directory will assume that all domains and entities within the domain are connected with high-speed, trusted, and reliable links.

The Physical structure of Active Directory allows for the utilization of physical network resources and bandwidth for the purpose of replicating directory data.
from one controller to the next. This is accomplished in two main ways. Consider the following scenarios:

**SCENARIO #1: High-Speed Connection & Dedicated Links**
The entire forest and all resources maintained by it are all on the same LAN (or VLAN) that is supported by a high speed dedicated connection. In this scenario it is favorable to have several domain controllers setup throughout the forest to facilitate authentication requests. Since all domain controllers are connected via a dedicated high speed connection (assumable 10+mbps), it is likely a good idea to use the Active Directory domain controller default replication schedule. By default Active Directory domain controllers replicate their data to the other controllers every 15 seconds*. This will have a minimal network impact since the network links are dedicated and high speed.

**SCENARIO #2: Unknown-Speed Connection & Non-Dedicated Links**
In this scenario the entire forest and all of its resources are still maintained on the same LAN (or VLAN), but the physical network connection between one of the domain controllers is in an untrusted area. Perhaps the sole domain controller is held in an unsecure office while the others are in a secure data center. The Active Directory enterprise administrator could place this domain controller on a separate site in order to treat the data updates that it might try to replicate throughout the forest as unacceptable and therefore not allowed. When a portion of an Active Directory "logical" network is divided by a site link the separated portion of the network can be place under specific rules that will ensure not only safe and efficient delivery of data, but also secure boundaries to protect the main site.

Design considerations that determine whether or not use of a site link is wise are exhaustive. Later chapters will cover AD Sites in great detail.

* Active Directory domain controllers do not actually send out replication data every 15 seconds. Instead, a domain controller waits until there is a change to its own global catalog, then it waits an additional 15 seconds before replicating out the GC changes. Each new change after the initial 15 second wait will be updated every 3 seconds instead. This replication process will continue until the domain controller goes a period of time without any new GC changes. More details will come in later chapters.
Active Directory Schema

• A forest-wide definition of object classes and attributes that can be extended
• Schema changes can be redefined or deactivated

<table>
<thead>
<tr>
<th>Examples of attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>accountExpires</td>
</tr>
<tr>
<td>department</td>
</tr>
<tr>
<td>distinguishedName</td>
</tr>
<tr>
<td>directReports</td>
</tr>
<tr>
<td>dNSHostName</td>
</tr>
<tr>
<td>operatingSystem</td>
</tr>
<tr>
<td>repsFrom</td>
</tr>
<tr>
<td>repsTo</td>
</tr>
<tr>
<td>firstName</td>
</tr>
<tr>
<td>lastName</td>
</tr>
</tbody>
</table>
What Is the Global Catalog?

The primary data store for objects located across the forest

A Global catalog is a distributed data repository that contains a searchable, partial representation of every object in every domain in a multi-domain Active Directory forest. The global catalog is stored on domain controllers that have been designated as global catalog servers and is distributed through multi-master replication. Searches that are directed to the global catalog are faster because they do not involve referrals to different domain controllers.

In addition to configuration and schema directory partition replicas, every domain controller in a Windows 2000 Server or Windows Server 2003 forest stores a full, writable replica of a single domain directory partition. Therefore, a domain controller can locate only the objects in its domain. Locating an object in a different domain would require the user or application to provide the domain of the requested object.

The global catalog provides the ability to locate objects from any domain without having to know the domain name. A global catalog server is a domain controller that, in addition to its full, writable domain directory partition replica, also stores a partial, read-only replica of all other domain directory partitions in the forest. The additional domain directory partitions are partial because only a limited set of attributes is included for each object. By including only the attributes that are most used for searching, every object in every domain in even the largest forest can be represented in the database of a single global catalog server.
Forest Level Authentication uses an account’s UPN name to gain access to resources from any domain in the forest.

Forest Level credentials are called upon whenever the “@” symbol is included behind a user’s account username. When this happens, the domain list (Log on to :) portion of the login dialog box becomes disabled. This is because the originating domain no longer matters for a forest level authentication.

Behind the “@” symbol on the username line is the originating domain. This UPN suffix is set to the originating domain’s name by default when the domain controller is brought up. HOWEVER, the UPN suffix can be changed and is therefore NOT the same as a domain level authentication. The fact that the UPN suffix is the same as the originating domain is merely a coincidence of convenience.
In the example above, an account object (*user object*) is attempting to login by not only inputting a username and password, but also a domain of choice.

Domain Level authentication can occur by contacting the originating domain of an account object. However, if that account is attempting authentication while outside of its own domain it will be redirected, whenever possible, to its originating domain for authentication credentials. This process requires that the guest domain controller and the originating domain controller are in two-way transitive trust of one another.
**Constrained Delegation** limits the requesting object to the resources that it has specific access. An alternate way to accomplish resource delegation is for one object to request resources be relayed to it via a middle object acting as a proxy. In this scenario the middle object serving as a relay must be given wide permissions to other secure assets to accommodate all of the requests it may receive from legitimate host objects. However, if an attacking object were to gain full access to the middle object or if the attacking were to pose as the middle object by using the middle object’s credentials the previously secure resources would be compromised. Windows 2000 has this vulnerability since its delegations are not in a constrained space. Windows Server 2003 solves this problem by passing the requesting objects credentials through the middle objects’ layers while blocking access to the final resource.

Constrained Delegation affects user and computer object authentication in Active Directory.
Active Directory uses single master and multi-master operations for replication. Replication allows each domain controller to have an equivalent copy of the domain objects from every other domain. Multi-master operation allows updates to the directory from any domain controller within the forest to be replicated out to all other domain controllers immediately, while Single-master operation requires that each domain send and accept updates to a central controlling server called an Operations Master.

Active directory is a multi-master enabled database. This functionality allows changes to the directory to occur at any domain controller. However, some changes using the multi-master method can cause errors in the Active Directory database. For these types of changes (ex: untrusted/unsecure domain controllers, domains with frequent changes, DCs with limited network connectivity) the use of an operations master is preferred. Operations masters perform updates to the directory in a single-master fashion, meaning that only the domain controller assigned to hold the operations master role is allowed to process the update. This single-master update model prevents conflicting updates from being made to the Active Directory database.
There are five specific operations masters’ tasks in an Active Directory environment. Temporarily ignoring performance limitations, multiple operation master roles can be combined on a single domain controller. The schema master and the domain naming master are forest-wide roles, meaning that only one of each of these types of operations masters is in a forest. The relative identifier (RID) master, infrastructure master, and primary domain controller (PDC) emulator master are domain-wide roles, meaning one of each of these types of operations masters is in each domain in a forest.

Operation masters are used to perform specific tasks and are important to the performance of the directory. They must be available to all domain controllers and directory clients that require their services.

Forest-wide Roles

1. Schema master governs all changes to the schema. The schema contains the master list of object classes and attributes that are used to create all Active Directory objects, such as computers, users, and printers. The domain controller that holds the schema master role is the only domain controller that can perform write operations to the directory schema. These schema updates are replicated from the schema operations master to all other domain controllers in the forest.
2. **Domain naming master** adds and removes domains to and from the forest. The domain naming master is a forest-wide operations master role because it manages the addition and removal of all directory partitions, regardless of domain, in the forest hierarchy. The domain controller that has the domain naming master role must be available in order to perform the additional or removal of domains, application directory partitions, and cross-referenced objects.

**Domain-wide Roles**

3. **Primary domain controller (PDC) emulator** - processes all replication requests from Microsoft Windows NT 4.0 backup domain controllers and processes all password updates for clients that are not running Active Directory enabled client software.

4. **Relative identifier (RID) master** - allocates RIDs to all domain controllers to ensure that all security principals have a unique identifier.

5. **Infrastructure master** - for a given domain it maintains a list of the security principals from the other domains that are members of groups within its own domain.
Domain Controllers

• Domain Controllers (DC)
  – Accepts account changes
  – Provides authentication control
  – Replicates to changes forest

• FSMO: Flexible Single master operations

Each domain requires active and reliable domain controller. Sometimes this is accomplished by placing additional domain controllers in a domain. Objects within a domain must authenticate to prove their credentials and gain access to Active Directory controlled networked resources. There is a benefit to having multiple domain controllers within a single domain. Deploying multiple domain controllers in one domain provides fault tolerance and load balancing. If one domain controller within a domain slows, stops, or fails, other domain controllers within the same domain can provide the necessary directory access, since they contain the same directory data.

In a forest, there are at least five FSMO roles that are assigned to one or more domain controllers. The five FSMO roles are:

1. **Schema Master**: The schema master domain controller controls all updates and modifications to the schema. To update the schema of a forest, you must have access to the schema master. There can be only one schema master in the whole forest.

2. **Domain naming master**: The domain naming master domain controller controls the addition or removal of domains in the forest. There can be only one domain naming master in the whole forest.

3. **Infrastructure Master**: The infrastructure is responsible for updating references from objects in its domain to objects in other domains. At any
one time, there can be only one domain controller acting as the infrastructure master in each domain.

4. **Relative ID (RID) Master**: The RID master is responsible for processing RID pool requests from all domain controllers in a particular domain. At any one time, there can be only one domain controller acting as the RID master in the domain.

5. **PDC Emulator**: The PDC emulator is a domain controller that advertises itself as the primary domain controller (PDC) to workstations, member servers, and domain controllers that are running earlier versions of Windows. For example, if the domain contains computers that are not running Microsoft Windows XP Professional or Microsoft Windows 2000 client software, or if it contains Microsoft Windows NT backup domain controllers, the PDC emulator master acts as a Windows NT PDC. It is also the Domain Master Browser, and it handles password discrepancies. At any one time, there can be only one domain controller acting as the PDC emulator master in each domain in the forest.
How Active Directory Works (1)

- **Account Types**
  - User
  - Computer
  - Printer

- **Groups Types**
  - Domain Local Groups
  - Global Groups
  - Special identity groups

How Active Directory Works (2)

- Load Object Attributes and GPO Rules
- GUID
- SID
- LDAP

How Active Directory Works (3)

Site Replication

How Active Directory Works (4)

User Access to Shared Resource

1. Log On to Windows
2. DNS
3. GC
4. Domain Controller
5. File Server
6. Site Replication

Site 1 → Site 2 → Site 3 → Site 7
Autonomy and Isolation Design

START

Service isolation required? Yes
No
Service autonomy required? Yes
No

Forest level autonomy? Yes
No

Data isolation required? Yes
No

Isolated from service admins? Yes
No

Data autonomy required? Yes
No

Use an existing OU

Separate OU required for data autonomy

Separate forest required for data isolation / autonomy

Separate forest required for service isolation / autonomy

Separate domain required for service autonomy

Separate forest required for data isolation / autonomy
This page intentionally left blank.
TAB

Insert Tab # 3 Here
Domains, Forests, and Trusts
What is a Domain?

- AD Base Element (Building Block)
- NT 4 Compatible
- Physically Implemented on Domain Controllers (DC)
- Border for
  - Replication Traffic
  - System Policies
  - Administration

A domain name uniquely identifies a website. Typing a domain name (also called a URL) into your browser allows you to visit that site.

One unique domain name can be used as a company's web address and for its email addresses.

Domains are much more than just names typed into a web browser. Your domain name is how the world will think of you and your business. It's part of your brand and your marketing plan. You can have multiple domains that each relate to some aspect of your business. A domain can be easy to remember and help establish you on the web.
What is a Tree?

- Single Hierarchical Namespace
  - “us.army.mil” is child domain to “army.mil”
  - “fortgordon.army.mil” is child domain to “army.mil”
  - “forthood.army.mil” is child domain to “army.mil”

- Transitive Trusts created automatically

- Sub-Domains…
  - are child domains
  - must be added to a parent domain
  - Eldest parent (1st domain created) is the tree root

What is a Forest?

- Combination of one or more domain trees

- Separated Namespaces
  - Army.mil
  - Navy.mil

- Automatically created…
  - Transitive Trusts for internal domains
  - One single tree-root
  - New trees must be added to Forest Root
Active Directory Forest

• Active Directory Forest
  – Autonomous
  – Logical Structure
  – Segregation of network trusts, schema, and accessible resources
  – Creates a control boundary

• Two prevailing models
  – Single Forest Model
  – Multiple Forest Model

The domain controllers in both forest 1 and forest 2 have the same schema, configurations, and DNS naming hierarchy. This is acceptable because they are segregated into two autonomous logical networks. The two forests can communicate between each other only because a forest trust has been setup. Each forest's Global Catalog carries information only about its own local forest objects. A meta-directory must be setup to maintain information about the other forest's objects.
Forest Design Considerations

• Organizational
• Legal
• Timescales
• Test Environments

• Operational
• Naming Considerations
• Management Overhead
• External-facing Environments

**Organizational**
Useful when organizations of merging or adjoined units have an existing AD structure and compatibility must be achieved without a complete restructure of the schema. This can be accomplished by setting up separate forest.

**Operational**
Organizations composed of merging or adjoined units may see fit to use separate forest to define similar schema for each unit while avoiding AD conflicts. This can be accomplished by setting up separate forest.

**Legal**
Laws and regulations occasionally constrict design possibilities and dictate constraints on the accessibility of data to sensitive business units. A complete separation of namespace, resources, and administrative control can be accomplished by setting up separate forest.

**Naming Considerations**
Separate forests allow for identical naming schemes thus allowing conflicting domain hierarchies to exist without conflict. This in accomplished by using separate forest because each forest is considered a logically separate and autonomous system.
**Timescales**
Designing or Redesigning an Active Directory schema can be time consuming. The use of multiple forests can assist with the design process by allowing domain hierarchy to be reused without conflicts.

**Management Overhead**
Multiple forest designs must consider management overhead. Single forest designs allow for a more consolidate administrative control, but can present troubleshooting challenges when business units of complex needs are deeply nested between domains and Organizational Units (OUs). A logical separation of complex business units could simplify the administrator’s job; increase troubleshooting speed, and lower management overhead.

**Test Environments**
Separate forest allow for a mirrored schema of the production environment while still limited the impact of a failed action to a non-critical portion of networked resources.

**External-facing Environments**
For the purpose of security isolations, when connecting to external network systems, it is best to stand-up the auxiliary site inside of a separate forest.
Single Forest Schemas

• Advantages:
  – Allows for simple sharing
  – Exchange can be optimize using a single directory
  – Easily replicated Global Catalog
  – Single set of objects; Minimizes object conflicts

• Disadvantages:
  – A change in object behavior could have forest wide impacts
  – Security controls have ripple-effect potential
  – Does not cater to test environments
  – Does not cater to externally-joined enviornments

Multiple Forest Schemas

• Advantages:
  – Separate schemas, DNS hierarchies, and configuration partitions
  – Inter-trust relationships between forest not required
  – Separate Enterprise Admins
  – Minimizes Active Directory vulnerabilities
  – Simplifies environment testing

• Disadvantages:
  – Higher design and implementation cost
  – No simple sharing of network resources
  – Global Catalog searches limited to local forest (requires Meta-Directory)
  – Future forest mergers are difficult and potentially impractical
Forest and Domain Structure

• Create a Forest and a Domain
  – Start → Run → “DCPROMO”
  – The wizard will verify:
    • User is a local Administrator
    • Operating System supports Active Directory
    • No pending Active Directory installs or removals are pending (*reboot first*)
  – Other setup options
    • Setup a new Forest
    • Setup a new Tree
    • Remove Active Directory

To setup a Forest and Domain structure, an Active Directory forest must first exist. If there is no forest already available then a new one may be created. The Active Directory streamlines this entire process by providing a wizard which may be accessed by entering “DCPROMO” in the textbox under the Windows RUN command.

The main purpose of the DCPROMO (Domain Controller Promotion) wizard is to setup domain controllers on new and existing domains. However, no Active Directory domains may exist outside of an Active Directory logical forest. Additionally, several other steps must take place depending on how a domain controller is configured to operate.

To assist the administrator with the various configuration options and dependencies, Microsoft has designed the DCPROMO wizard to not only setup domain controllers, but to also modify existing configurations, add entire new forest, and to validate system requirements. An error that occurs during the installation or setup process could have potentially serious implications and that should be solved before additional Active Directory (AD) configurations occur. Active Directories may be uninstalled, system rebooted, then re-installed if necessary. This often fixes most setup configuration mistakes.
Before a domain can exist there must exist a “logical” controller of that domain… Hence, the domain controller, every domain controller must be a part of a domain tree. Every domain tree must be a part of a forest. When DCPROMO attempts to setup a domain controller it will ask the AD Admin for the name of a forest to join. The AD Admin has the option of setting up a new forest, then providing that forest’s name. By default, adding a new domain to a forest that did not exist makes the domain controller of that domain the forest root and the tree root.
Forest, Trees, and Domains

- Forest – a hierarchical collection of one or more trees
- Tree – a hierarchical collection of one or more domain
- Domain – a logical partition of resources and administrative control

A forest is a collection of one or more Active Directory trees connected by two-way transitive trust relationships. Forest level credentials are universally passed down to all domains in all of the trees contained within the forest. **All domains within an Active Directory forest share a common directory schema, configuration information, and Global Catalog. Only domains within the same tree share a contiguous namespace.** A forest exists as a set of cross-reference objects and trust relationships that are known to the member trees. Transitive trusts at the root domain of each tree provide mutual access to resources.

A Domain Tree is a collection of one or more Active Directory domains. All domains within a tree are assumed to have two way transitive trusts unless otherwise specified.

A Domain is defined by its controller (DC). The domain controller is the administrative access point for assigning Active Directory roles and responsibilities to objects within the domain. A single domain forest constitutes a forest containing 1 forest, 1 forest root, 1 tree, 1 tree root, 1 domain, 1 domain controller, and 1 global catalog server all operating on a single server as an all encompassing domain. In a single domain forest, the existing domain controller is instinctively the forest root, the domain tree root, and the global catalog server.
In many cases, this single DC will also serve as a DNS server for Active Directory. **Although trees in a forest share a common directory schema, configuration, and Global Catalog, they do not form a contiguous namespace.** Trees in the same forest form a namespace that is based on dissimilar DNS root domain names that are noncontiguous. This sharing of common schema and configuration data, in addition to trust relationships between their roots, distinguishes a forest from a set of unrelated trees.
Forest Root

- Defaults to 1st DC
- **Cannot** be deleted
- **Cannot** be changed
- **Cannot** be renamed

**Forest Root Domain**

The first domain created in the forest is called the forest root domain. The forest root domain cannot be deleted, changed, or renamed. When you create a new tree, you specify the root domain of the initial tree, and a trust relationship is established between the root domain of the second tree and the forest root domain. If you create a third tree, a trust relationship is established between the root domain of the third tree and the forest root domain. Because a trust relationship is transitive and bidirectional, the root domain of the third tree also has a two-way trust relationship with the root domain of the second tree. The distinguished name of the forest root domain is used to locate the configuration and schema directory partitions in the namespace. The distinguished names for the Configuration and Schema containers in Active Directory always show these containers as child objects in the forest root domain.

Every domain controller in a forest stores a copy of the configuration and schema directory partitions, and every copy of these partitions has the same distinguished name on every domain controller.

When Active Directory is installed on a Windows Server (2000 or better), configuration and directory schema information is copied from the parent domain to the new server. Updates to configuration and directory schema information
are replicated to all domain controllers throughout the forest. The distribution of this configuration and directory schema information ensures that each domain controller is aware of all other trust-related domains and of the replication topology, which makes finding and using resources in other domains possible.
Transitive Trust Relationships

- Two-Way Transitive Trust  (Automatic)
- Two-Way Transitive Trust  (Explicit)
- One-Way Transitive Trust

A Transitive Trust is a shared agreement between two domain controllers that determines whether or not Global Catalog updates will be accepted from the source provided.

Domains within an Active Directory forest share a common directory schema, configuration information, and Global Catalog. They also have transitive trust relationships that allow users in each domain access to available resources in all other domains in the tree.

If two domains exist in a forest with a Two-Way Transitive Trust relationship, it is understood that each domain will accept Global Catalog updates to their own GC’s from the other domain GCs. This will happen because each domain “trusts” the other domain. It is possible that a domain can “trust” one set of domains in a forest and not trust others.

By default all domain controllers attempt to send out changes about their global catalog every time a change is made (with a 15 second delay).

The directory schema and configuration data are shared because they are stored in separate logical directory partitions that are replicated to domain controllers in every domain in the forest. The data relative to a particular domain is replicated only to domain controllers in the same domain. The Global Catalog is a domain
controller that stores all objects of all domains in an Active Directory forest, which makes it possible to search for objects at the forest level rather than at the tree level.
Global Catalog Server Physical Components

*Active Directory forest, Domain controller, Global catalog server, Ntds.dit*

Active Directory is a distributed directory service in which data is stored as replicas on multiple domain controllers to provide a virtual database that maintains consistency through Active Directory replication. Domain controllers provide the domain-wide distribution of directory data. Global catalog servers provide the forest-wide distribution of directory data in a multi-domain forest.

Active Directory is a distributed directory service in which data is stored as replicas on multiple domain controllers to provide a virtual database that maintains consistency through Active Directory replication. Domain controllers provide the domain-wide distribution of directory data. Global catalog servers provide the forest-wide distribution of directory data in a multi-domain forest.

- **Active Directory forest** The AD forest is the set of domains that comprise the Active Directory logical structure and that are searchable in the global catalog.

- **Domain controller** Server that stores one full, writable domain directory partition plus forest-wide configuration and schema directory partitions. Global catalog servers are always domain controllers.
• **Global catalog server**  Domain controller that stores one full, writable domain plus forest-wide configuration and schema directory partitions, as well as a partial, read-only replica of all other domains in the forest.

• **Ntds.dit**  Database file that stores replicas of the Active Directory objects held by any domain controller, including global catalog servers

**Global Catalog Processes and Interactions**

In addition to its activities as a domain controller, the global catalog server supports the following special activities in the forest:

**User logon:**
- In a multi-domain forest, domain controllers must contact a global catalog server to retrieve any SIDs of universal groups that the user is a member of. Additionally, if the user specifies a logon name in the form of a UPN, the domain controller contacts a global catalog server to retrieve the domain of the user.
- Universal and global group caching and updates: In sites where Universal Group Membership Caching is enabled, domain controllers that are running Windows Server 2003 cache group memberships and keep the cache updated by contacting a global catalog server.
- Global catalog searches: Clients can search the global catalog by specifying port 3268 or by using search applications that use this port. Search activities include:
  - Validation of references to non-local directory objects. When a domain controller holds a directory object with an attribute that references an object in another domain, this reference is validated by contacting a global catalog server.
- Global catalog server creation and advertisement: Global catalog servers register global-catalog-specific service (SRV) resource records in DNS so that clients can locate them according to site. If no global catalog server is available in the site of the user, a global catalog server is located in the next closest site, according to the cost matrix that is generated by the KCC from site link cost settings.
- Global catalog replication: Global catalog servers must either have replication partners for all domains or be able to replicate with another global catalog server. When changes to the PAS occur on, and are replicated between, domain controllers that are running Windows Server 2003, only the updated attributes are replicated. Changes to the PAS that occur on domain controllers that are running Windows 2000 Server prompt a full synchronization of the entire global catalog (all attributes in the PAS are replicated anew to all global catalog servers). For more information about PAS replication, see “Global Catalog Replication” later in this subject.
Global Catalog Authentication

1. Locate any domain controller in the site of the client.

2. The guest DC asks, "What is the DN of JohnSmith3@63bn.us.army.mil?"

3. A DC with a GC responds, "cn=JohnSmith3, ou=civilian, dc=63bn, dc=us, dc=army, dc=mil"

4. The guest DC responds, "Your domain is dc=63bn, dc=us, dc=army, dc=mil"

5. Locate a domain controller in 63bn.us.army.mil
Global Catalog Protocols

The directory system agent (DSA) is the process that provides access to the physical store of directory information that is located on a hard disk. The DSA manages the directory and therefore understands what each directory object and function represents.

The diagram shows the four interfaces into Active Directory and the protocols that package the data according to their specific applications. These protocols and interfaces are the same for all domain controllers and are not specific to global catalog servers. The significance for the global catalog server is that domain controllers use the proprietary RPC replication protocol not only for replication, but also to contact the global catalog server when retrieving universal group membership information and when updating the group membership cache when Universal Group Membership Caching is enabled.

Lightweight Directory Access Protocol (LDAP)
The LDAP is the primary directory service protocol that specifies directory communications. It runs directly over TCP/IP, and it can also run over User Datagram Protocol (UDP) connectionless transports. UDP access is primarily used by the domain controller Locator process and can also be used to query the rootDSE. The Root Directory Server Entry (RootDSE) is a resource in discovering information about LDAP directories that are version 3 compliant.
Clients use LDAP to query, create, update, and delete information that is stored in a directory service over a TCP connection through the TCP default port 389. Global catalog clients can use LDAP to query Active Directory over a TCP connection through the TCP port 3268. Active Directory supports LDAP v2 (RFC 1777) and LDAP v3 (RFC 2251). LDAP v3 is an industry standard that can be used with any directory service that implements the LDAP protocol. LDAP is the preferred and most common way of interacting with Active Directory.

Remote Procedure Call (RPC)
RPC is the Protocol for replication (REPL) and domain controller management communications, (including global catalog server interactions), NSPI address book communications, and SAM-related communications. RPC is a powerful, robust, efficient, and secure inter-process communication (IPC) mechanism that enables data exchange and invocation of functionality residing in a different process. That different process can be on the same computer, on the local area network (LAN), or across the Internet.

Simple Mail Transfer Protocol (SMTP)
Protocol for replication communications when a permanent, “always on” network connection does not exist between two domain controllers. SMTP is used to transport and deliver messages based on specifications in Request for Comments (RFC) 821 and RFC 822. SMTP can replicate only the configuration and schema directory partitions and global catalog read-only replicas (not writable domain data).

Global Catalog Data Store Interfaces
Lightweight directory access protocol (LDAP), REPL, Name Service Provider Interface (NSPI) by which Messaging API (MAPI), SAM

LDAP
LDAP is the primary interface for Active Directory access. Directory clients use LDAP v3 to connect to the DSA through the LDAP interface. The LDAP interface is part of Wldap32.dll. LDAP v3 is backward compatible with LDAP v2.

REPL
The replication management interface that provides functionality for finding data about domain controllers, converting the names of network objects between different formats, manipulating service principal names (SPNs) and DSAs, and managing replication of servers.

NSPI/MAPI
Name Service Provider Interface (NSPI) by which Messaging API (MAPI) clients access Active Directory. Messaging clients gain access to Active Directory by using MAPI address book providers. For compatibility with existing messaging clients, Active Directory supports the NSPI/RPC address book provider, which provides directory access, for example, to find the telephone number of a user. * The NSPI (MAPI) interface is provided only for support of legacy Microsoft Outlook clients. Development against this interface is no longer supported.
SAM
Sam is the Proprietary interface for connecting to the DSA on behalf of clients that run Windows NT 4.0 or earlier. These clients use Windows NT 4.0 networking APIs to connect to the DSA through SAM. Replication with Windows NT 4.0 backup domain controllers (BDCs) occurs through the SAM interface as well.
There are five specific operations masters’ tasks in an Active Directory environment. Temporarily ignoring performance limitations, multiple operation master roles can be combined on a single domain controller. The schema master and the domain naming master are forest-wide roles, meaning that only one of each of these types of operations masters is in a forest. The relative identifier (RID) master, infrastructure master, and primary domain controller (PDC) emulator master are domain-wide roles, meaning one of each of these types of operation masters is in each domain in a forest.

Operation masters are used to perform specific tasks and are important to the performance of the directory. They must be available to all domain controllers and directory clients that require their services.

Forest-wide Roles

1. Schema master governs all changes to the schema. The schema contains the master list of object classes and attributes that are used to create all Active Directory objects, such as computers, users, and printers. The domain controller that holds the schema master role is the only domain controller that can perform write operations to the directory schema. These schema updates are replicated from the schema operations master to all other domain controllers in the forest.
2. **Domain naming master** adds and removes domains to and from the forest. The domain naming master is a forest-wide operations master role because it manages the addition and removal of all directory partitions, regardless of domain, in the forest hierarchy. The domain controller that has the domain naming master role must be available in order to perform the additional or removal of domains, application directory partitions, and cross-referenced objects.

**Domain-wide Roles**

3. **Primary domain controller (PDC) emulator** - processes all replication requests from Microsoft Windows NT 4.0 backup domain controllers and processes all password updates for clients that are not running Active Directory enabled client software.

4. **Relative identifier (RID) master** - allocates RIDs to all domain controllers to ensure that all security principals have a unique identifier.

5. **Infrastructure master** - for a given domain it maintains a list of the security principals from the other domains that are members of groups within its own domain.
TAB

Insert Tab # 4 Here
Sites and Replication
What are “Sites”?  

- **Sites:**
  - AD Physical Structure
  - Object containing IP subnet ranges and replication schedules information
  - Can be renamed
  - Names must not exceed 63 characters
  - Mostly used to designate parts of a network connected by WAN
  - One or more IP subnets can be assigned to a site
  - Administrators can limit replication attempts

Sites help define the physical structure of a network. A set of IP subnets and address ranges define a site, which in turn defines a group of domain controllers that have similar speed and cost. Sites consist of server objects. Each server object contains connection objects. It is these connection objects that enable replication.

Sites in Active Directory are used to group servers into container that mirror the physical layout of your network. When you create the first domain controller, a default site called “Default First Site” is created. The site name can later be changed. Additional domain controllers may be added to the existing site or used to established new sites using **DCPROMO**.

Sites allow Active Directory administrators to limit the replication attempts of Active Directory over specified network links.
The Purpose of Sites (1)

- **Benefits of separate sites:**
  - Partitions client use of the network
  - Further optimizes the exchange of directory information
  - Facilitates administration by centralizing resources such as configuration information
  - Fine tunes replication behavior
The Purpose of Sites (2)

- Unreliable or Expensive Links

Sites may be used to limit or regulate bandwidth between domain controllers. Sites don’t necessarily have to be separated by WAN links; however, this is usually the most applicable use.
The Purpose of Sites (3)

• Un-trusted Links or Logically Divided Networks

Sites may also be used to organize trust relationships with other domain controllers. For example, domain controllers existing in certain sites might be administratively designed not to allow replication to domain controllers in other sites. This could be particularly useful if a set of domain controllers in a remote site wants to allow updates, but only apply them to the domains or forest segment within their own site.
Bridge-Head Server

- Intra-site replication is accomplish via bridge-head servers

- The network connection with the lowest cost wins

Within a site, each domain controller has the potential to act as a bridgehead server if needed. If you specify one or more computers to be preferred bridgehead servers, only a single domain controller acts as the bridgehead server at any given time. Although Active Directory does select another preferred bridgehead server for use in the event of failure, while this rerouting takes place, there is a period of time in which no inter-site replication will occur.

If you have good connectivity between a group of servers and clients, create sites and establish bridgehead servers in each site to provide load balancing. You can divide the potential for client requests between servers by establishing separate sites. Even though one site may be well connected to other sites, this reduces replication since the server or servers in the site handle all requests for clients within the site.

Where redundant network links are available, Active Directory will pick the link with the lowest cost.
What Is the Global Catalog?

• The primary to stores and locate objects across the forest.

A **Global Catalog** is a repository that contains a subset of the attributes of all objects in Active Directory.

It is the primary method by which Active Directory stores and locates objects for the purpose of replication. It is the object repository that domain controllers use to compare notes. Global Catalogs do not store complete object data. Rather they store only a subset of attributes that uniquely identify each object.

When one DC is found to have objects that another DC is unaware of, it will attempt to update itself by means of replication. Generally, the GC is updated on every domain controller anytime there is a modification. There are specific exceptions to this generalization such as a Global Catalog that resides on a domain controller across a site link might not be allowed to share its GC with the other domain controllers, despite the fact that the other domain controller could still be allowed to update their GCs across the site link. We would consider the sole domain controller and every other controller in the auxiliary site to be untrusted. An untrusted site is not the same as an untrusted physical network link. Either however could be a reason why a domain controller and hence its global catalog is not a part of the main site.

Global catalogs can be shared among domain controllers. Whenever an object attempts to authenticate from outside of its own domain, credentials can be
validated on the guest domain because the object credentials are present in the
guest’s domains global catalog. This only works however if the object is
attempting authentication using forest level credentials or when the object’s own
domain is trusted by the guest domain. In the case of a user account object, the
domain which the object belongs must be provided in addition to the username
and password.
Site Topology

- When to establish a Site

- Planning for new Sites

- Specifying the Bridgehead Server

- Originating time is the time of the originating write based on the system clock.

- Originating DC is the directory system agent

When to establish a Site:
If replication between all domain controllers must take place on the same schedule, it may seem to make sense to assign all those domain controllers to the same site. This will achieve a uniform replication schedule, but it limits your ability to adjust directory replication behavior.

By creating separate sites, you can fine-tune replication according to site conditions. If good connectivity exists between a group of servers and clients; Create sites and establish bridgehead server. You can divide the potential for client requests between servers by establishing separate sites. Even though one site may be well connected to other sites, this reduces replication since the server or servers in the site handle all requests for clients within the site.

Planning for new Sites:
When planning your sites, consider which domain controllers each computer should authenticate. During the logon process, a client attempts to find a domain controller in its local site. To ensure this happens efficiently and reliably, your fast network connections need to map directly to low-cost site links in your site topology. Associating client subnets to sites they are in ensures efficient client access to resources because the clients will have good connectivity to the resources.
Specifying the Bridgehead Server:
Within a site, each domain controller has the potential to act as a bridgehead server if needed. If you specify one or more computers to be preferred bridgehead servers, only a single domain controller acts as the bridgehead server at any given time. Although Active Directory does select another preferred bridgehead server for use in the event of failure, while this rerouting takes place, there is a period of time in which no inter-site replication will occur.

To mitigate the network effects resulting from failure of the bridgehead server; large, well-connected sites may be split into smaller sites. With a single large site, if the bridgehead is unavailable, the resulting halt in inter-site replication (not to be confused with intra-site replication) has a wider effect. It should be noted however that with each new site, directory design, management, troubleshooting, and configuration changes becomes increasingly complex.
Intro to AD Replication

• Replication
  – process of updating information in Active Directory from one domain controller to other domain controllers

• Types:
  – Intra-site replication
  – Cross-site replication

Active Directory object information includes name, location, type, access level, organizational grouping, and a coded identifying marker (GUID).

**Replication** is the process of updating information in Active Directory from one domain controller to other domain controllers. The replication process synchronizes the movement of updated information between domain controllers. Synchronization ensures that all information in Active Directory is available to all domain controllers and clients' computers across the network. Each domain controller has a “like” copy of the global catalog.

**Intra-Site** replication is the replication of Active Directory object information within a site's specified subnets. By default every domain created is in the same original site called “DEFAULT FIRST SITE”.

**Cross-Site** replication is the replication of Active Directory object information to domain controllers that belong to sites other than the updating site of the propagating DC.
Intra-Site Replication

1. Change on Trigger
2. Change Notification
3. Replication Latency
4. Urgent Replication
5. Convergence
6. Propagation
7. Conflicts
8. GUID

When a change is made to Active Directory, such as adding removing a user or modifying the user’s object attributes, the domain controller where the change is made updates its own database to store the change. HOWEVER, if the change is not propagated to any other domain controller other than the originating DC, the user’s changes will not be enabled throughout the forest. This may lead to an undesired access denial or security breach.

Active Directory is design to allow each domain controller to replicate its database information to all other domain controllers in a very controlled manner.

**Change on Trigger**
Active Directory will replicate changes whenever any of the following occur:
- Addition, deletion, or modification of objects
- Name changes to an object or an object containers

**Change Notification**
When either of the above changes is made to a domain controller’s database a change notification’ is sent to all other domain controllers within the same forest and the same site.
Replication Latency
After an initial change is made to an AD object, there is a 15 second countdown until the effected domain controller attempts to replicate its data to all the other DCs in the site. The 15 second delay is the default delay and can be changed. The purpose of not replicating to domain controllers in other sites every time a change is made to AD is because of the assumption that the unique circumstances of the domain controllers in the other site or network link between them and the originating site is flawed and replication should not occur as frequently as it normally would.

When the replication process has begun the domain controller will wait only an additional 3 seconds to replicate each additional AD change the replication process is complete. The key to remember is that replication waits until a change, then 15 seconds, then 3.

Urgent Replication
Security updates, such as the removal or granting of access permissions to users or computers, do not wait the typical 15 seconds. These updates are considered urgent and are replicated immediately.

Convergence
Active Directory will attempt to propagate the changes from one domain controller to all other domain controllers within the site. If there is more than one DC with updates to be propagated, convergence is still achieved.

Propagation Dampening
To prevent re-propagating objects which have no new changes or that have already been replicated, Active Directory assigns a USN (Unique Sequence Number) to track which objects remain to be sent.

Conflicts
Occasionally the replication process causes update conflicts, if two domain controllers attempt to update the same object but with inconsistent data it is said that the updates are in conflict. This causation can be minimized by limiting the domain controllers that are allow to propagate their changes to the forest. If conflict still arises, Active Directory has an internal mechanism to resolve these types of conflicts which commonly results from conflicting object attributes, deleted object containers, and RDN (Relative Distinguished Name) conflicts.

Globally Unique Identifier (GUID)
Active Directory requires a truly unique identifier that it may use to accurately identify each and every object throughout an Active Directory forest. **REMEMBER that each forest is treated as an autonomous system and is therefore considered completely separate.** This Globally Unique Identifier (GUID) carries with it information about its forest, its creation domain controller, and a reference to its own database entry. GUIDs are like street addresses, they carry just enough information to accurately redirect someone to a location of more information.
Cross-Site Replication

- Schedule
- Replication Period
- Replication Path

Schedule
Site Replication occurs at given intervals specified when the site is setup. These intervals specify the times and dates that Active Directory can attempt to replicate across a given site link.

Replication Period
During the specified replication schedule (or available window when two may replicate across a given link), Active Directory also allows administrators to limit the number of replication attempts by indicating the replication period. Active Directory will attempt to replicate across site links starting at each time and date given by the replication schedule.

While replication schedule window is open Active Directory will re-attempt replication every cycle as indicated by the replication period. For instance, if the replication schedule calls for replication every day between 2:00 AM and 4:00 AM, then the replication window will be open for 3 hours and 59 minutes. If the replication period is set to 30 minutes, AD will attempt to replicate at 2:00 AM, 2:30 AM, 3:00 AM, and 3:30 AM. AD will not attempt to replicate again at 4:00 AM because at 4:00 AM the replication window is closed. Replication must occur at least once while the window is open. Therefore AD will attempt to replicate to full convergence immediately when the window initially opens.
Replication Path
Active Directory, by default, sets up a two way ring replication path. The data can travel in both directions around the ring which provides redundancy and reliability. The two types of replication occur in the path are Direct replication and Transitive replication.

Direct replication is when replication is provided from a primary source of data, such as an originating domain controller.

Transitive replication is when replication is provided from a secondhand or replicated source of data, such as a non-originative domain controller.

The Knowledge Consistency Checker (KCC) runs on all domain controllers. The KCC maintains a list of connections, called a replication topology, to other domain controllers in the site. The KCC ensures that changes to any object are replicated to all site domain controllers and updates go through no more than three connections. Also an administrator can configure connection objects. The KCC uses information provided by the administrator about sites and subnets to automatically build the Active Directory replication topology.
Key Replication Points

- Changes occur at the attribute level

- Linked-value and Multi-value replicates as a single change…
  - But not always in same transmission

- Replication continues until all updates are transmitted

- Replication uses “Store-and-Forward”

- Multi-master conflicts resolve without clock synchronization

- Partially written objects are not possible because of “Transactions”

The following key points are central to understanding the behavior of Active Directory updates:

- Changes occur at the attribute level…
  - Only the changed attribute value is replicated, not the entire object
  - If an attribute value has changed multiple times between replication cycles, only the current value is replicated
  - An attribute value could potentially changed several times between replication cycles when sites are involved

- The smallest change that can be replicated in Active Directory is an entire attribute…
  - Even if the attribute is linked or multi-valued
  - Multi-valued or linked attributes replicate to all values as a single change
  - Called **linked-value replication** *(Server 2003)*
  - The individual values of a multi-valued attribute are replicated separately when the forest functional level is Windows Server 2003 interim or Windows Server 2003
  - An example of a **Linked attribute** is the “member of” attribute of a group object
• An example of a **Multi-valued attribute** is the “members” attribute of a group object; Can store a maximum of 1,300 entries

• Replication is…
  • **Store-and-Forward**: meaning that a domain controller will temporary store an attribute change before replicating to other domain controllers
  • A **sequentially scan** through a set of connected domain controllers that host directory partition replicas

• **Multi-master conflict** can resolve without depending on clock synchronization

• Partially written objects to the database are not possible and a consistent view of the object is maintained…
  • Transactions ensure that an entire process is completed before permanently committing changes to a database
  • Replicated updates to large numbers of values in linked multi-valued attributes are not always guaranteed to be applied in the same transaction
  • Omitted updates are guaranteed to be applied in subsequent transactions in the same replication cycle
  • After a replication cycle is initiated, all available changes to a directory partition on the source domain controller are sent to the destination domain controller, including changes that occur while the replication cycle is in progress
The GUID and USN have been simulated in this diagram. The USN keeps track of the order with which an object should be replicated.

The GUID cannot be used to track the convergence process because the GUID number does not change. GUIDs stay with each object and are only removed when that object no longer exists. A specific GUID will always belong to a unique and specific object.

The USN, however, can refer to multiple objects depending on the perspective of the reviewer. While each GUID is unique throughout the forest, each USN is unique only to each DC, meaning that DCs referring to the same object GUID might have two separate reference USNs.

Each DC uses its own USN scheme and has a method of determining if it is out of synch with the rest of the forest.

The proper way to describe “‘Semi-Full’ Convergence” in Active Directory is ‘loose convergence’.
Replication Process (1)

• Latency
  – The required time for all updates to be completed throughout all domain controllers on the network domain or forest.

• Convergence
  – The state at which all domain controllers have the same replica contents of the Active directory database.

• Loose Convergence.
  – The state at which all changes to the database are not yet replicated throughout all controllers in the database (not converged).

Active Directory replication is subject to latency in its attempt to update changes to other domain controllers. The initial delay in domain controller’s replication is caused by waiting 15 seconds after an attribute change has been made before beginning the replication process. Each subsequent delay is every 3 seconds. The 3 second count-down terminates after the originating DC has successfully replicated its changes throughout the directory.

When all changes have been replicated throughout the directory (typically the entire local forest) convergence is said to be achieved.

Since a domain controller does not immediately attempt to achieve full convergence after a change has been made, as in the case of the 15 second / 3 second delays and inter-site replication, we consider Active Directory to enforce loose convergence.
Replication Process (2)

- **Convergence**

  - **Latency** - The required time for all updates to be completed throughout all domain controllers on the network domain or forest.
  - **Convergence** - The state at which all domain controllers have the same replica contents of the Active directory database.
  - **Loose consistency** - The state at which all changes to the database are not yet replicated throughout all controllers in the database (not converged).
Replication Process (3)

**Step 1** - Replication-related Data on DC1 When a User Object is Created

**Table:**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Local USN</th>
<th>Version</th>
<th>Originating Time</th>
<th>Originating DC</th>
<th>Originating USN</th>
</tr>
</thead>
<tbody>
<tr>
<td>cn</td>
<td>Jeff Smith</td>
<td>4711</td>
<td>1</td>
<td>2000-09-10 10:49:09</td>
<td>-D1_GUID-</td>
<td>4711</td>
</tr>
<tr>
<td>userPassword</td>
<td>6ba87537-5656-</td>
<td>4711</td>
<td>1</td>
<td>2003-09-10 10:49:09</td>
<td>-D1_GUID-</td>
<td>4711</td>
</tr>
<tr>
<td>sAMAccountName</td>
<td>JsSmith</td>
<td>4711</td>
<td>1</td>
<td>2003-09-10 10:49:09</td>
<td>-D1_GUID-</td>
<td>4711</td>
</tr>
<tr>
<td>userPrincipalName</td>
<td><a href="mailto:JsSmith@contoso.com">JsSmith@contoso.com</a></td>
<td>4711</td>
<td>1</td>
<td>2003-09-10 10:49:09</td>
<td>-D1_GUID-</td>
<td>4711</td>
</tr>
</tbody>
</table>

**Step 2** - Replication-related Data on DC2 When a New User Object is Replicated From DC1

**Table:**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Local USN</th>
<th>Version</th>
<th>Originating Time</th>
<th>Originating DC</th>
<th>Originating USN</th>
</tr>
</thead>
<tbody>
<tr>
<td>cn</td>
<td>Jeff Smith</td>
<td>1746</td>
<td>1</td>
<td>2000-09-10 10:49:09</td>
<td>-D1_GUID-</td>
<td>4711</td>
</tr>
<tr>
<td>userPassword</td>
<td>6ba87537-5656-</td>
<td>1746</td>
<td>1</td>
<td>2003-09-10 10:49:09</td>
<td>-D1_GUID-</td>
<td>4711</td>
</tr>
<tr>
<td>sAMAccountName</td>
<td>JsSmith</td>
<td>1746</td>
<td>1</td>
<td>2003-09-10 10:49:09</td>
<td>-D1_GUID-</td>
<td>4711</td>
</tr>
<tr>
<td>userPrincipalName</td>
<td><a href="mailto:JsSmith@contoso.com">JsSmith@contoso.com</a></td>
<td>1746</td>
<td>1</td>
<td>2003-09-10 10:49:09</td>
<td>-D1_GUID-</td>
<td>4711</td>
</tr>
</tbody>
</table>

**Domain Controller (DC)** - a server that responds to security authentication requests (logging in, checking permissions, etc.) within a Microsoft Windows Server domain.

**Unique Sequence Number (USN)** - Each object has an Update Sequence Number (USN), and if the object is modified, the USN is incremented. This number is different on each domain controller.

**Stamps** = Version + Originating Time + Originating DC. Each object has a stamp with the version number, timestamp, and the GUID of the domain controller where the change was made.

**SAM Account Name** – an account name is required for compatibility with Windows NT 3.x and Windows NT 4.0 domains. The Windows 2000 user interface refers to the SAM account name as the "User logon name (pre-Windows 2000). “ SAM account names are sometimes referred to as flat names because, unlike DNS names, SAM account names do not use hierarchical naming. Because SAM names are flat, each one must be unique in the domain.

**User Principle Name** - are based on the IETF RFC 822, Standard for the Format of ARPA Internet Text Messages. A UPN is a friendly name assigned by an administrator that is shorter than the LDAP distinguished name used by the
system and easier to remember. The UPN is independent of the user object's DN and is used as a forest level authentication method.
Replication Process (4)

Step 3 - Replication-related Data on DC2 After the User Password Value Has Been Changed on DC2

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Local USN</th>
<th>Version</th>
<th>Originating Time</th>
<th>Originating DC</th>
<th>Originating USN</th>
</tr>
</thead>
<tbody>
<tr>
<td>cn</td>
<td>Jeff Smith</td>
<td>1746</td>
<td>1</td>
<td>2000-09-10:10:49:03</td>
<td>&lt;DC1 GUID&gt;</td>
<td>4711</td>
</tr>
<tr>
<td>userPassword</td>
<td>sEIP569991@2</td>
<td>2453</td>
<td>2</td>
<td>2000-09-10:11:53:29</td>
<td>&lt;DC2 GUID&gt;</td>
<td>2453</td>
</tr>
<tr>
<td>sAMAccountName</td>
<td>JSmith</td>
<td>1746</td>
<td>1</td>
<td>2000-09-10:10:49:03</td>
<td>&lt;DC1 GUID&gt;</td>
<td>4711</td>
</tr>
<tr>
<td>userPrincipalName</td>
<td><a href="mailto:JSmith@contoso.com">JSmith@contoso.com</a></td>
<td>1746</td>
<td>1</td>
<td>2000-09-10:10:49:03</td>
<td>&lt;DC1 GUID&gt;</td>
<td>4711</td>
</tr>
</tbody>
</table>

Step 4 - Replication-related Data on DC1 After the Password Change Has Replicated to DC1

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Local USN</th>
<th>Version</th>
<th>Originating Time</th>
<th>Originating DC</th>
<th>Originating USN</th>
</tr>
</thead>
<tbody>
<tr>
<td>cn</td>
<td>Jeff Smith</td>
<td>4711</td>
<td>1</td>
<td>2000-09-10:10:49:03</td>
<td>&lt;DC1 GUID&gt;</td>
<td>4711</td>
</tr>
<tr>
<td>userPassword</td>
<td>sEIP569991@2</td>
<td>5040</td>
<td>2</td>
<td>2000-09-10:11:53:29</td>
<td>&lt;DC2 GUID&gt;</td>
<td>2453</td>
</tr>
<tr>
<td>sAMAccountName</td>
<td>JSmith</td>
<td>4711</td>
<td>1</td>
<td>2000-09-10:10:49:03</td>
<td>&lt;DC1 GUID&gt;</td>
<td>4711</td>
</tr>
<tr>
<td>userPrincipalName</td>
<td><a href="mailto:JSmith@contoso.com">JSmith@contoso.com</a></td>
<td>4711</td>
<td>1</td>
<td>2000-09-10:10:49:03</td>
<td>&lt;DC1 GUID&gt;</td>
<td>4711</td>
</tr>
</tbody>
</table>
Replication Request Filtering

- Request Filters:
  - Up-to-datedness vector
  - High-watermark

- Replication Partners:
  - Direct replication partner
  - Transitive replication partner

Destination domain controllers use the originating USN to track changes they have received from other domain controllers with which they replicate. When requesting changes from a source domain controller, the destination informs the source of the updates it has already received so that the source never replicates changes that the destination does not need.

Two values are used by source and destination domain controllers to filter updates when the destination requests changes from the source replication partner:

1. **Up-to-dateness vector** - The current status of the latest originating updates to occur on all domain controllers that store a replica of a specific directory partition.

2. **High-watermark (direct up-to-dateness vector)** - The latest originating update to a specific directory partition that has been received by a destination from a specific source replication partner during the current replication cycle.

On Server 2003 domain controllers, the up-to-dateness vector includes a timestamp that represents the last time the local (destination) domain controller has completed a full replication cycle with the source domain controller. The replication cycle may have occurred directly (direct replication partner) or
indirectly (transitive replication partner). The timestamp is recorded whether or not the local domain controller actually received any changes from the partner. By examining the timestamps, a domain controller can quickly identify other domain controllers that are not replicating. Warning messages are posted to the event log on each domain controller when non-replicating partners are discovered. These warning messages may be found under Event ID 1864 in the Directory Services event logs.

The high-watermark, or direct up-to-dateness vector, is a value that the destination domain controller maintains during replication to keep track of the most recent attribute change that it has received from a specific source domain controller for an object in a specific directory partition. When sending changes to a destination domain controller, the source domain controller provides the changes in increasing order of uSNChanged.

Although the uSNChanged values from the source domain controller are not stored on objects at the destination domain controller, the destination domain controller keeps track of the uSNChanged value of the most recent object that was successfully updated from the source domain controller for a specific directory partition. This USN is called the destination’s high-watermark with respect to the directory partition and the source domain controller.
Conflict Resolution Stamp

• Version number increased by 1

• Conflict Resolution Stamp
  – The stamp that is applied during an originating write has the following three components:
  – The version is a number that is incremented for each originating write. The version of the first originating write is 1. The version of each successive originating write is increased by 1.
  – The originating time is the time of the originating write (rounded to the nearest whole second) based on the system clock of the domain controller that performed the write.
  – The originating DC is the directory system agent (DSA GUID) of the domain controller that performed the originating write.

Conflict Resolution Stamp

The stamp that is applied during an originating write has the following three components:

1. The version is a number that is incremented for each originating write. The version of the first originating write is 1. The version of each successive originating write is increased by 1.

2. The originating time is the time of the originating write (rounded to the nearest whole second) based on the system clock of the domain controller that performed the write.

3. The originating DC is the directory system agent (DSA GUID) of the domain controller that performed the originating write.

When stamps are compared, the version is the most significant, followed by the originating time and then the originating DC. If two stamps have the same version, the originating time almost always breaks the tie.
This page intentionally left blank.
Users, Groups, and Computer Accounts
Default Local Accounts

- Administrator
- Guest
- Help Assistant ("Security_<rand char>")

Access to Windows Server 2003 requires that users successfully authenticate themselves with a domain user account. Three user accounts are automatically created when Windows Server 2003 is installed. These three accounts are the “Administrator” account, the “Guest” account, and the “Help Assistant” account named “Support_”.

1. The Administrator account is used to initially configure the system and create additional user accounts. This account profile is unrestricted and may change any configuration on the local system.

2. The Guest account is a basic, very limited access user account that may be used to grant users access without giving full administrative privileges.

3. The Support account is the primary account used for remote assistance sessions. This account is managed by the Remote Desktop Help Session Manager services and is disabled by default for security purposes.

The Administrator account cannot be deleted but it can be locked out or disabled. It can be renamed and a password can be added and changed. Assuming security is not a concern; the Administrator password may be assigned a blank password.
Like the Administrator account, the Guest account cannot be deleted but it can be disabled or locked. It can also be renamed, password protected, or left unsecure. Unlike the Administrator account, changes to the environment are NOT retained while under the Guest account. The Guest account can be an overlook security liability. A good practice is to rename, password protect, and disable the Guest account. Re-enable only when needed.
Domain Accounts

- Account Scopes
  - Local
  - Domain

- Accounts Types
  - User Accounts
  - Computer Accounts
  - Group Accounts

Accounts may be made for users, computers, printers, or a collection of other accounts.

As the titles suggest, user accounts are designed for human users, computer accounts are for any non-printing machine that identifies itself electronically as a computer, and printer accounts are designed for printing machines that identify themselves electronically as a printer. In truth, printer accounts are modified versions of computer accounts. It is important to note that not all computers or printers are compatible with Active Directory accounts. For a complete listing of verified compatible hardware, consult Microsoft’s hardware compatibility list. Devices not listed may still be compatible, but might require additional drivers and/or attachments.

Group accounts are a container for one or more users, computers, or printer accounts that all share common functional requirements. Functional settings changed to a group account affect all of the accounts within the group. Group accounts are one of the cornerstones of Active Directory resource management. Group accounts are also permitted to contain other groups.

User accounts and Computer accounts go hand-in-hand. For a human user to access to computer terminal on an Active Directory controlled network, that user must either login using an Active Directory enrolled “domain” account or login
using a non-domain (or local system) account. Local system accounts will not have the ability to gain access to Active Directory controlled resources.

When a user logs on to a domain computer using a domain account, the settings for both the computer and the user are loaded. If there are configuration conflicts, generally the computer settings will win.

Domain accounts may be added, modified, and/or deleted by way of GUI, command prompt, special utilities, or command script files. Whatever, the method, users accounts should be considered serious security risks and should be disabled until needed. Computer domain accounts are created automatically when an unrecognized computer joins the domain. Computer domain accounts can also be created manually through the same methods as user accounts.
**Built-in Security**
**Domain Local Groups (1)**

- Administrators (Administrator, Domain Admin, Enterprise Admin)
- Guests (Domain Guests, Guests)
- Users (Domain Users, Authenticated Users)
- Cert Publishers (Domain Local)
- Debugger Users (Domain Local)
- HelpServices Group (Domain Local)
- RAS and IAS Servers (Domain Local)
- Telnet Clients (Domain Local)
- Pre-Windows 2000 Compatible Access (anonymous logon, Everyone)

**Built-in Security**
**Domain Local Groups (2)**

- Domain Admins (Global)
- Domain Computers (Global)
- Domain Controllers (Global)
- Domain Guests (Global)
- Domain Users (Global)
- Group Policy Creator Owners (Global)
- Enterprise Admins (Universal)
- Schema Admin (Universal)
Common Domain Account Features

- Password Security
- Identification
- Roaming
- Auditing
- Permissions
- User Rights
- Environment Layout

All domain-based accounts contain the following attributes for authentication and security purposes:
- Password Security
- Permissions
- Identification
- User Rights ("User Roles")
- Roaming
- Environment Layout
- Auditing
Group Types & Scopes

Group Types
– Distribution Groups
– Security Groups

Group Scopes
– Domain Local
– Global Groups
– Universal Groups

There are two types of domain group types for Server 2003: Distribution Groups and Security Groups

1. **Distribution Groups** are used only with email applications (ex. MS Exchange). These groups are used to send messages to collections of users. Distribution groups are not security enabled.

2. **Security Groups** are used to assign rights and permissions to groups of users and computers. The permissions assigned dictate which resources an account holder or group of account holders can access on the network. This access granting ability is not to be confused with network connectivity. It is assumed that all network resources with the Active Directory infrastructure have full connectivity at all times unless otherwise stated.

Security Groups can more effectively be utilized with **nesting**. Nesting allows the inclusion of one group inside of another group. The “nested” or “child” group then inherits all of the rights and restrictions of its parent group. Nesting may go several levels deep; however, deep nesting creates an administrative and management dilemma. It is very possible for a user or group to have conflicting rights and restrictions. Nesting further complicates this by making troubleshooting more obscure.
There are three group scopes: Domain Local, Global Groups, and Universal Groups.

1. Domain Local
   a. Membership
      1. (Mixed Mode) Domain Local groups can contain User Accounts and Global Groups from any domain
      2. (Native Mode) Domain Local groups can contain User Accounts, Global Groups, Universal Groups from any trusted domain, and Domain Local groups from the same domain.
   b. Can be a member of
      1. (Mixed Mode) Domain Local groups cannot be a member of any other group
      2. (Native Mode) Domain Local groups can be a member of domain local groups from the same domain
   c. Scope
      Domain Local groups are visible only to their own domain
   d. Permissions
      Permissions assigned to Domain Local groups are only applied to the domain where the domain local group exists
Group Scopes (2)

Global Groups:

– Membership…

– Member of…

– Scope…

  Global groups are visible in their own domain and all other trusted domains

– Permissions…

  Permissions assigned to Global groups are applied to all trusted domains

There are three group scopes: Domain Local, Global Groups, and Universal Groups.

2. Global Groups
   a. Membership
      1. (Mixed Mode) Global groups can contain user accounts from the same domain
      2. (Native Mode) Global groups can contain user accounts and Global Groups from the same domain
   b. Can be a member of
      1. (Mixed Mode) Global groups can be a member of domain local groups in any trusted domain
      2. (Native Mode) Global groups can be a member of universal and domain local groups from any domain and can be a member of global groups in the same domain
   c. Scope
      Global groups are visible in their own domain and all other trusted domains, which by default includes all domains in the forest
   d. Permissions
      Permissions assigned to Global groups are applied to all trusted domains
There are three group scopes: Domain Local, Global Groups, and Universal Groups.

3. Universal Groups
   a. Membership
   1. (Mixed Mode) Universal groups cannot be created in Windows 2000 mixed mode
   2. (Native & 2003 Server Mode) Universal groups can contain user accounts, global groups, and other universal groups from any domain in the forest
   b. Can be a member of
   1. (Mixed Mode) Universal groups are not available in Windows 2000 mixed mode
   2. (Native Mode & 2003 Server Mode) Universal groups can be a member of domain local and universal groups from any domain
   c. Scope
      Universal groups are visible in all domains in a forest
   d. Permissions
      Permissions assigned to Universal groups are applied to all domains in the forest
Delegation of Administration

- Administrative rights can be delegated to Users or Groups
- Administrative rights **cannot** be delegated to OUs
- Delegation possible via Wizards
- Chief Troubleshooting Challenges
  - NESTING - All objects must be viewed separately and manually
  - GPRESULT - Currently limited and without GUI support
  - PHYSICAL NETWORK – There is no way in AD to easily detect the status of the physical network

Computer Accounts Lab (1)

1. Creating a computer account
2. Disable/Enable a computer account
3. Manage a Computer via an Account
4. Deleting a computer account
Creating a computer account

1. Open “Active Directory Users and Computers”
2. Right-click on the Users folder then select New → Computer
3. Input computer name “GORDWKTestAccount”
4. Click next until finish
5. Check the User folder for the new object

Disable/Enable a computer account

1. Logon on to XP Client using the “test.2.user” user account
2. On the Glasgow DC server open “Active Directory Users and Computers”
3. Right-click on the Users folder then select All Tasks → Find…
4. Change the following values…
   1. Find: “Computers”
   2. In: “nwtraders”
   3. Role: “Any”
5. Click Find Now
6. Right-click on the xp computer account, then click Disable Account
7. Confirm by clicking YES
8. Right-click on the Bonn computer account, then click Enable Account

Disabling an account does not force a logoff!
Enabling an account does not force a logon!
Manage a Computer via an Account

1. Log on to XP Client Bonn using the “test.2.user” user account
2. On the Glascoe_DC server open “Active Directory Users and Computers”
3. Right-click on the Users folder then select All Tasks → Find…
4. Change the following values…
   1. Find: “Computers”
   2. In: “nwtraders”
   3. Role: “Any”
5. Click Find Now
6. Right-click on the Bonn computer account, then click Manage

Deleting a computer account

1. Open “Active Directory Users and Computers”
2. In the Users folder object list right-click on computer account “GORDWKTestAccount”
3. Select Delete
4. Click YES to confirm deletion
User Accounts Lab

1. Creating a user account
2. Enabling a user account
3. Finding a user account
4. Adding a “Member Of” link
5. Creating a User from a Template
6. Removing a “Member Of” link
7. Resetting a user account password
8. Deleting a user account

User Accounts Lab (1a)

Creating a user account

1. Open “Active Directory Users and Computers”
2. Right-click on the Users folder then select New → User
3. Input the following then click next:
   1. F.Name = “Test”
   2. M.Initial = “1”
   3. L.Name = “User”
   5. Select the “@nwtraders.msft” domain
4. Use “P@ssw0rd” for the password, then attempt to check all of the checkboxes.
   1. Observe the error messages from several combinations checked
   2. Leave only the bottom three checkboxes checked… Click next.
5. Review the summary… Click finish.
**User Accounts Lab (1b)**

**Creating a user account**

6. Verify in the User folder that the new user account has been created.
   - The refresh button is available to refresh the user folder's object list.

7. Attempt to login on the XP client using the new user account.
   - Username = "Test.1.User"
   - Password = "P@ssw0rd"
   - Domain = "NWTRADERS"

8. Logon fails because the account was optionally disabled when it was created.

**User Accounts Lab (2)**

**Enabling a user account**

1. Open “Active Directory Users and Computers”
2. Click on the Users folder
3. Right-click on user “Test.1.User”, then select All Tasks → Enable Account
4. Click OK to confirm
5. Reattempt logon to XP Client
   - Username = "Test.1.User"
   - Password = "P@ssw0rd"
   - Domain = "NWTRADERS"

Logon is successful this time because the account has been enabled.

*Don't forget to logout of Bonn when done.*
User Accounts Lab (3)

Finding a user account
1. Open “Active Directory Users and Computers”
2. Right-click on the Users folder then select All Task → Find…
3. Click on Find dropdownbox, then select “Users, Computers, and Groups”
4. Click on In dropdownbox, then select “Entire Directory”
5. Click on Find Now button
6. Type “a” in the Name textbox then click Find Now
7. Type “adm” in the Name textbox then click Find Now
8. Change the In dropdownbox to the “Users” folder, then in the Name textbox type “a” and click Find Now

User Accounts Lab (4)

Adding a “Member Of” link
1. Open “Active Directory Users and Computers”
2. Click on the Users folder
3. Right-click on user “Test.1.User”, then select Properties
4. Select the Member Of tab
5. Click the Add… button
6. In the object name textbox…
   1. Try clicking the examples link
   2. Type in the group named “domain admins”
   3. Click the Check Names button, then click OK
   5. If Check Names is unsuccessful, check the spelling, verify the object type, verify the object location or try using the advanced button to search for a group
7. Click OK until back at the user’s property box
8. Click APPLY, then OK
Creating a Users from a Template

1. Open “Active Directory Users and Computers”
2. Click on the Users folder
3. Right-click on user “Test.1.User”, then select All Tasks → Copy…
4. Input the following then click next:
   1. F.Name = “Test”
   2. M.Initial = “2”
   3. L.Name = “User”
   4. Username = “Test.2.User”
   5. Select the “@nwtraders.msft” domain
5. Use “P@ssw0rd” for the password
6. Review the summary… Click finish.

All user accounts created but not immediately needed or created by some form of automation should be disabled until needed. By using templates (or copy-and-paste), user accounts provide the benefit of automatically enrolling the new user into all of the group memberships that the master was a member of. Using templates to create users also produces the potential for creating users that are actively enabled but with unsecure standardized (or default) passwords. Be sure to disable the original user account before using as a template master.
User Accounts Lab (6)

Removing a “Member Of” link
1. Open “Active Directory Users and Computers”
2. Click on the Users folder
3. Right-click on user “Test.1.User”, then select Properties
4. Select the “Member Of” tab
5. Click on the “Domain Admins” group
6. Click the Remove button
7. On the popup dialog box, confirm the removal of group “domain admins” by clicking OK
8. At the user’s property box click APPLY, then OK

User Accounts Lab (7a)

Resetting a user account password
1. Open “Active Directory Users and Computers”
2. Click on the Users folder
3. Right-click on user “Test.1.User”, then select All Tasks ➔ Reset Password
4. Attempt to change the password to “temp1234”. Read the error message.
5. Attempt to change the password to “temp1234%”.
6. Logon to the XP Client as user “Test.1.User” with passwords:
   1. “P@ssw0rd”
   2. “temp1234”
   3. “temp1234%”

Did “P@ssw0rd” work? Why??
Would the command “GPUpdate” help??
Don’t forget to logout of Bonn when done.
User Accounts Lab (7b)

Resetting a user account password

6. Logon to the XP Client Bonn as user “Test.1.User” with passwords:

<table>
<thead>
<tr>
<th>Passwords</th>
<th>Successful?</th>
</tr>
</thead>
<tbody>
<tr>
<td>“P@ssw0rd”</td>
<td>Maybe</td>
</tr>
<tr>
<td>“temp1234”</td>
<td>NO</td>
</tr>
<tr>
<td>“temp1234%”</td>
<td>YES</td>
</tr>
</tbody>
</table>

7. While still logged on to Bonn…
   1. Change the password back to “P@ssw0rd”
   2. Use GPUPDATE at command prompt
   3. Logout of Bonn
   4. Attempt login again with all three passwords

*GPUpdate is optional… Else wait < 90 min.*

*Don’t forget to logout of Bonn when done.*

Domain group policies including password policies refresh automatically every 90 minutes for the computer accounts and for user accounts.

After a modification to a GPO or an account property, you can use the command-line tool **GPUpdate.exe** to refresh the GPOs and domain policies so that it takes effect immediately. GPUpdate replaces the Windows 2000 tool Secedit.exe.

To use GPUpdate go to Start  Run  type in “cmd”  OK. At command prompt type in gpupdate and press the enter key.
Deleting a user account

1. Open “Active Directory Users and Computers”
2. Click on the Users folder
3. Right-click on user “Test.1.User”, then select **Delete**
4. Click YES to confirm
5. Verify that user has been removed:
   1. Visually check the User folder object list
   2. Perform a search for the missing user account in the user Entire Forest
   3. Attempt logon to XP Client with the deleted account credentials
In a user account’s properties, there exist many options for implementing user account level policies. The most commonly used are Logon Hour restrictions, Account Expiration dates, Profile Paths, and client Remote Control via Active Directory.

**Logon Hours**
Is used to render a user’s account incapable of logon during certain hours of the day, if a user is already logged on, passed Logon Hour policies will not force the user to log off.

**Account Expiration**
Disables a user’s account after a specified date

**Profile Path**
Active Directory uses this user account policy option to determine where user’s account profiles will be saved. Profile paths may point to local disk space or the networked drives.

**Remote Control**
Active Directory administrators and delegated users have the capability to use Active Directory to gain remote controlled access to a user’s desktop. Remote control access can be set to request connection permission from the client user or can connect stealthily. Remote control can also be set to prevent the
administrator from interfering with the user’s session (known as observation mode).
User Account Login DoS Issues

- Incorrect usernames and passwords
- Account Lockout
  - Multiple incorrect logins
- Disabled User Account
- Logon hour restrictions
- Domain controller is not available
  - Too busy
  - DNS server is not available
  - Network connection lost
- Filter / Access Control List
- Client time settings
- UPN logon issues

DoS – Denial of Service

There are some common user account issues that occur which could lead to a denial of user access.

Incorrect usernames and passwords are the most common source of service denial to users. If a user cannot gain access, the first step should be to reset the password and verify that the user is using the correct username / domain-name combination.

Account Lockouts could occur due to several reasons. A user could accidentally input an incorrect password too many times (typically more than three attempts), a user's account might be locked by an administrator for administrative reasons, or a user may be attempting to log into a computer that has been disabled.

A Disable User Account could cause a denial of service. Only enabled accounts on enabled computers will be allowed authentication. By default user accounts created from the Active Directory Users and Computers snap-in will be automatically enabled. New computer accounts will also default to enabled, even if the computer account was implicitly created.
Logon Hour Restrictions could cause a user to be denied access when attempting to logon to an Active Directory domain. If a user is currently logged in when a logon hour restriction starts, that user will not be forced to log off. However, the user will not be allowed to logon again until the time logon hour’s restriction expires.

Domain Controller is not available – A domain controller can become overburdened with too many authentication requests. This could be in part due to a poor network connection, too many objects stored in the LDAP database, or too many requests for the hardware processes to support. A domain controller can also become unavailable if it losses DNS service.

Filter / Access Control List
An individual user, file, folder, object, or group of objects might have been edited under their security permissions to deny access to the object being denied access.

Client time settings
Client computers must remain in time synchronization with the controlling domain controller. If access is lost due to a time synchronization incompatibility, the client’s computer must change its clock by logging on as a local administrator.

UPN logon issues
Forest level credential authentication will not be successful when there exists a forest wide issue. This includes, but is not limited to, lost access DNS services, network connectivity, global catalog server, transitive trusts, and/or cross-site restrictions.
User Principle Names

- **User principal name (UPN)**
  - Optionally assigned to user accounts
  - Composed of a prefix and suffix
  - “@” symbol separates Prefix from Suffix
  - Example: “objectname@domain.com”

- **UPN Prefix**
  - Assigned at will by the Domain Admin
  - Normally mirrors a user’s email username
  - Example: “Joe Plumber” is “joe.plumber3@us.army.mil” and has UPN “joe.plumber3”

- **UPN Suffix**
  - Same as name of local domain
  - Example: “dc=us, dc=army, dc=mil” ---> “us.army.mil”
UPN Logon Issues

- UPN is multiple domain
- If users cannot login but they can login using the dropdown,
  - The global catalog is down making the forest unavailable
  - User should still be able to login to own domain.
  - Whenever attempting to login to non-owned domain, the GC is needed

The second part of the UPN, the UPN suffix, identifies the domain in which the user account is located. This UPN suffix can be the DNS domain name, the DNS name of any domain in the forest, or it can be an alternative name created by an administrator and used just for log on purposes. This alternative UPN suffix does not need to be a valid DNS name.

In Active Directory, the default UPN suffix is the DNS name of the domain in which user account created. In most cases, this is the domain name registered as the enterprise domain on the Internet. Using alternative domain names as the UPN suffix can provide additional logon security and simplify the names used to log on to another domain in the forest.

For example, if your organization uses a deep domain tree, organized by department and region, domain names can get quite long. The default user UPN for a user in that domain might be sales.westcoast.microsoft.com. The logon name for a user in that domain would be user@sales.westcoast.microsoft.com. Creating a UPN suffix of "microsoft" would allow that same user to log on using the much simpler logon name of user@microsoft. For more information about user accounts, see User and computer accounts and Object names.
Cached Credentials

• Hidden in the SAM hive of the registry

• Account restrictions are also cached and remain in effect

• Records username and passwords (encrypted) of every previous user logged into PC

• Provides local access to PC
  – No access to Active Directory controlled network resources (shared folders, printers, remote access, etc.)

• Can be turned off or limited
  – “Turn-off” could cause ZERO trouble-shooting access

When a local client computer cannot gain authentication with a domain controllers, typically due to network connectivity, the client will rely on its cached of saved authentication credentials. Abuse of this ability could pose a security risk; however, disabling the feature could make troubleshooting and emergency access impossible.
Passwords

• Minimum & maximum history limits
  • Work together (with reset counter) to prevent password cycling
• Reset counter
  • Typical remembers last 40 passwords per user
• Character length
  • Typically requires a length between 8 and 12 characters
• Complexity requirements (3 out of 4)
  • At least one Lower case character
  • At least one Upper case character
  • At least one Numeric character
  • At least one Non-alpha (symbol) character

It is common for group policies concerning password policy to require a minimum password length, a complex string of characters, a minimum amount of time that password must be kept without administrative intervention (typically 24 hours). In addition a maximum length of time that a password may be kept (commonly referred to as password expiration), and a high threshold reset counter that defines how many passwords the system will remember and prevent the user from reusing.

Any of the user account password policies can be adjusted as needed by a domain admin. Password and account lockout policies should be set at the domain group policy level!
Account Lockout

• Account Lockout Policy concerns…
  – Lockout Duration
  – Failed Login Threshold
  – Password Reset counter

Account lockout policies work well with password policies. A normal account
lockout policy will dictate that an account will become locked after three
erroneous username/password authentication attempts. It will also dictate that
the account will not automatically reset itself until 1 hour has passed or possibly
not at all.

Any of the account lockout policies can be adjusted as needed by a domain
admin. **Password and account lockout policies should be set at the domain
group policy level!**
AGDLP - User/Group Organization (1)

AGDLP = A → G  DL  P
“A Great Domain Layout Plan”

AGDLP is a suggested best practice for the design, creation, organization, and allocation of privileges to Active Directory users. The grand idea is to start with your users, work your way across, and then give them the permissions that they need.

**Order of assignment**
A = Accounts
G = Global Groups
DL = Domain Local Groups
P = Permissions

**Accounts**
It is a good practice to create a user account for each user present in a domain. Sharing account usernames and passwords, even for system administrators can be complicated to monitor and pose serious security risks. In Active Directory, it is not necessary for admin users to share accounts.

Active Directory allows any number of individual user accounts be added to the administrators groups and therefore have the same rights and privileges. Users
can be granted temporary accounts separate from their permanent accounts strictly for granting elevated or complex user roles. These separate accounts, like all other user accounts, can be limited to logon hours, set to terminate at a specific date, and can even be limited to access only certain computers.

**Global Groups**
User accounts are best added to global groups instead of being directly added to a local group, organizational unit, or directly assigned permission. Adding users to groups (global or domain local) prior to assigning permissions allows administrators to more accurately predict how policies will conflict on a user account. Adding users to groups also gives a visually comprehensible view of a user and his/her roles.

When a network scales-up, object organization directly limits the ability of administrators to manage the directory properly. The benefit of adding users to global groups first, before adding to a domain local group is that global groups are available to all trusted domains in the forest. Therefore, a single global group can support the needs of a specific task for an entire forest.

A universal group can be used in most cases in place of a global group if the forest is at functional level 2.

**Domain Local Groups**
Domain Local groups are necessary components for assigning privileges at the domain level. Domain Local groups can support individual user memberships but cannot support users from other domains. The best use of domain local groups is to be a central control point of local domain resources, where global groups become a “Member” of the domain local group and the domain local group is in turn added to a single organization unit which supports very specific rights.

**Permissions**
This portion of the acronym represents the roles, rights, and restrictions enforced on an Active Directory controlled network resource or an Active Directory object. Just to name a few objects in Active Directory that can be assigned permissions, review the following: shared files, shared folders, network drives, terminal services, profile folders, user accounts, computer accounts, printer accounts, domain local groups, global group, universal groups, group policy objects (GPOs), and the domain itself.
1. A user account needs access to a shared folder in another domain. It is added to a global group.

2. The global group is then added to a domain local group.

3. The domain local group is given permissions to the shared folder.

4. The user from an outside domain gains access to the files in the folder.
With the help of an instructor perform the following tasks using domain local security groups:

1. Create two new security groups:
   1. “test group 1”
   2. “test group 2”
2. Create a new user account of with a name of your choice.
3. Add the user account and “test group 2” to the Member tab “test group 1”.
4. Remove “test group 1” from the “Member Of” tab of the user account.
5. Add “test group 2” to the “Member Of” tab of the user account.
6. Find the two domain local groups by using the AllTasks → Find… action.
7. In the found results box, delete both testing groups
8. Review the users “Member Of” property tab to confirm removal

Domain Local groups may contain individual users, other domain local groups, global groups, and universal groups.

**“Member” Tab**
An object included in the “Members” tab of a group object specifies that the listed objects are containers for the selected object. A user account can never act as a container for another account or for a group.

**“Member Of” tab**
An object included in the “Members Of” tab of a group object specifies that the listed objects are a part of the group.
**Global Groups Lab**

With the help of an instructor perform the following tasks using global security groups:

1. Create two new security groups:
   1. “test group 1”
   2. “test group 2”
2. Create a new user account of with a name of your choice.
3. Add the user account and “test group 2” to the Member tab “test group 1”.
4. Remove “test group 1” from the “Member Of” tab of the user account.
5. Add “test group 2” to the “Member Of” tab of the user account.
6. Find the two global groups by using the AllTasks ➔ Find… action.
7. In the found results box, delete both testing groups.
8. Review the users “Member Of” property tab to confirm removal.

Global groups may contain individual users and other global groups. Global groups cannot contain domain local groups or universal groups.
Raise the Functional Level (1)

Raising the domain Functional Level

• Right-click on the domain controller icon “nwtrader.msft”
• Navigate to AllTasks → Raise Domain Functional Level…
• Select functional level “Windows Server 2003”
• Click the RAISE button to confirm

Raising the functional level is an irreversible change to the entire Active Directory structure.
Raise the Functional Level (2)

Raising the Forest Functional Level
- Administrative tools
- Domains and trust
- Right Click DC
- Navigate to AllTasks → Raise forest Functional Level…
- Select functional level “Windows Server 2003 Native”
- Click the RAISE button to confirm
Universal Groups Lab

With the help of an instructor perform the following tasks using universal security groups:

1. Create two new security groups:
   1. “test group 1”
   2. “test group 2”
2. Create a new user account of with a name of your choice.
3. Add the user account and “test group 2” to the Member tab “test group 1”.
4. Remove “test group 1” from the “Member Of” tab of the user account.
5. Add “test group 2” to the “Member Of” tab of the user account.
6. Find a the two universal groups by using the AllTasks → Find… action.
7. In the found results box, delete both testing groups.
8. Review the users “Member Of” property tab to confirm removal

Universal groups can contain individual users, other universal groups, and global groups. Universal groups cannot domain local groups.
Renaming Groups Lab

1. Create 2 of each of the security groups:
   1. Domain Local Groups
      1. “test domain local group 1”
      2. “test domain local group 2”
   2. Global Groups
      1. “test global group 1”
      2. “test global group 2”
   3. Universal Groups
      1. “test universal group 1”
      2. “test universal group 2”

2. Attempt to rename one of each a group type:
   1. Right-clicking on the object
   2. Select rename
   3. Change both the group name and the Pre-Windows 2000 group name to “temp group name change”
   4. Rename the group back to its purposeful name

Globally Unique Identifier

- Assigned to every object in a domain
- Carries information about original domain
- Unique and Unchanging
- 128-bit string
- Used by AD for search and replication
Creating Accounts without a GUI

- Command Line (Create, Modify, Delete Accounts)
- Batch Files (Create, Modify, Delete Accounts)
- CSVDE Command Line Tool (Create Accounts)
- LDIFDE Command Line Tool (Create, Modify, Delete Accounts)
- Windows Script Host (Create, Modify, Delete Accounts)

**Directory Service Tools** assist in the creation, modification, and deletion of Active Directory objects. Any Active Directory object that can be created using a GUI can also be created using a script or command line tool. Directory Service command line commands include:

- **DSADD** – Directory Service Add is a command line action term used to create an Active Directory object and many of its defining attributes. For example, a new user could be simultaneously created by using DSADD and set his/her personal information, group membership, and password. For more examples of the available attributes to be set go to command prompt on a domain controller then type “DSADD /?” DSADD may be used to add users, computers, contacts, groups, OUs, and quota specifications to a directory partition. A quota specification determines the maximum number of directory objects a given security principal can own in a specified directory partition.

- **DSMOD** - Directory Service Modify is used for modifying existing objects in the directory. The DSMOD command supports piping of input to allow you to pipe results from the DSQEURY commands as input to the DSMOD command. DSMOD may be used to modify users, computers, contacts,
groups, OUs, quota specifications to a directory partition, and domain controller servers.

- **DSRM** – Directory Service Remove is used for deleting directory objects
- **DSGET** – Directory Service Get is used for displaying directory objects
- **DSMOVE** – Directory Service Move is used for moving directory objects
- **DSQUERY** – Directory Service Query is used for finding directory objects matching search criteria.

The **CSVDE** or **Comma Separated Value - Delimited** text file is a comma-delimited text file that may be used to create multiple user accounts and other objects to Active Directory. CSVDE format cannot be used to modify or delete objects. Additionally, before the file can be properly used, the formatting (i.e. comma separations) should be reviewed to prevent errors.

Text editors such as Microsoft Excel and Microsoft Word make the CSVDE file creation process easier and help mitigate oversight accidents often caused when using manual typing or cut-and-paste methods within Notepad. In CSVDE scripts, all attributes related to each object are listed on the same line and separated by commas. Each new object, however, must be placed on a new line. When adding new objects using CSVDE the input file must contain the following attributes for each object created:

1. A path to the user account in Active Directory, the object type, a user logon name, a UPN (User Principle Name), and the enable/disable state of the account.
2. If a UPN is not provided, the account is disabled by default. Personal information (such as telephone numbers, first name, last name, Address, email address) for the account can also be included at option.
3. Passwords are not allowed. Bulk inserts leave the password field blank for user accounts. Good practice requires that all bulk insert accounts be created in the disabled state for security purposes.

The **LDIFDE** Tool is a command-line tool that uses a line-separated value format to add, modify, and delete users. The key differences between LDIFDE and CSVDE are that LDIFDE requires that all new objects and each attribute related to that object be placed on separate lines and that LDIFDE allows the modification and deletion of Active Directory objects. In a LDIFDE file, any line that begins with a pound-sign (“#”) is treated as a comment and is ignored when the script is loaded. As with all bulk inserts, newly created user accounts should be set to the disabled state until they are needed.

**Windows Script Host** is scripts that use Active Directory Service Interfaces (ADSI) to add, modify, and delete Active Directory Accounts. The most common use for this scripting method is for changing the values of attributes for multiple Active Directory objects or when the criteria for selecting the affected objects is complex. Windows Script Host establishes a new Active Directory object by declaring a new local variable that shares the same class as that object type. The ADSI interpreter then interprets the local variable as a new object and allows
the assignment of attributes and other objects to that variable by referencing the variable’s name. This adds a layer of complexity but also great flexibility to the bulk account management process. VBScript may also be used.
Moving AD Objects

“MOVETREE” command

/s <server> - Identifies the source domain controller (DC) to use.
/d <server> - Identifies the destination DC to use.
/sdn <source DN> - Identifies the source DN of the tree to be moved.
/ddn <dest. DN> - Identifies the destination DN.
/check - Instructs ‘movetree’ to perform a test to determine whether it can move the tree without actually moving it.

GUI methods such as Drag-and-Drop and Cut-and-Paste may be used to move objects within a domain. Nevertheless, sometimes objects need to be moved to other domains. The “movetree” command is available from the command line prompt to move Active Directory tree structures and all objects contained therein. MOVETREE command is used when the move required takes an entire tree out of one domain and into another.

When using the MOVETREE tool:
- The destination domain must be in Win2K Native mode or later.
- The source and destination DNs should be written in lower-case.
- Computer Accounts may be moved, but by default, they will not be active in the new domain. Use the NETDOM command instead.


Each object has a unique Global Unique Identifier (GUID), which does not change during the move, so moving objects generally has no effect on their behavior. In some cases, however, there are implications for moving objects. If you move a user who is a member of the global group Support in the Software domain to the Hardware domain, that user's account takes on a new Security ID (SID) in the Hardware domain, the user's new home. Resources that were
previously available to the user through that group membership are no longer available because of the SID change.

There are several ways to remedy the situation, including adding the new SID to the resource access control lists (ACLs), moving the group, and creating a parallel group in the new domain, among others.
TAB

Insert Tab # 6 Here
Organizational Units
The Organizational Unit (OU)

- Organization Units
  - Logical container
  - Stores objects
  - Organizes objects
  - Operates in a single domain

- Logical Separation
  - Department
  - Job Functions
  - Geographical Location
Common uses for an OU

• Organization Units (OU)
  – Application of group policy
  – Delegation of functionality

• Group Policy
  – Roles or tasks assigned to an entire group of users or computers
  – Groups Policies must be inside an OU

• Common Functionality
  – Organizes Active Directory objects
    • Common Objects: Users, Computers, Groups, Printers, & “nested” OU’s
  – Provides logical organization
  – Desktop restrictions
  – Software deployment

A common purpose of using an OU is its ability to organize Active Directory objects and logically separate object groups. Logical separation is important in AD. It allows group policies and permissions to have no effect on other group policies and permissions outside of the OU.

There are a few exceptions to the rule that OUs separates objects. One of the primary exceptions is the assignment of permissions or functional tasks directly to the user or computer object. The assignment of rules to users or computers at a multiple levels (i.e. account level, group level, resource level) could potentially cause an overlapping of two or more conflicting rules. For example, one GPO assigned to a user account might not allow Windows Internet Explorer to be launched…EVER!
When is an OU Appropriate?

- Unified Administrative control
- Very VERY similar GPO requirements
- Legal or Business requirements

Organizational Units are appropriate when there is an administrative need for grouping and/or segregating one or more sets of Active Directory accounts according to group policies.

Organizational Units might also be created specifically to provide a group policy contradict (or to override higher GPO rules). All objects contained in an OU are subject to all GPOs in that OU, and all other associated OUs.

Organizational Units simplify the administration of related accounts but increases the likelihood of administrative difficulty or policy conflict. A set of objects may be governed by more than one separate OU, where each of the OUs has its own list of GPOs and deputy administrators.

Placing account objects in an OU simply to clean up the Active Directory workspace such as creating a 'misc.' OU is not a good practice.
Nesting OU

- Nested OUs inherit the GPOs from their parent OU
- Nesting simplifies organization
- Nesting complications management and troubleshooting
Inheritance in Active Directory

OU Inheritance flows from Top to Bottom
- Parent OUs give inheritance to child OUs
- An OU can inherit from multiple levels of OUs at once
- It is possible for an OUs inheritance to conflict
- Blocked policies do not bypass a parent OU

Inheritance can only be blocked completely
- Not like Novell’s IRF
- Any OU can completely block inheritance from all parent OUs
- Inheritance blocking is all or none
- A child OU, who blocks inheritance but also serves as a parent to another OU will still pass on only its own inheritance to its child.

In the chart above, there are seven Organizational Units present in the domain. Most are nested, two OUs are not (“OU 1.0.0” and “OU 2.0.0”).

Each of the OUs has GPOs that define their policies. Some of the OUs are in a position to pass down their policies. Some of the OUs are in a position to inherit policies from their parent OU.

Two of the OUs are both parent OU and child OU. Two of the OUs are blocking inheritance from their parent OU, whenever a child OU is blocking inheritance, that OU will not receive policies from its parent OU. It also will not pass down the blocked policies to its child OU. To put it simply, “OU 1.b.1” will not receive the policies from “OU 1.0.0” because “OU 1.b.0” has blocked inheritance.

Blocked policies do not bypass a parent OU. Additionally the policies from “OU 1.a.0” will be blocked to “OU 1.a.1” but will not be blocked by “OU 1.a.2”. This means that “OU 1.a.1” will not receive inherited policies from any other OU. “OU 1.a.2” will inherit from “OU 1.a.0” and “OU 1.0.0”. The acceptance or blocking of inheritance by either “OU 1.a.1” or “OU 1.a.2” will not have any effect on one another since both are sibling organizational units and therefore at the same hierarchical level. Likewise, “OU 2.0.0” is unaffected by any of the policies occurring in any of the OUs under “OU 1.0.0”.

Organizational Unit Labs (1)

1. Creating an OU
2. Nesting an OU
3. Moving an OU
4. Rename an OU
5. Delegation of Control over an OU
6. Deleting an OU

Organizational Unit Labs (2)

Creating an OU
1. Open “Active Directory Users and Computers”
2. Right-click on the domain icon ‘nwtraders’, then select New → Organizational Unit
3. Name the new OU “ouTest1”
4. Click OK to confirm creation
5. Create two more OUs with the following names:
   1) 2nd OU Name = “ouTest2”
   2) 3rd OU Name = “ouTest3”
Creating an OU

6. Create the following users under in “ouTest1”:

<table>
<thead>
<tr>
<th>FirstName</th>
<th>LastName</th>
<th>Username</th>
<th>Password</th>
</tr>
</thead>
<tbody>
<tr>
<td>TestUser</td>
<td>Admin</td>
<td><a href="mailto:TestUser.Admin@nwtraders.msft">TestUser.Admin@nwtraders.msft</a></td>
<td>P@ssw0rd</td>
</tr>
<tr>
<td>TestUser</td>
<td>Helpdesk</td>
<td><a href="mailto:TestUser.Helpdesk@nwtraders.msft">TestUser.Helpdesk@nwtraders.msft</a></td>
<td>P@ssw0rd</td>
</tr>
<tr>
<td>TestUser</td>
<td>CommonUser</td>
<td><a href="mailto:TestUser.CommonUser@nwtraders.msft">TestUser.CommonUser@nwtraders.msft</a></td>
<td>P@ssw0rd</td>
</tr>
<tr>
<td>TestUser</td>
<td>Contractor</td>
<td><a href="mailto:TestUser.Contractor@nwtraders.msft">TestUser.Contractor@nwtraders.msft</a></td>
<td>P@ssw0rd</td>
</tr>
</tbody>
</table>

Nesting an OU

1. Open “Active Directory Users and Computers”
2. Locate the OU named “ouTest1”
3. Right-click on the OU’s icon (“ouTest1”), then select New → Organizational Unit
4. Name the new OU “ouTest4”
5. Click OK to confirm creation
Moving an OU
1. Open “Active Directory Users and Computers”
2. Locate the OU named “ouTest3”
3. Right-click on the OU’s icon (“ouTest3”), then select All Tasks → Move…
4. Expand the following branches:
   1) “ouTest1”
   2) “ouTest4”
5. Click “ouTest4” to highlight
6. Click OK to confirm move

Rename an OU
1. Open “Active Directory Users and Computers”
2. Right-click on the OU named “ouTest1”:
   1) select Rename
   2) Type “ouAllUsers” in the highlighted section
3. Rename the following OUs:
   1) “ouTest2” = “ouSpecialUsers”
   2) “ouTest3” = “ouLimitedUsers”
   3) “ouTest4” = “ouAdvancedUsers”
Delegation of Control over an OU
1. Open “Active Directory Users and Computers”
2. Right-click on “ouAdvancedUsers”, then select Delegate Control…
3. Follow the wizard while filling in the following information:
   1) Selected Users & Groups → Add → “TestUser Helpdesk” → Check Names → OK → OK → Next
   2) Delegate common task:
      a. Check the box for Create, delete, and manage users…
      b. Check the box for Reset password…
      c. Check the box for Read Users info…
3) Click next until the wizard is finished
4. Repeat the delegation for “TestUser Admin” in “ouAdvancedUsers”
   1) Grant the first 8 permissions
   2) The last permission is “Generate Resultant Set of Policy (Logging)”

Deleting an OU
1. Open “Active Directory Users and Computers”
2. Right-click on “ouSpecialUsers”, then select Delete
3. Name the new OU “ouTest1”
4. Click YES to confirm creation
This page intentionally left blank.
TAB

Insert Tab # 7 Here
Creating & Managing Group Policies
Group Policies

- Configures desktop settings
- Controls security settings
- Run scripts
- Redirects folders
- Automates software distribution and maintenance
- Also sets:
  - Password & account policies
  - Folder redirection
What are GPOs?

- **Group Policy Objects**

- **Application Levels:**
  - Sites
  - Domains
  - Organizational Units

- **A GPO’s possible settings:**
  - Computer configurations
  - Windows settings
  - Security settings

Group Policy Objects (GPO) is an object in Active Directory that can be assigned to a site, domain, or organizational unit, but applies to all inherited objects within the object where it has been assigned. GPOs are the inheritable rights and restrictions placed upon objects within Active Directory. It is recommended that you minimize the number of Group Policy objects that apply to users and computers. Multiple Group Policy objects increase computer startup and user logon time. Each computer and user Group Policy object must be downloaded to a computer during startup and to user profiles at user logon time. Additionally, multiple Group Policy objects can create policy conflicts that are difficult to troubleshoot.

A **GPO linked to a site** applies to all domains at the site and help to separate settings. This application of GPOs can be assigned using Active Directory Sites and Services. Applying policy settings to a site, such as the default-first-site, is a possible alternative to setting a group policy setting at the forest root domain since the policies will be inherited by all domains in the hierarchy. Placing policy settings at the site level can cause potential troubleshooting issues since an admin user could have administrative access to the forest root domain, but not the site settings.

A **GPO linked to a domain** applies to all users and computers in the domain. The GPO is inherited by all users and computers in any child object, including
organizational units and generic Active Directory containers in the domain. GPOs can be applied to local domains as well as other domains in the forest. Whenever both exist the local domains policies are applied first, followed by the guest domain. Domains in the same tree inherit the group policies of their parent domain (the tree root).

A **GPO linked to an organizational unit (OU)** applies to all users and computers within the organizational unit and its child container objects only. Its policies will have no effect on other objects at the same hierarchical level or at a higher hierarchical level such as a parent OU or the local domain.
What do GPOs effect?

• User Logon policies
• Computer Authentication policies
• Application deployment policies
• File deployment policies
• Script policies
• Software policies
• Security policies

User Logon policies
These policies dictate the rights and restrictions that pertain specifically to an individual user on the network.

Security policies
These policies allow an administrator to restrict user access to files and folders, configure how many failed login attempts will lock an account, and control user rights.

Computer Authentication policies
These policies dictate the rights and restrictions that pertain specifically to an individual computer or computing device on the network.

Software policies
Administrators can use software policies to configure most of the settings globally in user profiles, such as desktop settings, Start menu options, and applications.

Script policies
Using a script policy, an administrator can specify scripts that should run at specific times, such as login/logout or system startup/shutdown.
Application deployment policies
These policies assign or publish applications to users or computers, and affect the applications that users access on the network.

File deployment policies
These policies allow an administrator to place files in special folders on the user's computer, such as the desktop or My Documents areas.
Key Features of GPOs

2. List Order                    7. Effect of moving an OU
3. Linked GPOs                   8. “Not Configured” settings
4. Rename a GPO                  9. Block Policy setting
5. Deletion                      10. No Override setting

Creation
A GPO can be created under a domain’s properties, a site’s properties, or an organizational unit’s properties. When a GPO is created, all of its settings are in the default or non-configured position. A GPO can be created by going to a domain, site, or an OU’s properties settings, navigating to the Group Policy tab, and then clicking the NEW button or the ADD button. GPOs can be completely new or linked to an existing GPO that resides in a separate container. The creation of a GPO could cause a policy conflict since group policies are inheritable.

List Order
The order, which the GPOs are listed in the Group Policy tab, is important. The listed order of GPO specifies the priority where the highest priority is given to GPOs at the top of the list. The GPOs priority is a list of when the group policies will be applied. The highest priority GPO will be applied last in order to override conflicts with other group policies.

Linked GPOs
A GPO may be added to a domain, site, or OU by linking to an existing GPO. The linked GPO may exist in the same object container (domain, site, OU), or in a separate object container. A linked GPO takes on all of the settings as its original GPO. The original GPO or any linked GPO can change the configuration.
settings of the GPO. Linked GPOs can be deleted without removing all the other linked GPOs and without deleting the original GPO. Original GPOs can also be deleted without removing all the other linked GPOs. However, if either the original GPO or any of the linked GPOs are deleted, all will be deleted.

**Renamed GPOs**
A GPO may be renamed without effecting its configurations. More than one GPO can share the same name. This is OK because Active Directory tracks GPOs by assigning each GPO a unique identifier that will never be reused. A named changed to a linked GPO will change the names of all other linked GPOs.

**Deletion**
When a GPO is deleted it does not affect the other GPO (if any) present under the container object’s group policy list. When deleting a GPO, two options are provided: “Remove the link form the list” and “Remove the link and delete the Group Policy Object permanently”.

1. If a GPO is removed using the first option, it is only removed from the list. If it is not linked to any other GPO containers, then it has also been permanently deleted. Conversely, if the GPO has links that exist, those links and the configuration settings will remain.

2. The second option “Removes the link and deletes the Group Policy Object permanently”, it will remove the original GPO, and any existing linked GPOs, and permanently removes the configurations and their effect on all other objects with Active Directory. A deletion cannot be un-done. Creating a new GPO to replace a deleted GPO is still a new GPO and will not have any of the previous GPOs configuration settings.

**Enable / Disable GPO**
An alternative to deleting an unwanted GPO is using the enable/disable feature. A GPO can be disabled, causing its policies to be ignored, but leaving its configurations, priority, and links intact.

**Effect of moving an OU**
Moving an OU can cause GPO conflicts and lead to troubleshooting challenges, especially when the OUs are nested. When an OU is moved, the GPOs that it contains are not effected. Since Organizational Units and the GPO that they contain are inheritable policies, moving an OU to another hierarchy could cause a dramatic effect in the way that accounts respond to group policy.

**Non-Configured Settings**
All GPO settings not expressly set are by default not configured and therefore have no effect on the hierarchy of group policies to be passed down to lower account objects.

**Block Policy Setting**
The “Block Policy” setting allows a qualified object container (domain, site, organizational unit) to deny the application of group policies passed down by
previous policy objects (GPOs). “Block Policy” is an all or none setting. When active, “Block Policy” blocks all policies from all inheritable objects. If it is desired to block only certain policies, there are many administrative options such as:

- The container object could be moved from the scope of the undesired policy. WARNING - this solution might remove wanted policies.
- “Block Policy” could be used in conjunction with “No Override”. Read below for an explanation of “No Override”.
- An intentionally conflicting GPO could be created to over-rule the unwanted policies.
- The logical structure of Active Directory could be redesigned to remove the need for avoiding the unwanted policies. This option is mostly like used when the Active Directory structure and/or hierarchy has scaled well beyond the original specifications. When troubleshooting becomes grossly difficult, there is good cause for considering a redesign.

**No Override Setting**
The “No Override” setting is used to prevent the blocking of group policies by lower priority GPOs or by GPO in an inheritance position. “No Override” applies individually to marked GPOs.

“Block Policy” and “No Override” should be used sparingly because of the high potential for complicated GPO conflicts.
GPO Order of Application (1)

- GPOs are applied in the following order
  1. Local Computer
  2. Site
  3. Domain
  4. Organizational Unit
     a) Parent OU
     b) Child OU

GPOs are applied in the following order of the Active Directory hierarchy:
- Local Domain
- Site
- Domain
- Organizational Unit
  - Parent OU
  - Child OU
Within each level of the hierarchy, GPOs are applied according their priority number (list order). The list order dictates which GPO in a given container will be applied first, second, third, etc. The GPO with the lowest priority is lowest on the list and will be applied first. The GPO with the highest priority is highest on the list and will be applied last. Wherever conflict exists between similar policies of two or more GPOs, the last applied GPO wins.
GPO Order of Application (3)

Group Policy Inheritance

Best options of conflict resolution

1. Application Order
2. Priority Rank (GPO List Order)
3. Computers over Users

Last Resort of conflict resolution

4. Block policy (modifier)
5. No override (modifier)
6. Filtering (modifier)
7. Loopback Processing (modifier)

Using these means of policy resolution makes “good” sense.

Over use of these means of policy resolution suggest the need for a directory structure redesign!

Conflicts that occur within a single GPO are also possible. Active Directory uses a seven tier ranking policy to determine the winning policy to apply.

Group Policy Inheritance

Group Policy inheritance is concerned with how GPOs negotiate which policies will be enacted. The first three orders of Group Policy inheritance mitigate nearly all errors. When a conflict beyond the three orders occurs, there are four additional directives that instruct Active Directories how to react. Careful design of directory objects will also significantly lower the occurrence of unintentional results due to conflicting group policies. Group policies conflicts can exist among competing GPOs. They can also conflict among competing policies within a single GPO. For example, what might happen if a user account was assigned a policy that permitted access to any computer on the domain, but a specific group of computer accounts was assigned a policy that expressly denied access to all users, except enterprise administrators?

Active Directory eliminates most policy conflicts with the following three rules…

1. Application Order
   Group policies will be applied first from the local domain, followed by the site’s polices, then continue to the parent domains (starting with the senior
most domain and working down the hierarchy towards the present domain), and finally the organizational units.

2. **Priority Rank (GPO List Order)**
   At each hierarchical level, group policies objects are listed in a priority order. The GPO with a lower priority might possibly be overruled by a following GPO with a higher priority. The higher the priority, the longer the GPO waits to be applied.

3. **Computers over Users**
   When conflicts exist in a given GPO, the portion of the conflict that is scheduled for application to the computer will always win against the portion designed for the user. The primary purpose for computer policies winning against user policies is to prevent user policies from attempting an enforcement that the computer cannot perform.

When full conflict resolution cannot be achieved by the first three rules, Active Directory enacts the next four rules:

1. **Block policy (modifier)**
   This modifier of group policy inheritance restricts the ability of a set of policies to be passed on by deny all inheritable policies from previous GPOs.

2. **No override (modifier)**
   This modifier of group policy inheritance overrides the ability of “Block Policy” to deny an individual policy to be passed down by forcing all lower object containers and related GPOs to accept the marked policy.

3. **Filtering (modifier)**
   This modifier of group policy inheritance restricts or enables a custom set of policies by individually specifying which policies will be applied and which will not. Filtering is a “permissions trick”.

4. **Loopback Processing (modifier)**
   Like the “nuclear option” setting the loopback option causes the user configuration settings in GPOs that apply to the computer to be applied to every user logging on to that computer, instead of (replace mode) or in addition to (merge mode) the User Configuration settings of the user. This allows you to ensure that a consistent set of policies is applied to any user logging on to a particular computer, regardless of their location in Active Directory. Loopback is an advanced Group Policy setting that is useful on computers in certain closely managed environments, such as servers, kiosks, laboratories, classrooms, and reception areas. Loopback only works for computers that are not joined to a workgroup and only when both the user account and the computer account are in a Windows 2000 or later domain. Loopback is controlled by the “User Group Policy
loopback processing mode” setting located under Group Policy in Group Policy Object Editor (GPMC).

In any of the seven rules, each rule may act to eliminate a conflict, meaning that rules can act in combination. The first rule that removes the final outstanding conflict ends the conflict resolution process.

When using **Loopback in Replace Mode**, the GPO list for the user is replaced in its entirety by the GPO list that is already obtained for the computer (obtained during computer startup). The user’s configuration settings from this list are then applied to the user.

When using **Loopback in Merge Mode**, the list of GPOs for the user is obtained, as normal, but then the list of GPOs for the computer (obtained during computer startup) is appended to this list. Because the computer's GPOs are processed after the user's GPOs, they have precedence if any of the settings conflict.

The result is that the Loopback process ignores any attempts at blocking or shifting group policies and forces a set of preconfigured policies into a user object’s policies. **Replace mode** will completely ignore a user object’s previously inherited policies and replace the user object’s policies with the loopback policies specific to a given computer. **Merge mode** will attempt to keep a user object’s previously inherited policies and replace only the user object’s policies that conflict with the loopback policies specific to a the same given computer.
• **Group Policy Result (GResult.exe)**
  – Discover all group memberships and nested groups
  – Shows which policies are in effect
  – Useful in troubleshooting permissions policies
Group Policy Refresh Rates

- Update Interval
  - 90 minutes (default)
  - Default time can be changed to a minimum of zero seconds

- Triggering Events
  - Computer startup
  - User logon / logoff
  - An application requests a refresh by way of the RefreshPolicy() API
  - The user requests an immediate refresh
  - One of the Group Policy refresh interval policies is enabled, and the interval has transpired (90 min. by default)

- Manual update → “GPUpdate.exe”

The minimum setting of the Group Policy Refresh Interval is zero minutes, which actually takes about seven seconds between each request for a refresh. However, very short settings should not be used in a production environment and then only for testing.

Two exceptions to the rules are Folder redirection and Software Installation. These refreshes may only take place during startup or shutdown, not while a user’s session is taking place. To solve this, either request that a user logoff, then logon again, or an administrator can force a logoff.
A GPO contains several sections under computer configurations and user configurations which in turn contain additional sections and eventually policy settings. One policy section that is present in both the computer configuration and user configuration section is the **Administrative Templates** section.

Administrative Templates are a large repository of registry-based changes that can be found in any GPO. There are over 1300 individual settings available on Windows 2000, Windows XP, and Windows Server 2003 operating systems.

By using the Administrative Template sections of the GPO, you can deploy modifications to machine (called HKEY_LOCAL_MACHINE in the registry) and user (called HKEY_CURRENT_USER in the registry) portions of the Registry of computers that are influenced by the GPO.

The Administrative Templates are saved in text files with the extension .ADM. These text files are used to create the Administrative Templates portion of the user interface for the GPO Editor.

It is here, in the Administrative Templates section, that the true power of Active Directory takes form.
Group Policy Labs (1)

1. Creating a GPO
2. Linking a GPO
3. Changing a GPO’s Priority
4. Administrative Templates

Group Policy Labs (2)

Creating a GPO
1. Create the OU structure as pictured in Fig 1
   - “ouAllUsers”, “ouAdvancedUsers”, “ouLimitedUsers”, “ouTempUserAccounts”
2. View the properties of “ouAdvanceUsers”
3. Navigate to the Group Policy tab
4. Click the NEW button to create a new GPO
   - Type “linked GPO test 1” in the highlighted section to name the group policy
5. Create a new GPO in “ouAdvancedUsers” named “linked GPO test 2”
6. Create a new GPO in “ouAllUsers” named “linked GPO test 3”
**Group Policy Labs (3)**

**Linking a GPO**
1. View the properties of “ouLimitedUsers”
2. Navigate to the *Group Policy* tab
3. Click the *ADD* button to add a link to an existing GPO
4. Change the *Look In*: dropdownbox to “ouAdvancedUsers.ouAllUsers.nwtraders.msft”
5. Select “linked GPO test 1”
6. Click OK, then APPLY
7. Link group policy “linked GPO test 1” to “ouAllUsers”

A linked GPO is shared between the original GPO and all other linked object containers. Each has an equal right to add or remove configuration changes.
Changing a GPO’s Priority

1. Create a new group policy named “linked GPO test 2” in “ouAllUsers”
2. Use the UP and DOWN buttons to place the GPOs in the following order:
   1. “linked GPO test 3” (highest)
   2. “linked GPO test 2”
   3. “linked GPO test 1” (lowest)
3. Click APPLY
4. Run GPUPDATE at command prompt

Avoid creating GPOs with the same name as another GPO, even if the other GPO is in a separate container. Instead, use a GPO link or use a different name.
**Administrative Templates**

1. Select “linked GPO test 2” in “ouAllUsers”
2. Click the EDIT button to view the policies contained in the GPO
3. Examine each section by clicking to expand and collapse object trees

---

**Administrative Templates**

4. Create a new user inside of “ouAllUsers”:
   - FirstName = “Temp”
   - LastName = “User”
   - Username = “temp.user@nwtraders.msft”
   - Password = “P@ssw0rd”
   - Uncheck all optional checkboxes

5. Logon to the XP-Client *Bonn* as “temp.user”
   - Confirm that the background is plain
**Administrative Templates**

6. Select edit group policy “linked GPO test 2” in “ouAllUsers”
7. Navigate to the following section:
   - User Configuration → Administrative Templates → Desktop → Active Desktop
8. Enable “Enable Active Desktop”
9. Enable “Active Directory Wallpaper”
   - Name = "C:\classroom scripts\wallpaper1.bmp"
   - Style = “Tile”
10. Enable “Allow only bitmapped wallpaper”
11. Logoff, then Logon again to XP Client **Bonn** as the local administrator
12. Copy the wallpaper files into folder “C:\classroom scripts”
13. Logoff, then Logon again to XP Client **Bonn** as “temp.user”

**TROUBLESHOOTING…**

14. Disable “Active Desktop Wallpaper”
15. On the XP client **Bonn** attempt to set wallpaper2 as the wallpaper. **What happens to the wallpaper and why?**
16. Fix the error from step 15. Then re-enable “Active Desktop Wallpaper”
17. Change the wallpaper path name “C:\classroom scripts\wallpaper2.bmp”, then logoff and re-logon to **Bonn** as “temp.user”. **What happens to the wallpaper and why?**
18. Fix the error from step 17
19. Disable “Enable Active Desktop” policy, then logoff and re-logon to **Bonn** as “temp.user”. **What happens to the wallpaper and why?**
20. Fix the error from step 19
Filtering (Security Access List)

- Grants / Restricts access
- Specific individual functions
- Customizable
- Applicable to any available user account or user group
- Found under the security tab
- Available from:
  - GPOs
  - Files
  - Folders
- Matches an individual user or group to a list of permissions
- Not recommended for use...
  - Preference to OU / GPOs

SAL = SACL

SAL  = **Security** Access **List**
SACL = **Security** Access Control **List**
Individual GPO Options

• “No Override”
  – Forces a specific group policy be accepted by lower hierarchies despite the lower policy using “Block Policy inheritance”

• “Disabled”
  – Disallows the policies set in a given GPOs configuration from taking effect
  – Does not remove the group policy or change its priority level
  – GPO is still available to re-enabled at a later time

Both “No Override” and “Disabled” are set distinctly to an individual GPO. The “Block Policy Inheritance” option is applied to all GPOs. Within a single GPO “Block Policy Inheritance”, “No Override”, and “Disabled” are respectively discrete and individual.
Deploying and Installing Software (1)

• ...is a GPO configuration
  – *Computer Configuration*
    • Applies to the computer
    • Will enact to the computer for any user logged on
  – *User Configuration*
    • Applies to the user
    • Will enact for a user logged on to any computer

• Requires an installation Package
  – Only supports ".msi" files
  – .msi must be present locally or shared
There are two deployment methods available with a group policy software deployment:

1. **Published deployments** are not actually installed. As the name suggested they are merely published to a certain area of the user’s awareness so that the user can choose when or if the installation will occur. Obviously, computers cannot make decisions for themselves; thus, the published deployment option is not available under the software deployment policy for computer configuration.

2. **Assigned deployments** may be assigned to a computer’s logon procedures or a user’s logon procedures. Assigned deployments start and nearly complete an installation at login time. However, normally the user must provide some feedback to finalize the installation. Installation packages can be made to reduce the amount of user feedback required. Packages can be coupled with installation scripts that provide the installation prompts appropriate information without the users intervention.

With either deployment method, the installation package must be available on the client’s local hard-drive or available via a network share. Although network shares offer a central point of distribution, they also produce a LAN access bottleneck. Consequently, placing a potentially large installation package on a
great number of client computers can be time consuming and produce network slowdowns. A possible compromise is to rollout the package to only a few users at a time by making the installation package available on a network share, but limited to only a few users. It is also possible to provide users with a CD-ROM loaded with the installation package.
Deploying and Installing Software (3)

• Advanced Options
  – Found in properties dialog box of an installation package
  – Deployment Options**
    • Auto-install by file extension activation
    • Uninstall when outside management scope
    • Do not display in Add/Remove Programs control panel

There are three basic Deployment Options…

1. **Auto-install by file extension activation**
   Auto install, installs the software package when the user clicks on a file that has a file extension associated with the uninstalled software. This option is checked by default. Without clicking on an associated file, users can install (or finalize) the installation of a deployed package by launching the program formally or by locating the software installation package in the Add/Remove Programs control panel.

2. **Uninstall when outside management scope**
   Removes the program from the client computer when the software package no longer exists or when the GPO containing the software package is deleted. This is useful for software that is to be installed on a temporary basis.

3. **Do not display in Add/Remove Programs control panel**
   This option prevents the software package from being visually available from the Add/Remove Programs control panel. Users can still install published software or finalize the installation of assigned software by attempting to open an associated file.

**category and/or sub-categories have been abbreviated or summarized
**Slow Link Processing**

<table>
<thead>
<tr>
<th>Group Policy configured to run only when there is an adequate network connection:</th>
<th>When Group Policy detects a slow link, it applies the following default settings, unless they are modified:</th>
</tr>
</thead>
<tbody>
<tr>
<td>– Software installation and maintenance</td>
<td></td>
</tr>
<tr>
<td>– Scripts</td>
<td></td>
</tr>
<tr>
<td>– Disk quota</td>
<td></td>
</tr>
<tr>
<td>– IP Security</td>
<td></td>
</tr>
<tr>
<td>– Internet Explorer maintenance</td>
<td></td>
</tr>
<tr>
<td>– Security settings: ON</td>
<td></td>
</tr>
<tr>
<td>– Administrative templates: ON</td>
<td></td>
</tr>
<tr>
<td>– Software installation and maintenance: OFF</td>
<td></td>
</tr>
<tr>
<td>– Scripts: OFF</td>
<td></td>
</tr>
<tr>
<td>– Folder redirection: OFF</td>
<td></td>
</tr>
<tr>
<td>– Internet Explorer maintenance: OFF</td>
<td></td>
</tr>
</tbody>
</table>

Where a fast or *noisy* internet link is unavailable, a group policy can be configured to limited bandwidth requirements. Two aspects of Active Directory, however, cannot be limited:

1. Security settings

2. Administrative templates.
TAB

Insert Tab # 8 Here
Disaster Recovery
Disaster Recovery

Unit objectives
• Create backups of the Active Directory database
• Check the integrity of the Active Directory database
• Recover the Active Directory database
• Create a disaster recovery plan

Topic A

• Topic A: Backing up Active Directory
• Topic B: Identifying Active Directory corruption
• Topic C: Recovering Active Directory
• Topic D: Creating a disaster recovery plan
The Need for Backups

• To minimize the chances of losing data when a domain controller fails

Active Directory Backups (1)

• Backed up as part of the System State
• System State includes
  – Registry (always)
  – COM+ Class Registration database (always)
  – Boot files (always)
  – Certificate Services database (if Certificate Services is installed)
  – Active Directory (only on domain controllers)
Active Directory Backups (2)

- SYSVOL directory (only on domain controllers)
- Cluster service (if the server is part of a cluster)
- IIS Metadirectory (if IIS is installed)
- System files (always)

Backing up System State Data
The Backup Utility

- Normal backup
- Differential backup
- Incremental backup

Activity A-1

Creating a System State backup
Lab 1

1. Log on to your server
2. Choose START, ALL PROGRAMS, ACCESSORIES, SYSTEM TOOLS, BACKUP
3. When the wizard starts click advanced mode
4. Click the backup tab
5. In the left tree pane select system state
6. Build a folder on your computer named backup
7. Click start backup
8. Click close on the backup dialog box
9. Go to your back up and check out the properties

Backups and the Tombstone Lifetime

- Controls how long tombstones remain in the directory
- Controls the maximum age of a backup that can still be restored
Topic B

• Topic A: Backing up Active Directory
• **Topic B: Identifying Active Directory corruption**
• Topic C: Recovering Active Directory
• Topic D: Creating a disaster recovery plan

File Integrity Check

• Performs a low (binary) level check of the Active Directory database file (NTDS.DIT)
• Checks overall structures of database tables
• Ensures tables are accessible and have the correct columns
Activity B-1

Performing an integrity check

Lab 2

1. Restart your server in to directory service mode
2. Log on to your server
3. Go to start then run type in cmd
4. Type in ntdsutil
5. Type files
6. At the file maintenance prompt enter integrity
7. Once it is done then type in quit
Semantic check

- Verifies Active Directory database at higher level
- Determines if every object has a
  - GUID
  - Distinguished name
  - Valid security descriptor

Activity B-2

Performing a semantic check
Lab 3

• At the ntdsutil enter semantic database analysis
• Enter go
• Type in dsdit.Dmp.0
• Enter exit
• Enter quit

Topic C

• Topic A: Backing up Active Directory
• Topic B: Identifying Active Directory corruption
• Topic C: Recovering Active Directory
• Topic D: Creating a disaster recovery plan
Active Directory Recovery Options

- Soft recovery
- Restore from backup
- Reinstall Active Directory
- Repair Active Directory database

Activity C-1

Using NTDSUTIL to perform soft recovery
Lab 4

1. At the ntdsutil
2. Type files
3. Type recover
4. When completed type in quit
5. Restart the server

Restoring Active Directory

• Non-authoritative restore
  – Restores damaged database from a System State backup
• Authoritative restore
  – Performed on domain controller
  – Restores objects that were mistakenly deleted or modified
Activity C-2

Performing non-authoritative and authoritative restores

Lab 5 (1)

- Go to active directory users and computers
- Right click a ou and delete it
- When ask are you sure say yes
- Restart your server in directory services mode
- Go to start then run
- Type in ntbackup
- Advanced mode
- Click restore and manage media
- Go to where you had built your backup
- Make sure you click system state
- Click restore
- When done when ask restart say no
- Go to cmd prompt
- Type in ntdsutil
- Type in authoritative restore
Lab 5 (2)

1. Type in the following
2. RESTORE SUBTREE "OU=, dc=bnoc,dc=class"
3. Click yes
4. Once done enter quit
5. Close cmd prompt
6. Restart your server
7. Go to your user and computers to see if it restored your deleted ou.

Steps for Reinstalling Active Directory

• Run DCPROMO
  – Demotes domain controller with the corrupt database
• Clean up the server’s metadata in Active Directory
• Run DCPROMO again
  – Promote the server to a domain controller
Repairing Active Directory

• The Repair command
  – Low-level rebuild of database
  – Deletes data that is not valid
  – Can fix only data that it can find; data that is missing or incomplete cannot be repaired

Activity C-3

Reinstalling and Repairing Active Directory
Disaster Recovery Plan

• Determine:
  – What to back up
  – When to back up
  – Where to back up
Creating a Plan

• Things to be included in a plan
  – Server hardware specifications
  – Network layout
  – Active Directory physical layout
  – Active Directory logical layout
  – Server software configurations
  – Active Directory file layout (store and log file locations)
  – Label removable media and include a backup and rotation description

Testing the Plan

• Stage a fake disaster
• Use spare hardware
• Follow the recovery plan
• Ensure all the information is available
• Make corrections and additions
• Test backups
Activity D-1

Planning backups

Unit Summary

• Created backups of the Active Directory database
• Checked the integrity of the Active Directory database
• Recovered the Active Directory database in the event of failure
• Created a disaster recovery plan
This page intentionally left blank.
For comments or suggestions on this book, please email us at:

itfsb.cecom@us.army.mil

Subject Line: Books