UCRL-MA-148650

A Guide to Microsoft Active Directory (AD) Design

John Dias

May, 2002

U.S. Department of Energy

Lawrence
Livermore
National
Laboratory

DISCLAIMER

This work was performed under the auspices of the U. S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48.

TABLE OF CONTENTS

Executive Summary *1*

1.0 Introduction *1*

Part I: Active Directory Overview *3*

2.0 Active Directory Tutorial *3*

2.1 Directory Services **3**

2.2 Microsoft Active Directory **4**

2.3 Components of the Active Directory **4**

2.3.1 Domain 4
2.3.2 Trees 5
2.3.3 Forest 6
2.3.4 Organizational Units 7
2.3.5 Schema 7
2.3.6 Group Policy Objects 8
2.3.7 Global Catalog 10

2.4 Naming Contexts, Partitioning, and Replication **11**

2.5 Kerberos Trusts **12**

2.6 Delegation of Authority **13**

3.0 Microsoft's Active Directory Design Process *15*

3.1 Forest Plan **17**

3.1.1 Forest Planning Process 17
3.1.2 Determining the Number of Forests 17
3.1.3 Forest Change Control Policy 18
3.1.4 Changing the Forest Plan after Deployment 19

3.2 Domain Plan **19**

3.2.1 Domain Planning Process 19
3.2.2 Determining the Number of Domains in each Forest 20
3.2.3 Choose a Forest Root Domain 20
3.2.4 Assign a DNS name to each domain to create a domain hierarchy 20
3.2.5 Plan the DNS Server Deployment 21
3.2.5.1 Background 21

3.3 Organizational Unit Plan **22**

3.4 Site Planning Process **25**

4.0 Scope of AD Design *27*

Part II: Active Directory Design Scenario *29*

5.0 Description of Hypothetical Site *29*

5.1 Pragmatic Discussion of Forest and Domain Planning **29**

5.2 LCIS Design Requirements **31**

5.2.1 Programmatic Requirements 31

6.0 Comparison of Three Design Approaches 32

6.1 Single Domain 33
6.1.1 Single Domain Design Description 33
6.1.2 Single Domain Benefits 34
6.1.3 Single Domain Draw Backs 34
6.1.4 Single Domain User Perspectives 35
6.1.5 Single Domain Concluding Remarks 35

6.2 Multiple Domain Model 36
6.2.1 Multiple Domain Description 37
6.2.2 Multiple Domain Benefits 38
6.2.3 Multiple Domain Draw Backs 38
6.2.4 Multiple Domain User Perspective 39
6.2.5 Multiple Domain Concluding Remarks 39

6.3 Multiple Forests 40
6.3.1 Multiple Forest Description 41
6.3.2 Multiple Forest Benefits 42
6.3.3 Multiple Forest Drawbacks 42
6.3.4 Multiple Forest User Perspectives 43
6.3.5 Multiple Forest Concluding Remarks 43

6.4 LCIS' Active Directory Design 44

Part III: Active Directory Best Practices 46

7.0 Best Practices for Active Directory Design 46

Appendix A. DNS Options 48

Appendix B. Bibliography 55

Executive Summary

The goal of this paper is to facilitate the design process for those DOE sites that are currently engaged in designing their Active Directory (AD) network. It is a roadmap to enable analysis of the complicated design tradeoffs associated with Active Directory Design. By providing discussion of Active Directory design elements which are permanent and costly to change once deployed, the hope is to minimize the risks of sponsoring failed designs, or joining existing infrastructures not suitable to programmatic needs.

Specifically, most Active Directory structures will fall under one of three common designs: Single Domain, Single Forest with Multiple Domains, or Multiple Forests. Each has benefits and concerns, depending on programmatic and organizational structures. The comparison of these three approaches will facilitate almost any Active Directory design effort.

Finally, this paper describes some best practices to consider when designing Active Directory based on three years of research and experience.

1.0 Introduction

Active Directory design is an enormous task. The technology has more capabilities and is therefore much more complex than any other networking technology available today. Because of this, many organizations are late deploying AD into their production environment.

The goal of this guide is to facilitate the design process for those DOE sites that are currently engaged in designing their Active Directory network. This guide is based on personal experience and a two-year design process that included planning, meetings, documentation, and training. This information has the potential to cut the design time by 50% and produce more tangible results than using the Microsoft design process alone.

This guide provides a general tutorial of Active Directory concepts as well as highlights some of the pitfalls, issues, and misinformation to be aware of when designing Active Directory for a site. Additionally, this guide demonstrates three common Active Directory designs and design tradeoffs by presenting a pragmatic scenario. To accomplish this, it is broken into 3 parts. Part I is comprised of an overview of Active Directory. Specifically, Section 3 outlines an Active Directory Tutorial and Section 4 describes the Microsoft Design process. Section 5 describes how to scope an AD Design. Part I provides the basis for understanding the design scenarios illustrated in Part II and best practices described in Part III.

Part II is a scenario designed to illustrate Active Directory concepts in context of a realistic situation. More specifically, Section 6 begins to describe the scenario by presenting AD design requirements for a fictitious DOE site based on a typical operational networking environment (laboratory or production site--for the purposes of

this design, the Active Directory design would be similar). Section 7 compares three common AD designs through a scenario which tracks a fictitious AD design team's process, progress, and decisions. Finally, Part III highlights some best practices useful to gauging new designs and facilitating discussions.

Part I: Active Directory Overview

2.0 Active Directory Tutorial

Unfortunately, many aspects of AD are technically complex and most of the terms used to describe this suite of technologies are new. As a result, this tutorial is complicated but necessary to comprehend the design process.

2.1 Directory Services

What is a directory in computing terms? A classic analogy is the white and yellow pages of a telephone book. A common feature of both white and yellow pages is the ability to search for information; the difference in the two is the way they are indexed.

Publishing information in a directory and allowing users, applications, and systems administrators to make use of this information is the fundamental advantage of a Directory.

Directories, such as Lightweight Directory Access Protocol (LDAP) and Active Directory (AD) are types of databases that can be searched to provide useful network information. A user can find network information without any knowledge of the structure of the network. For example, the user can search the Active Directory for a share, requiring no knowledge of the network. This is because the directory has abstracted a server's share to a directory share. Without Directory Services, a user has to know the server name and its share name to mount a network file share. AD changes this.

Searching is a fundamental service provided by LDAP, so the more information "published" in the directory, the more productive the user community becomes. LDAP is a standard and the Active Directory is LDAP compliant. Since AD adheres to the LDAP standard, third party applications are leveraging the directory. AD-aware applications can use Windows 2000 services for authentication and access controls. These applications can store configuration information in the directory.

For example, consider Microsoft's firewall Internet Security and Acceleration Server (ISA) as an LDAP aware application. When ISA is used as an Intranet Proxy and cache server, the security policy for each proxy server is published in the Active Directory. Picture an enterprise with 10 internal firewalls protecting internal web based applications. Since the policy is located in the directory, the security organization can enforce common rules on each and every firewall. The directory makes complicated policies possible such as applying a baseline firewall policy for all servers, then a more restrictive policy for specific servers.

System managers can gain the most benefit from directory services. Currently, NT and UNIX models for system management are comprised of discrete tools for each type of management operation. Each tool has its own configuration data storage (files, databases)

and the configuration information is scattered throughout the system. Also, there is a steep learning curve for the systems managers to learn nuances of each management utility.

Active Directory, on the other hand, stores all of the domain information in a common and searchable format. All the user accounts, computer accounts, group accounts, access control lists, security identifiers, Group Policy Objects (GPOs), shares, printers, properties about people and their locations, are all stored in the Active Directory. Moreover, a common interface and management paradigm, Microsoft Management Console, is provided to the administrator for each of the administrative tasks and functions.

2.2 Microsoft Active Directory

Active Directory is Microsoft's implementation of directory services. It is based on various standards, most importantly LDAP and X.500 (the schema is based on X.500).

In addition to compliance with LDAP, AD has additional features and compatibility such as the close integration of the directory services to Windows domains and Domain Name Service (DNS). The integration of directory services to Windows domains is the key to directory scalability (domains and scalability will be described below). AD security, authentication, and access control are also provided by the integration of the domains to the directory. While this approach works well, the integration of AD to Windows domains forces the choice of Active Directory services when selecting the Windows 2000 operating system.

The integration of DNS to Windows domains is a feature that makes the design and implementation of Active Directory both complicated and invasive to the existing infrastructure**.** Importantly, **A Windows domain must be named identically to its DNS domain**. The same DNS name is used for both the IP address resolution and the Active Directory domain name.

2.3 Components of the Active Directory

2.3.1 Domain

The core unit of logical structure in the Active Directory is the domain, which can store millions of objects. Objects stored in the domain are considered "Interesting" to the network. "Interesting" objects are items the networking community members need to do their jobs: printers, documents, e-mail addresses, databases, users, and other resources. All network objects exist within a domain and each domain stores information only about objects it contains. Active Directory is made up of one or more domains.

Grouping objects into one or more domains will allow the network to reflect a DOE site's organization. Domains will allow each internal division to partition their information from the rest of the organization.

Domains share these characteristics:

- All network objects exist within a domain and each domain stores information only about the objects that it contains.
- A domain is a security boundary. Assess control lists (ACLs) control access to domain objects. All security policies and settings such as administrative rights, security policies, and ACLs do not cross from one domain to another.
- The Domain Administrator has absolute rights to set policies only within that domain.

The atomic unit of the Windows 2000 is the domain. A domain is an administrative boundary, a security boundary, and represents a name space that corresponds to a DNS domain.

The first domain created in a Windows 2000 deployment is called the root domain. **Since Windows 2000 domain structure is married to DNS domain hierarchies, the structure of the domain hierarchies are similar.**

Most organizations large enough to require more than one domain have a logical structure that divides responsibilities or work focus. Domains are ideal for logical partitioning.

Windows 2000 network domains are organized in a hierarchy. The concepts of forests and trees were introduced to leverage the hierarchical approach to domains.

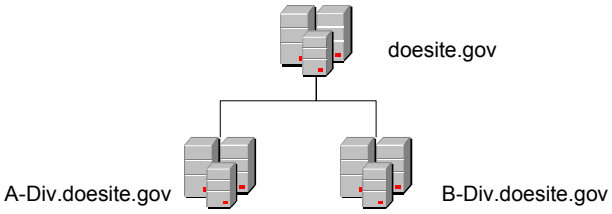
An important note: in the domain hierarchical structure, user rights and group policy are inherited throughout the OU hierarchy.

2.3.2 Trees

Domain trees are collections of Windows 2000 domains that form a contiguous name space. A domain tree is formed as soon as a child domain is created and associated with a given root domain. A domain tree looks like an inverted tree (with the root on top), with branches (child domains) sprouting out below.

Trees are the structural elements that ensure the scalability of the Active Directory. As each domain is a partition (part of the entire directory), trees allow the hierarchical structure necessary for organizations, much like DNS domain structure does for the Internet. **Domains within a tree must be named identically to their DNS domain names**.

The diagram below shows a hierarchical tree; the domain names are the same as the DNS domain names (note: the domain names are hypothetical examples only).



Active Directory Domain names are the same as the DNS domain names

2.3.3 Forest

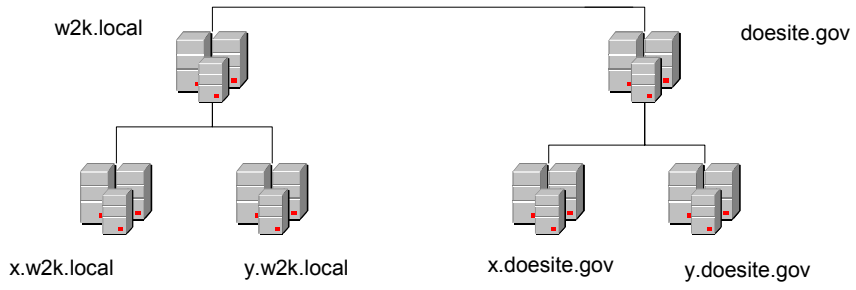
There are cases where two or more domain trees, each represented by separate DNS name space, need to be included as one enterprise. A tree must be represented by a contiguous DNS name space and disallow participation of domains that are not within its name space. The mechanism for connecting one or more trees is the **Forest.**

All trees within a forest share the following benefits:

- Common Schema
- Common Configuration (AD infrastructure information)
- Global Catalog
- Each and every domain within the forest can leverage the Kerberos transitive trust mechanism

(Note: An Active Directory consisting of only one tree with a single domain is considered a forest. A non-contiguously named single domain is still considered a tree).

Consider the diagram of a forest below. This forest consists of two trees. The w2k.local DNS domain (which is a private unregistered DNS domain), and the doesite.gov DNS domain are non contiguous and therefore are separate trees. The w2k.local domain is the root of its tree and the doesite.gov domain is the root of its tree. Since w2k.local was the first domain created in the forest, it is also the root of the forest.



A Forest with 2 Trees

2.3.4 Organizational Units

The Organizational Unit (OU) is a critical design factor impacting security, policy, efficiency, and the cost of administration. Organizational Units are a type of LDAP (X.500) container. It can be thought of as a sub-domain element with similar properties to domains. They are components internal to domains. OUs are part of the LDAP name space and not the DNS name space.

OUs can be arranged in a hierarchical structure. **Unlike the domain hierarchical structure, user rights and group policy are inherited throughout the OU hierarchy.**

One of the main benefits of OUs is their ability to accomplish domain functions and therefore reduce the total number of required domains. In fact, a common NT to Windows 2000 migration strategy is to upgrade the NT domain master domain to a Windows 2000 AD, then collapse all of the NT resource domains into Organizational Units.

OUs are commonly used to contain user accounts, group accounts, and computer accounts. Powerful configurations can be obtained when the OU design is harmonized with group policy and security groups.

Another benefit of Organizational Units is the concept of delegation of authority. Domain Administrators can delegate partial administration rights through the OU. The granularity of the delegated rights is quite fine. Take the case of a help desk as an example. The domain administrator can delegate the "right" to reset passwords to help desk personnel and therefore, offload the domain administrator's responsibility of fielding calls pertaining to lost or expired passwords. The change-the-password right is usually enforced by a Group Policy Object (GPO), filtered by security groups, and applied at the OU level.

Architecturally, the design of the OU structure usually reflects the Information Technology structure. To paraphrase many authors on this subject, "design the OU structure with the administrators in mind."

2.3.5 Schema

The schema dictates the data definitions for the AD. If an object or attribute is not in the schema, that object/attribute will not be stored in the AD.

The directory contains information in the form of objects and object attributes. The directory is actually a type of database that is optimized for querying. Data that is more or less static and is searched often can be beneficially stored in the directory. Data that changes often is not a good choice for storage in the directory. For example, user properties such as phone number, building number, pager number, and application configuration data are examples of information that can be effectively managed by directory services, as these types of data are fairly static. These types of data are queried much more often than they are changed. System logs and file systems are not good candidates for the directory as these data are extremely dynamic.

The schema manager, an administrative utility, defines what attributes are published in the Global Catalog (GC), see below. A very important aspect of the AD design is choosing the information to be published in the GC. The schema manager allows this definition.

2.3.6 Group Policy Objects

Group Policy Objects are especially critical to the justification for additional domains. Group Policy is the primary component of Windows 2000's implementation of Change and Configuration Management (CCM), and is the primary mechanism for establishing uniform, effective security policies within a Windows 2000 domain.

As the name implies, Change and Configuration Management involves managing the ongoing change and configuration issues that arise as administrators try to ensure that people are productive as they use their computers. This ability, once the associated GPOs are designed correctly, is central to reducing the Total Cost of Ownership of a Windows network.

The table below highlights CCM.

	Feature		Benefits	Technologies
Change and Configuration Management	IntelliMirror*	**User Data Management**	*"My data and documents follow me!"* Users can access the data that they need to do their job, whether they are working online or offline, or when roaming from one computer to another on the network. Administrators centrally manage this feature by policy to minimize support costs.	• Active Directory™ • Group Policy • Offline Folders • Synchronization Manager • Enhancements to the Windows Shell • Disk Quotas
		Software Installation and Maintenance	*"My software follows me!"* Users have the software they need to perform their job. Software is self-repairing, and both the software and features install 'just-in-time.' Administrators centrally manage this feature by policy to minimize support costs.	• Active Directory™ • Group Policy • Windows Installer Service • Add/Remove Programs in Control Panel • Enhancements to the Windows Shell

		User Settings Management	*"My preferences follow me!"* Users get the same experience from any desktop. Personal preferences and settings for desktops or software are available whenever the user logs on. Administrators centrally manage this feature by policy to minimize support costs.	• Active Directory™ • Group Policy • Offline Folders • Roaming User Profiles • Enhancements to the Windows Shell
		Remote OS Installation	Administrators can enable installation and configuration of the Windows 2000 operating system on new or replacement computers without staging or on-site technical support.	• Active Directory • Dynamic Host Configuration Protocol • Remote Installation Server

Change and Configuration Management is the realization of the original goals of Microsoft's Zero Administration for Windows (ZAW) initiative that Microsoft announced in October 1996. The main goals are bulleted below.

Automatic system update and application installation
The operating system will update itself when the computer is booted, without user intervention, seeking the latest necessary code drivers from a server. The automatic desktop feature will provide users with all available applications, installing them automatically when invoked.

- **All state kept on servers**
  User's data can be automatically "reflected" to servers, ensuring high availability and allowing mobile users to have access to their information whether they are connected to a network or not. Additionally, users will be able to roam between PCs while maintaining full access to their data, applications, and customized environments.

- **Central administration and system lockdown**
  All aspects of client systems will be controllable by a central administrator across the network. In a few simple steps, the system can be "locked down" to maintain controlled, consistent, and **secure configurations** across sets of users.

The goals of ZAW are very aggressive; in fact these goals did not seem achievable in 1996. GPO, along with a few other technologies, has met these goals.

The benefits of a good GPO design are great. Consider the current requirements of fixing vulnerabilities as reported by the scanning project. Currently, a system manager has to visit each computer and perform a registry fix. Using Group Policies and the Active Directory, an administrator fixes the policy once, in the Group Policy Object; the AD will then push the fix to every computer on the domain.

Another case is a hot fix (patch). The hot fix can be pushed out to every computer in the domain via GPO.

Security policies can be pushed out to every computer and user account within the domain; these policies are not only enforced, they are also refreshed at a settable interval.

All of the benefits of Group Policy Objects come at a cost. Designing a GPO strategy and applying GPO is one of the most complicated aspects of Windows 2000 and the Active Directory. There are over 700 settings that are configurable with GPO, but the vast number of settings is not the most complex aspect. The architecture and implementations are more complicated.

There are some other key technologies that are also implemented via GPOs that were not covered in this introduction. These technologies may be the key to implementing controls for critical computer systems. **The GPOs associated with these technologies are Domain concepts.**

- Encrypted File System
- IP Security (IPSec: Secure extensions to the TCP/IP protocol stack)
- PKI (Certificate Authority and services)
- SMB Signing (protection for Microsoft file shares)

Utilization of the above technologies can be justification for new programmatic domains.

2.3.7 Global Catalog

The Global Catalog is used to improve the response time of LDAP searches. The GC consists of selected properties from every object in the forest. The properties included in the GC are generally useful for searches and are considered static (dynamic properties would cause excess replication).

A functional description of the GC follows. Wherever a query (LDAP, not DNS) comes into the Active Directory, the first repository searched is the Global Catalog. This is why the GC only holds properties of objects that are useful for searching. If the GC does not contain the property of the object being searched for, the query is referred (LDAP referral) automatically to the Active Directory. Therefore, the AD is searched only for lesser-used properties.

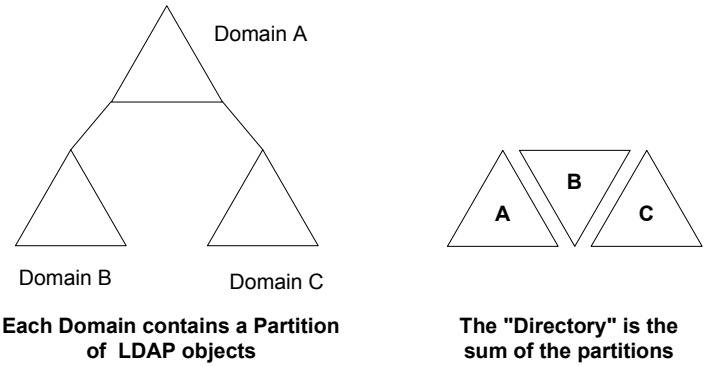
One of the critical points to remember with the GC is that searching for a property in the GC will be forest-wide as the GC is a forest-wide catalog; however, if you are searching for a property that is not in the GC, ***then the search will be conducted only in the current domain.***

The concept of "current domain" is critical to an LDAP search. This has a major impact on the DNS design. If the information is not located in the GC, for example, a computer (client) is located in the DNS domain X.gov, and the domain controller is located in the Y.gov domain, then **the LDAP query will search the wrong domain,** as the "current

domain" is Y.gov for the client, but the information is located on the domain controller located in X.gov domain, (see DNS Option #5, in Appendix A).

2.4 Naming Contexts, Partitioning, and Replication

The Active Directory contains all the network information for the forest. As described above, each domain is a separate partition of the directory and is also considered a separate name context.



This partitioning ensures that the directory will scale. Although a single domain can contain millions of objects, there are various cases for adding a domain to the forest. Adding a new domain has a minimal effect on the other domain's contents. The new domain is another partition containing its own information (objects).

The Active Directory is partitioned into three naming contexts: Domain Naming Context, Configuration Naming Context, and Schema Naming Context. A domain is its own naming context and its scope is localized to its domain members. There are two other naming contexts whose scopes are forest-wide:

1. **The schema**, which contains the object data definitions, is a separate name context and is replicated to every domain controller in the forest.
2. **The configuration** is also a separate naming context that is also replicated to every domain controller in the forest. The configuration has structural information, such as the location of sites (see below), the location of domain controllers, subnets, global catalog servers, and a complete list of all the domains in the forest. The configuration also has information for each domain that is not in the forest and has a trust relationship with any domain in the forest.

Each name context must be replicated through its scope. The Domain Naming Context is replicated to all the domain controllers within the domain. The Schema and Configuration Contexts are replicated to every domain controller in the forest.

Replication is another major aspect of designing an AD. (Replication design is outside the scope of this paper).

There is one more major function that is also replicated throughout the forest, the Global Catalog (GC). The Global Catalog is not considered a separate naming context; it is actually a partial replica of all the objects in the forest.

2.5 Kerberos Trusts

Trusts allow for the potential of authenticating security principles from domain to domain. Windows NT3 and 4 trust mechanism was based on LAN Manager (NTLM) trusts. Trusts are mechanisms that will allow trusting domains to authenticate and authorize principles from domain to domain. For example, if I have an account in Domain A, and Domain B trusts Domain A, I can login to Domain B using my credentials from Domain A. There are two big problems with NTLM trusts; the trusts are one-way only and are not transitive. In order for Domain A and Domain B to trust each other, a one-way trust must be established in both directions.

- Domain A must establish a trust to Domain B
- Domain B must establish a trust to Domain A

Not having transitive trust adds to the number of trusts that must be established using NTLM. The aspects of transitive trusts are bulleted bellow.

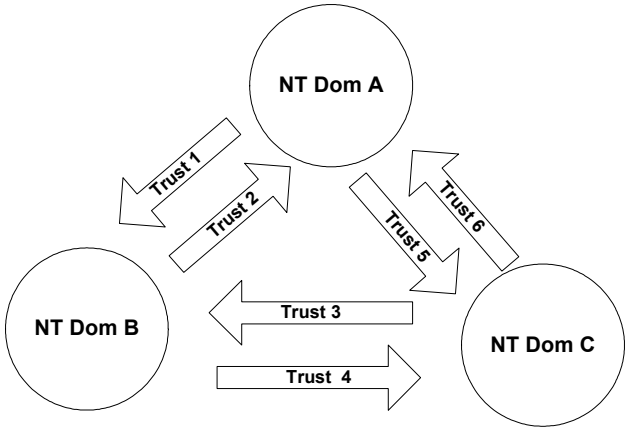
- If Domain A trusts Domain B
- And, Domain B trusts Domain C
- Then, Domain A trusts Domain C

Without transitive trusts the scenario above would be:

- If Domain A trusts Domain B
- And, Domain B trusts Domain C
- Then, Domain A **does not trust** Domain C

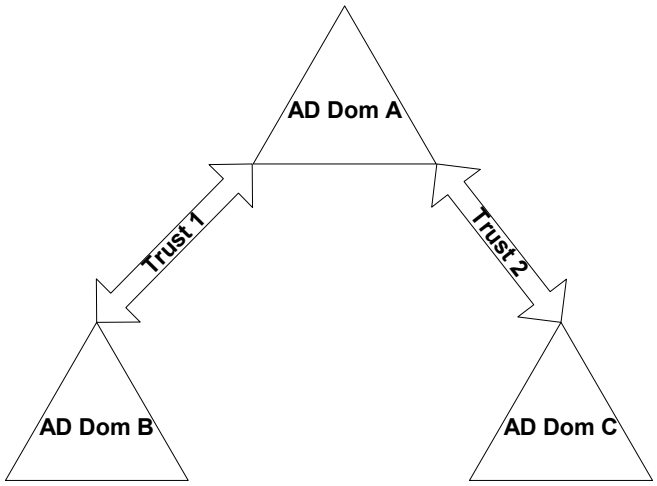
Kerberos Trusts of Active Directory are transitive and bi-directional trusts. This simplifies the management of trusts (reducing the number of trusts) and facilitates the sharing of information within a Forest. A Forest can be viewed as a complete trust model of authentication.

The diagram below shows the trusts necessary for 3 NT4 domains. A complete trust model requires the (number of Domains) times (number of Domains minus one). Complete trust for 3 domains require 6 one-way trusts, 4 domains require 12 one-way trusts, and 5 domains require 20 one-way trusts.



NT 4 Domains, one way Trusts.
Trusts are not Transitive

Since Kerberos trusts are bi-directional and transitive, the number of Kerberos trusts per domain for Active Directory is simply equal to the number of domains minus one, see diagram below.



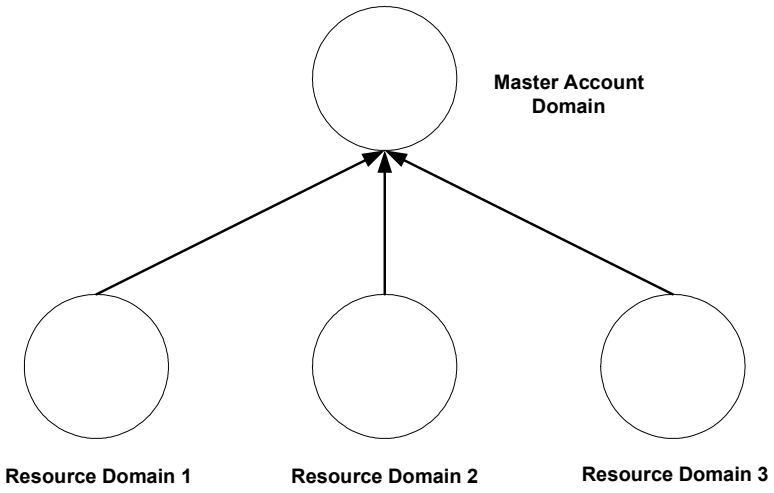
The Number of Kerberos Trusts is the
Number of Domains minus 1

The main benefits of Kerberos trusts are the reduction of the number of trusts, and since trusts are transitive, the trust model is that of complete trust. (Note that the benefit of complete trust can be a design constraint for strict security requirements). (Also note, Microsoft's symbol for NT domain is a circle, and their symbol for AD domain is a triangle).

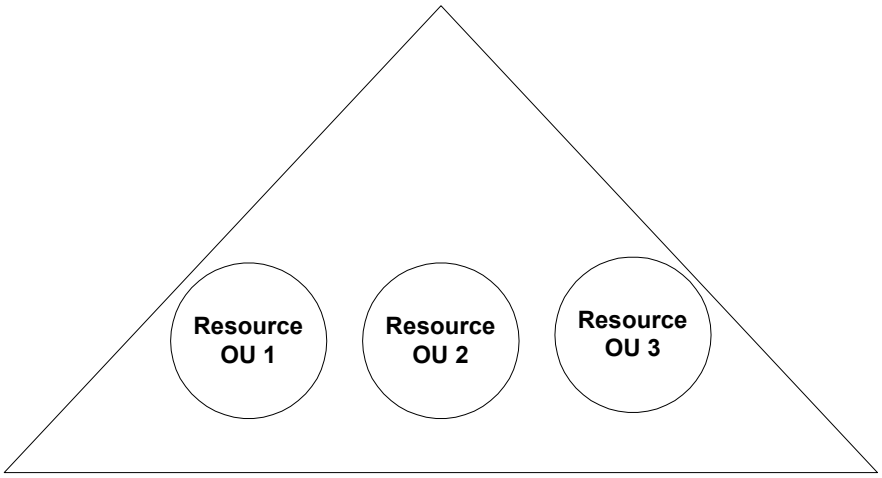
2.6 Delegation of Authority

Delegation of authority was covered briefly in a previous section. This section covers some more details of this concept.

Large NT 4 architectures (anything with over 40,000 accounts, a SAM limitation) required a master account domain model where all the intuitional accounts were in the accounts domain and the managed resources were located in resource domains. Resource domains trusted the account domain; the master account domain did not usually trust the resource domain, (arrows point to trusted domains). See the diagram below.



The fundamental problem with this model is that the number of domains would tend to grow. In order for an organization to manage their resources, they need a separate domain. The administrator account scope is that of the domain, and there was no mechanism for the delegation of sub administrator accounts. Active Directories' "delegation of authority" has greatly reduced the required number of domains. It is now possible to delegate administrator authority to Organizational Units. The AD equivalent of the four NT4 domains above is now a single domain with three Organizational Units (see the diagram below).



Active Directory Domain with the NT4 resource domains collasped into OUs. The Domain Administrator has delegated authority to the OU administrators.

The Domain Administrator can delegate "Full Control" of the OU to a security group which effectively gives the members administrative privileges for the OU. However, the **administrators of the OU must trust the domain administrator**, as this account is still all-powerful within the domain.

Another interesting feature of "delegation of authority" is that delegation can be accomplished at a very granular level. The most common example of this type of limited authority is that of a help desk. A domain administrator can delegate the authority "change user password" to the help desk personnel. When a domain user forgets their password, the help desk can reset the user's password. The help desk personnel could have no other privileges within the domain.

The main benefit of "delegation of authority" is that it can reduce the number of domains, and also provide specific privileges closely tailored to the task (think of exact privileges to do the job).

3.0 Microsoft's Active Directory Design Process

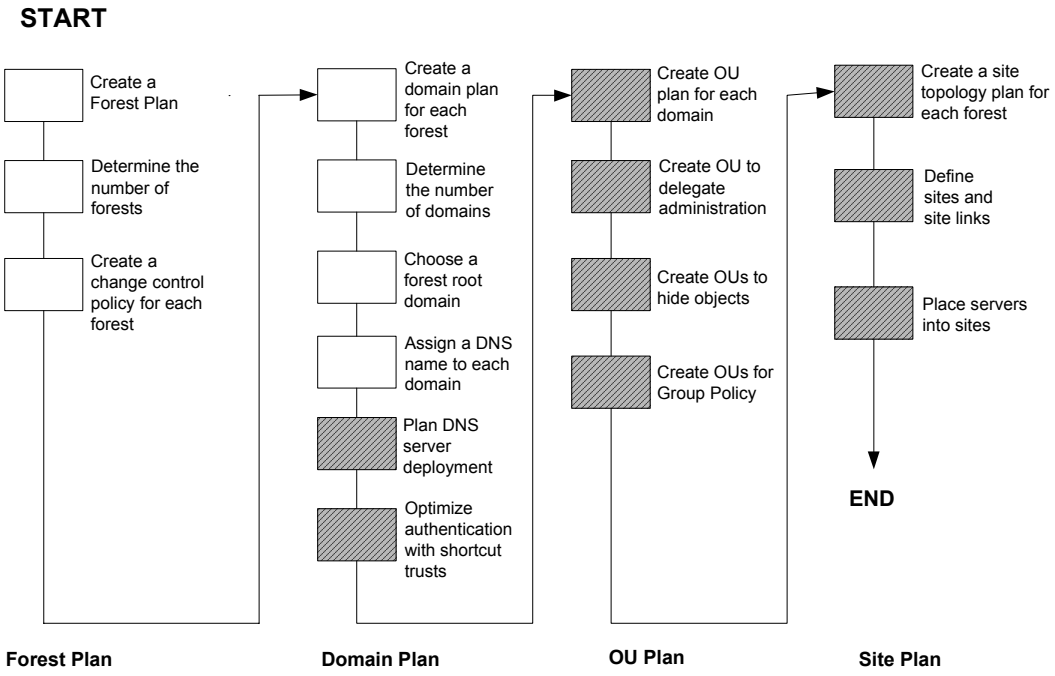
Active Directory design is an enormous task. Many organizations are late deploying AD due the design complexity. Recognizing the complexity of this task, Microsoft has provided an Active Directory planning process in the form of an Active Directory Deployment and Planning Guide (see bibliography). Section 5 (Microsoft's Design Process) will explain this process.

The scope of an Active Design can be an entire enterprise and this effort will require a design team with members from many and various organizations. Generally, the design processes that were used to design NT4 domains will not work for AD design.

There are many design methodologies for AD or LDAP design. Some of these design methodologies used by Corporate America are very formal and very rigorous. It is not the intent of this paper to cover or develop a formal methodology for AD design; the scope of this paper is merely to provide a road map for developing and tracking such a process.

Microsoft's *Deployment and Planning Guide*, which is part of the Windows 2000 Server Resource Kit, is an excellent starting point for a successful design. This section is a synopsis of Chapter 9: "Designing the Active Directory Structure," (note: a manager responsible for an Active Directory Design effort should master all the concepts in the Deployment Planning Guide).

The diagram below depicts the Microsoft design process.



This paper focuses on the first 7 steps of the design process, the permanent aspects of the design. These are the most important because incorrectly designing and implementing them will result in an unusable architecture, requiring a complete wipe of services, and starting over. A fundamental architectural approach for Active Directory design is to push the design complexity to the lower level aspects of the architecture down to the Organizational Units, (see Best Practice #1 in Part III).

The design aspects of OUs require complete knowledge of the mission, personnel, and operational procedures of the department or project represented by the OU. The parameters of the OU can be subject to rapid changes such as personnel moves, new projects, new compliance requirements, to name a few. The justification for pushing complexity down to the OUs is that the AD technologies will accommodate changes at the OU level simply by a drag-and-drop procedure.

Changes to the Domain structure or the Forest structure, however, are much more difficult to accomplish, as Microsoft has yet to provide grafting and pruning tools for their directory and also, any changes to domains and/or forests will require corresponding changes to the DNS infrastructure. Therefore, any design mistakes early in the design process will be difficult to rectify after deployment.

Collectivity the first seven steps shown in the diagram above (white rectangles) will produce a Domain Name space design. **The concept of Domain Name space actually fuses the Forest and Domain Plans with DNS design.**

The Microsoft planning methodology results in four planning documents as shown at the bottom of each column in the diagram above. These documents are the:

- Forest Plan
- Domain Plan
- Organizational Unit Plan
- Site Plan

The following sections will describe each plan. The Forest and Domain Plans are explained in much more detail than the OU and Site Plans, as mistakes made in the forest and domain designs are harder to recover from, as explained above.

3.1 Forest Plan

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

- Single Schema
- Single Configuration Container
- Complete Trust
- Single Global Catalog
- Users Search the Global Catalog
- Users log on using User Principal Names

3.1.1 Forest Planning Process

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

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

3.1.2 Determining the Number of Forests

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

Creating a Single Forest Environment

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

Creating a Multiple-Forest Environment

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

Because forests have shared elements, such as schema, it is necessary for all the participants in a forest to agree on the content and administration of those shared elements.

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

- Network administration is broken into multiple autonomous groups
- The multiple autonomous groups do not trust each other
- Each autonomous group wants individual control over the schema
- The need to limit trust relationships between domains and trees

The consequences of having more than one forest:

- You will have multiple schemas and maintaining consistency between them will create overhead
- You will have multiple configuration containers. Network topology changes will have to be replicated manually to each additional forest, thereby creating more management requirements
- Users will have to explicitly query resources outside their own forest
- Any replication of information between forests will be manual
- You cannot easily move accounts between forests

3.1.3 Forest Change Control Policy

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

Schema Change Policy

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

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

Configuration Change Policy

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

- The name of the team in your organization that controls the enterprise administrators group
- The starting membership of the enterprise administrators group
- Guidelines and a process for creating new domains in the forest

- Guidelines and a process for modifying the forest site topology

3.1.4 Changing the Forest Plan after Deployment

When a domain is created, it can be joined to an existing forest. You can create a domain by promoting a Windows 2000 server to the Active Directory domain controller role, or by upgrading NT Primary Domain Controller to Windows 2000.

Individual objects can be moved between forests. However, the current tools for importing and exporting objects between multiple forests are crude. It is important to remember that two forests cannot be merged in a one-step operation, nor can you move a domain between forests as a one-step operation.

Best Practice # 2 (see Part III). It is important that the forest plan requires a minimum amount of restructuring as your organization evolves.

3.2 Domain Plan

The domain plan is perhaps the most complicated aspect of the Active Directory design process. Microsoft has closely integrated Microsoft Domains, LDAP Directory Services, and DNS. Each of these technologies is complicated; the integration of these technologies exacerbates complexity.

The planning process described below is divided into three parts:

- Determining the number of domains
- DNS and Domain Names
- Post Deployment Change management

There are a few more steps but bullets 1 and 2 above are the bulk of the planning effort. Reducing the number of domains in the forest is on everyone's short list of design goals. DOE sites may require a few more domains than average corporate America's organizations due to organizational structures and the security compliance issues.

The close integration of DNS name space and domain name space (which is actually LDAP name space) is not only complicated, this aspect of AD is also very intrusive to the existing DNS infrastructure. DNS options are described in more detail in Appendix A.

3.2.1 Domain Planning Process

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

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

- Current domain administrators who are responsible for user accounts, groups, and computers
- Teams that manage and monitor the physical networks
- Team that manage DNS
- Security teams"

The steps to creating a domain plan for a forest are:

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

3.2.2 Determining the Number of Domains in each Forest

Three possible reasons for creating additional domains are:

1. Preserving existing Windows NT domains
   "If you have existing NT domains, you might prefer to keep them instead of consolidating them into fewer Active Directory Domains"

2. Administrative Partitioning
   Administration partitioning may be required to support autonomous administration, security, and privacy

3. Physical Partitioning
   There are very complicated "replication" issues with the Active Directory Services." In a nutshell, domains can scale to millions of objects and any domain controller is capable of providing updates, which in turn causes this information to be replicated to all the domain controllers. There are cases where a new domain is justifiable just to control replication traffic.

3.2.3 Choose a Forest Root Domain

See Best Practice # 3 in Part III.

3.2.4 Assign a DNS name to each domain to create a domain hierarchy

Active Directory domains are named with DNS names that are the locator services for the Active Directory. Clients query DNS to locate services such as LDAP and Kerberos Key Distribution Centers. Also, a client uses DNS to determine what site it is in and what site its domain controller is in. The location service is a complicated mix of DNS and LDAP queries.

Associated with this task is the planning of the number of trees. The goal of this task is to minimize the number of trees because each tree requires a separate DNS zone. Additional trees require maintaining a separate DNS zone per tree.

3.2.5 Plan the DNS Server Deployment

Microsoft recognizes that most existing DNS infrastructure is based on Berkeley Internet Domain Daemon (BIND). Bind 8.1.2 does support dynamic updates and also supports service resource records all in accordance to RFC 2136. BIND servers can support the Active Directory; however, Microsoft's strategy of "embrace and extend the standards" has caused Active Directory DNS to be noncompliant with the current DNS standards. AD DNS supports Unicode and the use of the underscore character in their resource records (note, there is a pending DNS RFC that will support Unicode).

3.2.5.1 Background

Windows 2000 Active Directory has integrated DNS name space with their Domain Name space, (which is actually LDAP name space). Novell Directory Services and Netscape iPlanet have not integrated DNS names with any structure related to their LDAP Directory Information Trees.

Microsoft's utilization of DNS name space has made the deployment of AD into established networks a confusing and intrusive task. This "requirement" is probably a reason why industry has been slow to adopt Microsoft's Active Directory Services.

Appendix A, DNS Options, lists some of the design possibilities that are available to the Active Directory designers. Note, the author of this paper has reviewed and tested many other DNS designs (referred to as short cut DNS designs), all with the intent to defeat the AD DNS requirements. These "clever DNS designs" always create problems in some other aspect of the design (see Option # 5 in Appendix A).

Here is a brief description on how the name space integration works. This is an excerpt from the Distributed Services Guide in the Resource Kit. Always keep the information below in mind, when reviewing a DNS "short-cut" design.

> *Every Windows 2000 domain has a DNS name (for example, doe.gov), and every Windows 2000-based computer has a DNS name (for example, talos.doe.gov). Thus, domains and computers are represented both as objects in Active Directory and as nodes in DNS.*
>
> ***Because DNS domains and Active Directory domains share identical domain names, it is easy to confuse their roles***. *The difference is that the two name spaces, although sharing an identical domain structure, store different data and, therefore, manage different objects: DNS stores zones and resource records, and Active Directory stores domain and domain objects. Both systems use a database to resolve names.*
>
> *DNS resolves domain names and computer names to resource records through requests received by DNS servers as DNS queries to the DNS database. Active Directory resolves domain object names to object records through requests that*

are received by domain controllers, as LDAP search requests or as modify requests to the Active Directory database.

Thus, the Active Directory domain computer account object is in a different name space from the DNS host record that represents the same computer in the DNS zone.

The sentence above, (***bold and underlined***), is the technical reality that the DNS designers may try to defeat. Microsoft has designed the AD where the DNS Domain name and the AD Domain name are identical. Measures to defeat this reality can crop up during your design process. If so, insist that the DNS designers produce "reference sites" that are utilizing the proposed design. Also, consider that the entire list of Books in Appendix B will not describe any designs that defeat this fundamental DNS-to-AD Domains naming constraint.

This coincidence of the name spaces is cause for confusion as pointed out above. The complexities are compounded again by the fact that LDAP queries will use DNS to locate domain controllers and AD services.

The main thought to keep in mind when reading through the examples below in Appendix A, is that a client's DNS domain name determines where in the AD a client searches for its resources. This example should offer clarification; client talos.w2k.local has w2k.local as the DNS domain portion of its fully qualified name. When this client conducts an LDAP search, the LDAP query will first search the Global Catalog (GC). If the information is not found in the GC, the client will then search its partition. The DNS portion of its fully qualified DNS domain name, which is the same name as its AD domain (w2k.local) determines its partition.

DNS Option #5 (Appendix A) points out some of the problems of having a client in one DNS domain and its partition in another DNS domain.

3.3 Organizational Unit Plan

OU design and planning is another very complex aspect of the design. However, changes to the design after deployment, are relatively easy to accomplish. A well-designed OU plan will ensure a return on investment for your AD effort.

Executive management should have a support role in this process but they will be more dependent on their technical resources than in the case for Domain Name space Design.

The decisions on OU design, GPO, security groups, and delegation are critical; however these aspects of AD are designed to handle the changes to your directory. Therefore, these design decisions do not represent as great a risk as the more permanent aspects of the design (Domain Name space)!

Best Practice #1 advocates pushing complexity down to the OU. Here are some reasons why complexity should be handled at the OU level.

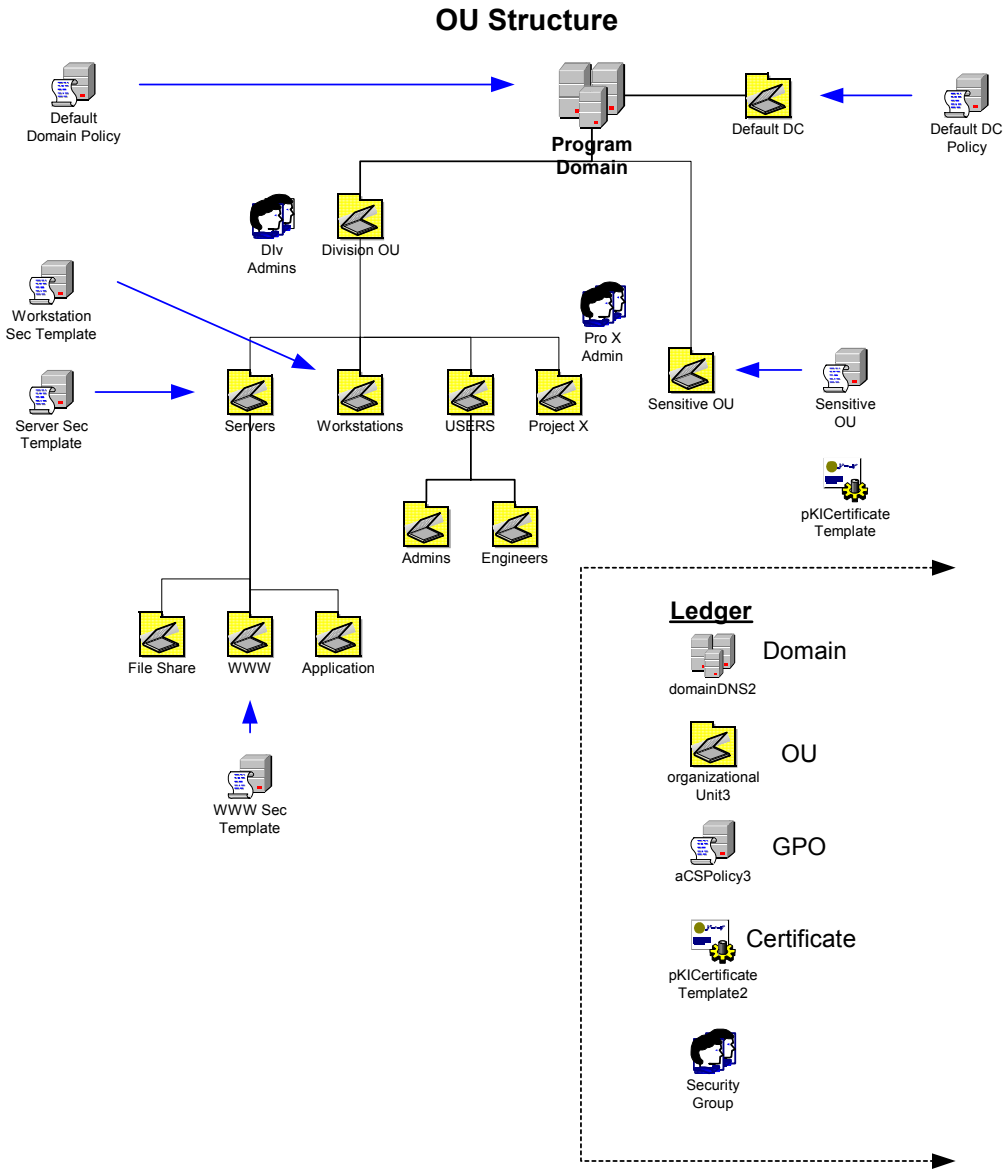
- Changing the OU Structure is fairly easy
- OUs are very flexible when used in conjunction with security groups and Group Policy Objects
- OUs offer a type of security boundary
- GPOs as a parent OU are inherited by a child OU (remember this does not happen at the domain level: a child domain does not inherit policy from its parent domain in the domain name space)
- OUs can be delegated administration rights, thus saving the cost of adding a domain just for administrative reasons
- The initial OU design requirements can be influenced by the down level domain migration requirements. The OU infrastructure can be redesigned after the migration

The flexibility of the OU also leads to many complicated design scenarios. As a sub component of a Domain, it follows that each Domain will have different criterion for its OU design. (Domains that are migrating from NT 4 will have additional considerations for its OU design).

Below are some general design guidelines and descriptions of the potential use of the OU. **There is however an important best practice** (Best Practice #10, Part III).

> "The best approach that you can take with OUs is to create them based on your IT administrative structure and not on your organization's management structure (or any other structure, for that matter)." The above is an excerpt from Microsoft's AD technical Reference, but you can find similar statements in most of the works listed in the bibliography.
>
> There are some other reasons why you might want to create OUs, such as to use them as Group Policy boundaries. Since OUs can be nested into many levels, there are a number of reasons why you might create additional (nested) OUs to make administration of Windows 2000 and Active Directory easier.

The figure below is a fictitious Domain with its associated OU. The narrative description that follows is a description of each OU (why it was created), which also explains the types of administrative delegation where applicable, what GPO's are applied at the OU, and what security templates are applied to the computers within the OU. (Security templates are registry settings to the local computer. The templates are applied either as a local policy on a specific computer or at the domain level by a GPO. Computers that require different security templates should be placed in separate OUs).



The diagram above depicts a typical domain with its OU structure. Also noted are the Group Policy Objects. Starting with the domain you can see two GPOs, the default domain policy and the default domain controller. All the domain controllers are located in a separate OU and the GPO is specific to the DCs.

This domain has multiple divisions but only one divisional OU structure is shown. The entire divisional OU structure inherits the default domain policy. The divisional system administrators operate independently from the domain administrators; therefore the domain administrators have delegated “full control rights” to the Div. Administrators security group.

The administrators of the divisional OU have created four OUs directly under their divisional OU. Below are descriptions of each OU.

- All division servers are located in the servers OU and its child OUs. A separate GPO is applied at the server OU, to enhance the security and functionality of each server. The web servers located in the WWW OU has an additional GPO, which contains the Security Templates specific to the Internet Information Servers and prevents any other services from running on the IIS servers.
- All workstations are located in the workstation OU. For safe operation of workstations. The workstation GPO is applied to provide software distribution and security templates. Settings in response to vulnerabilities (ISS Scans) are included in the security template and are applied to every workstation in the OU.
- All the division's users are located in the USER OU structure. The default domain policy (inherited) is adequate (account policy, account lockout policy, and Kerberos policy). There are two other groups of users that have special requirements. The engineering group has special software requirements and also need extra privileged systems rights to develop their software. The divisional OU administrators also have a separate OU (this group's distinction is obvious). Not shown in the diagram is separate GPO and security templates for each of these child OUs.
- The project X OU contains the infrastructure necessary for work on this sensitive project. All user accounts and computers that are associated with the project are located in the OU. Full control administrative rights have been delegated to the Proj X security group. This OU may have a hierarchical structure below it but these OUs are hidden from the Active Directory.
- The last OU is the program's sensitive OU that can protect "critical computer systems."

3.4 Site Planning Process

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

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

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

- Define sites and site links using your physical topology as a starting point. (Site links are connection objects, used to connect two sites, which are normally connected as a Wide Area Network)
- Place servers into sites

- Understand how changes to your site topology after deployment will impact end users

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

- Teams that manage and monitor the TCP/IP networks
- Domain administrators for each domain in the forest

Warning on Applying Group Policy at the Site Level
Group Policy will flow down a hierarchy in the following order:

- Site
- Domain
- OU

The bulleted list above shows that the domain, then the OU, and lastly the local computer policy, will inherit policy that is applied to a site. Some designers will think it natural to apply all security policies at the Site. This way all domains within a site will have the same policy. This practice is not a good idea for the following reasons:

1) Best Practice Number 1, states that complexity should be pushed down the hierarchy. Multiple sites increase the complexity at the top of the hierarchy
2) An incorrect GPO applied at the site level will break every computer at the site
3) Site Level GPO will make troubleshooting GPO very difficult
4) The Active Directory Domain has a separate OU for domain controllers. A separate and special GPO is applied to the domain controllers OU. Policy at the site level will override theses settings

Consider also, the following excerpt from Jennings (Windows 2000 Group Policy, page 11)

> "You can apply Group Policy at the site level, but it is more common to establish a basic set of policies on a domain-wide basis and then establish policies that apply to individual OUs. The primary use of site-level Group Policies is to specify different servers to store redirected folders, roaming user profiles, or both, depending on the client's site membership.
>
> Another reason to start at the domain level is that domains have a Default Domain Policy, and sites don't have a Default Site Policy."

Security templates and OUs can be deployed for the purpose of associating computers that require a unique and custom Security Template. Any setting generated by GPO, or Administrator Templates that change the registry of any computer, should be applied directly to the OU that houses the computer account (Security Training Guide page 271).

Here is another reason for not applying many policies at the site level. Many policies are domain concepts. Lowe-Norris lists domain centric policies on page 112.

- Password Policies, such as password length, password expiry interval and so forth
- Account Lockout Policies
- Kerberos policies
- Encrypted file system recovery policies
- IP security policies
- Public Key encryption policies
- Certificate authorities

When designing Group Policy, always test your design on a pilot network. What makes sense at first glance may be a total disaster!

Replication and Query Traffic
As stated above, site design effects query traffic and replication traffic. The subject of replication is very complicated (second to name space design). There are hard limits to the number of domains and sites the Knowledge Consistency Checker can handle. Trouble shooting replication problems is not easy and increasing the number of sites makes the replication topology more complicated. Most DOE sites have a high bandwidth backbone that translates into a very well connected campus. Below are two excerpts on the subject of breaking up a well-connected collection of well-connected TCP/IP sub networks.

> From Lowe-Norris page 163, "Remember that a site is a well-connected set of subnets (well-connected tends to mean about 10-MBps LAN speed). A site does not have to have a server in it; it can be composed entirely of clients. If you have two buildings, **or an entire campus that is connected over 10.100-MBps links, your entire location is a single site."**
>
> Lowe-Norris continues on page 165. "To summarize, I would suggest that, by default, you create one site per 10-MBps-or-higher location, unless you have an overriding reason not to do so."
>
> From Rand Morimoto (Windows 2000 Design and Migration, page 150) under the bold section heading:
>
> "**Don't Divide Well-Connected Segments into Multiple Sites"**
> "It's not a good idea to create multiple sites on a well-connected network. By dividing well-connected subnets into multiple sites, you can actually decrease the performance."

4.0 Scope of AD Design

Directories are a fundamental change to Information Technology design and management. The limitations of previous technologies, such as NT4 domains and UNIX's NIS and NIS+, would actually dictate organizational operations. Lightweight Directory Access Protocol (LDAP) directories can actually accommodate organizational

processes and even facilitate Business Process Reengineering. The potential rewards and risks of designing and deploying a directory are enormous.

The promise of LDAP based directories with its inherent ability to consolidate and disseminate corporate information, is achievable by "General Purpose Directories." The design and deployment of a General Purpose LDAP Directory requires cooperation and buy in from the top executives, divisional management, IT management, and end users. The development of new "directory aware" applications is central to directory planning. New LDAP applications are the vehicles that can lead to Business Process engineering (see Reed).

The formal design process of a general purpose directory involves the business justification, total cost of ownership issues, return of investment cycles, restructuring of the IT department, and so forth. **The AD design process does not necessarily have to include these high level functions.**

In spite of Microsoft's marketing efforts, Active Directory (AD) is not considered a general-purpose directory. Active Directory is a Network Operating System (NOS) directory. What this means to the scope of an AD design is that the return on investment (ROI) can be calculated by the savings in the cost of IT management and the cost savings of security compliance. It is true that future releases (Server.NET) of AD will move toward a general-purpose directory, but currently there are very few AD aware applications that could lead to the streamlining of business processes.

Designing a NOS Directory such as AD simplifies the design process, especially at the corporate application levels. You shouldn't need to include the consolidation of all current directories into the Active Directory as part of your design process. Today such planning is best left to other technologies such as Meta Directories (see Burton Group in bibliography). However, even omitting directory consolidation from the design, the design of and the design process of an Active Directory is still a very daunting and complicated task.

With all of Active Directory's shortcomings, the benefits of a good AD design are very valuable to all DOE sites. Security compliance issues have greatly raised the cost of ownership for all DOE computers. The time required to deploy a fully compliant Windows 2000 or NT4 workstation has been stated as 45 minutes to a full hour for each computer. **A proper Active Directory design can automate most of these compliance issues, saving thousands of hours of administrative time per DOE site**. As new vulnerabilities are discovered, "hot fixes" and patches are produced to close these security holes and an administrator has to visit each and every computer to apply these fixes. **With Active Directory, these "hot-fixes" and patches can be deployed to an entire DOE site from just a few locations, or even from a single location.**

A well-designed and timely deployment of an Active Directory can ease the time effort of security compliance and the IT staffs can rededicate their efforts of facilitating the programmatic mission.

Part II: Active Directory Design Scenario

5.0 Description of Hypothetical Site

This section provides a general discussion on the issues and trade-offs of Domain Name space design. The following design will be examined from the perspective of a typical DOE research and production site, LCIS. LCIS' design team worked through the design in the following order.

- Design 1: Single Forest with a Single Domain
- Design 2: Single Forest with Multiple Domains
- Design 3: Multiple Forests

Section 6 will describe some of the issues that LCIS faced to accomplish their Active Directory design. The design team needed to answer the following questions.

- How many Forests?
- How Many Domains?
- What is the best DNS Design for the Domain Name space? (see Appendix A, DNS Options)
- What are the Security verses Ease of Management Tradeoffs?

LCIS did accomplish a design that is documented at the end of Section 7.

Before describing the LCIS site and the programmatic requirements for the Active Directory design, the next section will generally explain what is involved with this part of the design process.

5.1 Pragmatic Discussion of Forest and Domain Planning

During the planning process, expect a debate to cover the entire range of forest architecture from a single forest with a single domain, to a separate forest for each and every organization at your site! Expect these debates to be heated as Microsoft's marketing has overstated the case of a single domain as the optimal design, when in reality most AD deployments have forced organizations into multiple forests! Quoting from the Butron Group Paper "Microsoft Active Directory: Not Perfect, But Good Enough for Specific Roles," page 12.

> "When Active Directory arrived with Windows 2000 Server, Microsoft's efforts oversimplified deployment issues by consistently implying that companies should and could deploy a single forest. Deployment experience has proven otherwise…"
>
> Note: The paper also implies that deploying a single forest for an Enterprise is achievable if the AD is designed as a "Server Directory" (NOS) and not an Enterprise Directory (or General Purpose Directory; see Burton Group).

A good percentage of DOE AD designs should be achievable via a single forest. The number of domains within a single forest is yet another design aspect, which will require much time and debate. Here again Microsoft has pushed a single domain as the optimal model. Many large sites will find it difficult to implement a single domain without losing some utility of the Directory. Here is a quote from Olsen, page 126.

> "Start with a single domain, and then prove you need more. In reality, there will be very few situations other than in a small, single location office that will successfully implement a single domain, but that's where to start."

While there are many possible forest/domain designs it will be instructive to compare three designs. (Note: if more complex and convoluted designs are advocated by designers, make sure to ask them for references or produce the design from the written word, like books of magazine articles).

1. Single Domain (single forest of course)
2. Single Forest with multiple domains
3. Multiple Forests

We will step through the design tradeoffs and compare each design. For each design we will discuss the technology aspects and where applicable we will view the aspects of the design from the following four perspectives.

- **DNS name space**, and the organization that manages DNS. Forests and number and location of forests and domains may also require changes to the process of registering the computers at your site. Engage the DNS managers and technologists very early in the design process.
- **Programmatic End Users and ease of network use**. Training users where and how to search LDAP space may or may not be an issue. More forests will lead to user confusion not only from the LDAP perspective, but also on the DNS name space (see Appendix A, DNS options).
- **System management**. Ease of system management verses autonomy of departmental systems and system use will be an issue. A separate forest for each department represents maximum autonomy but greatly increases the cost of management of these systems. Here is an example of this trade off; if each department is managing a private name space forest (see DNS Options, Appendix A), each department must have in-house DNS experts to maintain their autonomy!
- **The security perspective**. Can a single forest containing a single domain provide the proper level of security in a need-to-know environment? Remember, one of the main advantages of LDAP and Active Directory is that more information is accessible to more people. Security requirements will, in the end, influence the number of forests and domains for each and every DOE site!

5.2 LCIS Design Requirements

Section 7 analyses the design decisions of a fictitious Department of Energy laboratory/production site. LCIS (Laboratory of Cool and Interesting Stuff) is a typical research and production DOE site, and provides research in the area of environmental safety. Also, there is a large engineering department that supports all the research and production. The production department supplies sensitive products to many government agencies. There are various support organizations and business functions such as Management Information Systems, payroll and legal.

A DOE site will have specific security requirements. Many of the research organizations will have sensitive information that must be confined to specific projects within the department. The production facilities will have sensitive information on process, inventory, and cost. While these security requirements may not be as restrictive as classified computing networks, they are generally more restrictive than a typical commercial concern.

The comparative analysis of three designs below in Section 7, will include LCIS's unique security and compliance issues and requirements. These requirements may not necessarily be restricted to the DOE orders; the programmatic management typically dictates the requirements for each program. It is expected that the physics and production departments will have stricter security and privacy requirements than the environmental departments. Legal and payroll will also have unique security and privacy requirements.

5.2.1 Programmatic Requirements

1. Physics Department (PD)
   - Some of the information on their network is very sensitive and Physics requires absolute control of this information. PD must identify and protect its "critical computer systems."
   - PD collaborated with different departments, such as engineering and scientific computing. Some of these collaborations involve sensitive information. The collaborators are located in their own departments.
   - PD personnel must be able to network to Payroll to fill out their timecards.
   - PD also provides information that is not sensitive to various departments within the complex.
2. Environmental Research (ER)
   - Ascertains information from most of the scientific programs within the complex.
   - Currently, ER has no sensitive information on their network and has no critical computer systems.
   - Personnel must be able to get to the Management Information System (MIS) for time cards, vacation, etc.

3. Engineering (Eng)
    - Supports most of the scientific projects and has access to sensitive information from most scientific departments. Must identify and protect "critical computer systems" in accordance with the regulators.
    - MIS Requirements (time cards, etc.)
4. Payroll, Accounting, Human Resources (MIS)
    - Must provide many on-line services to all LCIS personnel.
    - Stringent security, privacy, and integrity requirements.
5. Legal Department (LD)
    - Network has very sensitive legal information and any form of disclosure of this information can lead to a law suite against LCIS.
    - LD does not collaborate or share any information with the scientific departments.
    - LD personnel must have access to the MIS systems, (timecards).
6. Internal third world networks (ITW)
    - It is estimated that LCIS has over 200 existing NT4 Domains and Workgroups. It is also believed that most of the small domains have inadequate network management. It is a requirement that the AD design will consolidate most of these domains/workgroups to ensure consistent security policies and compliance audits.

6.0 Comparison of Three Design Approaches

The next three sections will provide information for each of the three designs:

- Narrative description of the characteristics and features of the design
- The benefits of the approach
- The shortcomings
- Security aspects and implications
- Administrative model (Central, Distributive, Hybrid; definitions are provided in the appropriate sections)
- Perspectives of the 4 identified groups listed in Section 6

There is a lot to cover, and therefore some of the discussion will be brief. I will point to the page number and book (listed in the bibliography), so the interested reader can research to their satisfaction. I will also paraphrase and quote authors to stress certain points, as some of these points can be considered subjective.

The following analysis of the three most common designs is provided as a road map to the issues of AD design. Understanding this information should facilitate your design. However, the design of Active Directory is a huge exercise of technical, organizational, managerial, and security tradeoffs, so do not make design decisions on this paper alone. You must go through the process with a team that represents many different organizations of your enterprise.

6.1 Single Domain

Most design methodologies advocate starting with a single domain and justifying any additional domains. As a review of the section on the planning process, a list of extra domain justification is listed below:

- Preserving existing Windows NT domains
  "If you have existing NT domains, you might prefer to keep them instead of consolidating them into fewer Active Directory Domains"

- Administrative Partitioning
  Administration partitioning may be required to support autonomous administration, security, and privacy, see Appendix, "Leveraging Group Policy for Programmatic Efficiency."

- Physical Partitioning
  There are very complicated "replication" issues with the Active Directory Services. In a nutshell, domains can scale to millions of objects and any domain controller is capable of providing updates, which in turn causes this information to be replicated to all the domain controllers. There are cases where a new domain is justifiable just to control replication traffic.

The above list is taken directly from the "Deployment and Planning Guide." Below is a list taken from various authors and AD designers. Most of these points are taken from the three categories above.

- Unique security requirements
- A program, such as weapons, does not trust administrators from other organizations

Since a single domain is a single forest, we must account for the justification of a single forest and below is a list of reasons to deploy multiple forests:

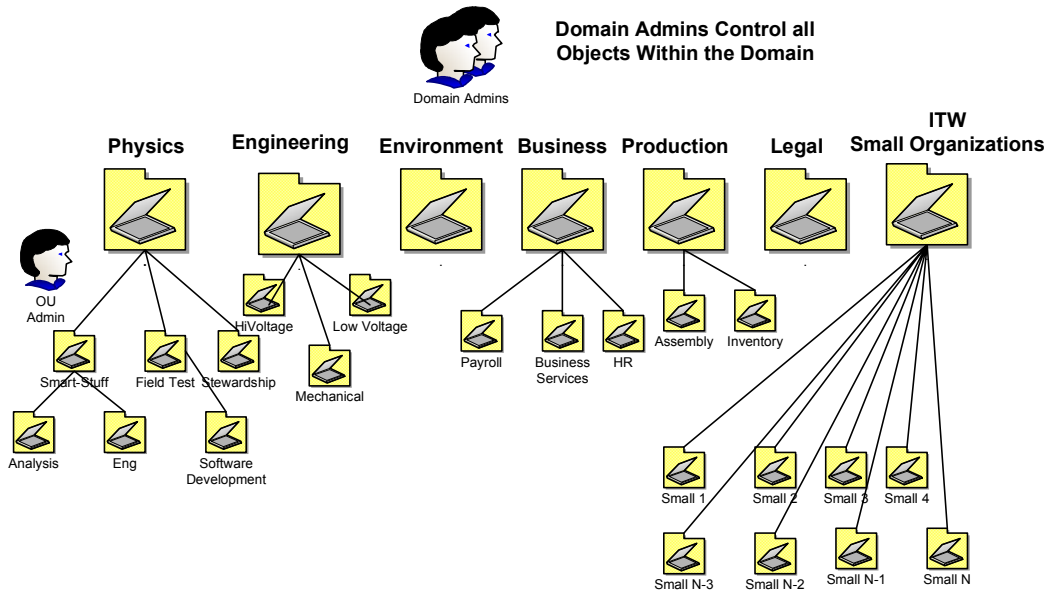
- Network administration is broken into multiple autonomous groups
- The multiple autonomous groups do not trust each other
- Each autonomous group wants individual control over the schema
- The need to limit trust relationships between domains or domain trees

Keep the lists above in mind as you read the description of a single domain deployment below.

6.1.1 Single Domain Design Description

This design consists of a single domain for the entire laboratory. Each department, program, support division, and business department is provided an organizational unit structure. The domain administrators have delegated each department full control of their respective top level OU. Each department's OU administrators can add additional OUs in

a hierarchical fashion. OU administrators from one department have no authority or administrative rights in any other department's OU. However, all the departments must trust all of the domain administrators, as they have total control over any object in the domain. The following diagram is a depiction of **LCIS's domain**.



6.1.2 Single Domain Benefits

This is the simplest of the AD designs to manage, as long as your IT administration is organized centrally. Here is a list of system management benefits:

- All Group Policy applied at the domain level is consistent throughout the organization.
- Microsoft's current administrative tools will allow most system management performed via drag-and-drop tools. As an example, if a user transfers from Weapons Department to Engineering, the Domain Administrator merely drags and drops this account into the Engineering Domain.
- Since a single domain has all the objects for the domain, the information in the Global Catalog is less of a design issue.
- Very easy model for end users to understand and use.
- Least intrusive model to existing DNS infrastructure. DNS managers will love this model as all they have to do to their current DNS infrastructure is delegate out the Microsoft Services sub-domains. See Option 4 in Appendix A, DNS Options.
- A single domain can also save hardware and server license costs.

6.1.3 Single Domain Draw Backs

- Physics, Engineering, and Production must trust the central Domain Administrators.

- Legal and Privacy issues could prevent the Legal Department from joining.
- Distributive File System (DFS) is a new network file system that allows an abstraction of user data. The AD can publish a share point independent of the physical computer that contains the actual data. DFS also offers hi-availability and load balancing for network information. Unfortunately, any domain controller can only be the "root" of one and only one DFS share. Therefore, departments that have control of an OU only cannot leverage DFS shares.
- A single domain requires a very stringent disaster recovery plan. (A multi domain forest with an "empty root domain" will have a much simpler disaster recovery plan (see Section 6.2 on multiple domain design).
- An Active Directory Domain is the fundamental security boundary. Any department OU will not have autonomy and control of the security of their resources.
- A single domain can only be in native mode or mixed mode. Switching to native mode as soon as possible is desirable from a security standpoint. (Kerberos is much safer than NTLM). When does a single mode design cutover to native mode? Many of the small networks may not have the budget or resources to upgrade to native mode, and therefore departments with critical systems will not be able to protect their critical systems until all the "Windows computers" at LCIS are upgraded to Windows 2000/XP.

6.1.4 Single Domain User Perspectives

- DNS managers. The single domain is by far the easiest infrastructure to maintain. DNS managers will not have to plan for new DNS Domains in response to new AD domains.
- Departmental Users. A very Simple model for searching for information; this approach has the least amount of confusion for end users.
- System and Network Managers. A single domain model does not work well for a large organization with a distributive system management model. It is easier for departments with complex operations and security requirement to have autonomy over their resources.
- Security managers. Most departmental security managers who have responsibility for securing "critical systems" will require their own domain at the least and possibly a separate forest.

6.1.5 Single Domain Concluding Remarks

There have been successful single domain deployments at large sites. Leicester University (see Lowell-Norris) has deployed a single domain. Universities have some sensitive information (grades come to mind) and Leicester has trusted the Domain Administrators with control of all the information in the Domain. Lowell-Norris favors a single domain deployment; however, he states that the following concepts are best implemented in a per domain fashion, (see page 112 of Lowell-Norris).

"Here is a list of what types of settings can be set only on a domain-wide basis."

- Encrypted File System Recovery Policies

- IP Security Policies
- Public Key Infrastructure Policies
- Certificate Authorities
- Password Policy
- Account Lockout Policy
- Kerberos Policies

All of the features in the Lowell-Norris list above can be utilized to protect "critical systems." He is stating that these utilities are domain concepts.

LCIS concluded that a single domain works for the Environment Department and most of the small networks in the "Third World Network." However, Physics, Engineering, Business Systems, Production, and Legal should have autonomy to manage their security.

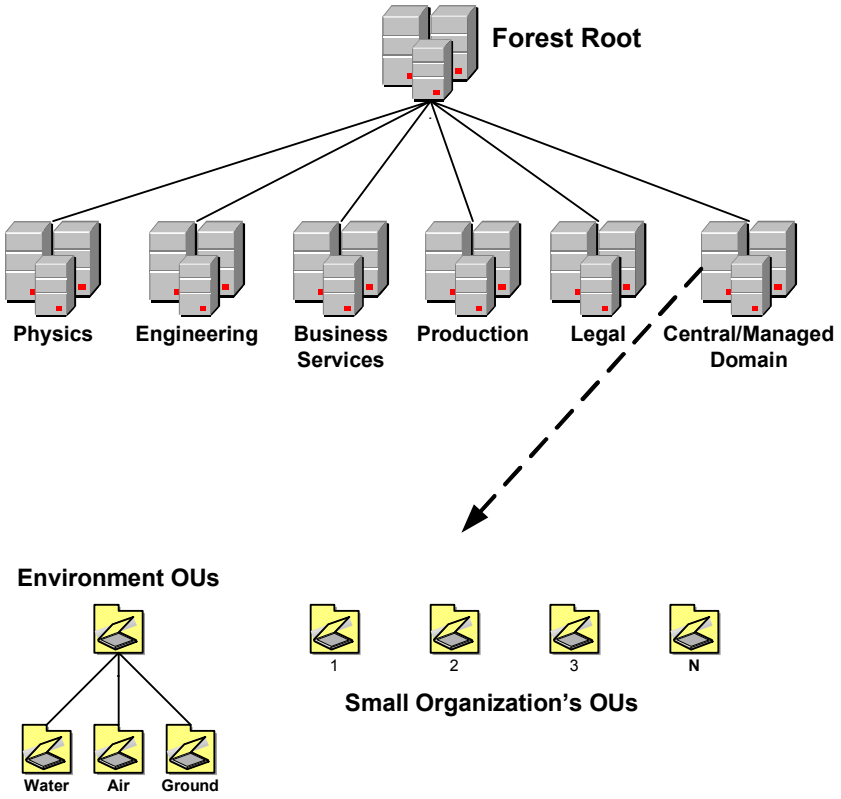
(Note: CIAC is currently working on "How to" papers for protecting critical computer systems within an Active Directory Forest. These papers will provide theoretical information as well as how to deploy, EFS, IPSec, PKI, CA, and DFS).

6.2 Multiple Domain Model

LCIS has followed the recommendation to start the design with a single domain and justify any additional domains. Physics, Business Services, Engineering, Production, and Legal are not convinced that they can protect sensitive information and/or critical systems by managing an OU (as delegated by domain administrators who do not directly report to them).

The "multiple domain model" will provide the departments with a security boundary (according to Microsoft, a "domain" is a security boundary). However, all the departments have expressed concerns as to the scope of power for both the Enterprise Administrators and Schema Administrators. These two groups have powerful rights and privileges throughout the Forest.

The diagram below shows LCIS' multiple domain architecture.



6.2.1 Multiple Domain Description

The following features characterize the LCIS Directory Design Team's multiple domain design:

- Each autonomous department with sensitive information and/or "critical systems" has the option of controlling their own domain. These departments are listed below:
  - Physics
  - Engineering
  - Business Services
  - Production
  - Legal
- The system administration model for LCIS is distributive, therefore, the departments that currently have their own system management team will retain these teams.
- Any department, program, or business unit that does not have critical systems or sensitive information will be provided a top level OU within the Central/Managed Domain (see the "folders" in the diagram above).
  - Environment
  - All other small NT4 domains and Windows workgroups
- The Central/Managed Domain will provide system services to these departments in a centralized fashion.

- The Forest Root will contain no unnecessary computer accounts, user accounts, or applications. This is an "empty root" model (see the section on benefits below).

6.2.2 Multiple Domain Benefits

- Each department has control of their security as each department's Domain Administrator is a direct report (or Matrix) to the department, and is directly accountable to the departmental senior management.
- Each department can design and implement the following security features and functions on a domain level.
    - Public Key Policies
    - IP Security Policies
    - Encrypted File System Recovery Policies
    - Certificate Authority
    - Password Policies
    - Account Lockout Policies
    - Kerberos Policies

  Proper design, implementation, and control of these security functions can lead to reasonable security to "critical computer systems."
- Each department has control of the authentication and authorization with its collaborators.
- Departments can choose when they will cutover to "Native Mode." They do not have to wait for financially strapped or mismanaged departments to upgrade, and therefore can leverage the security of "Native Mode" as soon as possible.
- Each department can selectively remain in "Mixed-Mode" until such time they can afford to upgrade and migrate to Native Mode. The time spent planning a migration can reduce the risk of migration.
- The empty forest root will reduce the number of Enterprise Administrators (any Domain Administrator in the Forest Root is an Enterprise Administrator). This could help facilitate managerial controls on these two powerful accounts.
- The "empty" Forest Root will allow easier and faster recoveries from disasters. Also, limiting the Forest Roots' objects will keep this domain very small and almost static. This leads to very little replication traffic for the forest root; it is therefore easy to install Forest Root Domain Controllers in various (secure) locations throughout the site. This measure will help prevent the network from becoming a single-point-of-failure.

6.2.3 Multiple Domain Draw Backs

- Each domain must purchase multiple domain controllers. This could manifest itself as an overall increase to LCIS IT budget (Domain Controller's licenses and hardware can get expensive).
- Each domain must have at least two well-trained system administrators. These administrators must have or develop skills that are not historical to managing Windows environment, e.g., DNS, and/or IPSec. This could lead to an increase to the cost of IT management throughout the institution.

- Moving user accounts from domain to domain is much more difficult than moving the same objects from OU to OU.
- Achieving this architecture will impact the current DNS Infrastructure and possibly inventory process. If system registration and inventory are provided by in-house applications, these applications will have to be modified.
- Each domain must have a separate disaster recovery plan.
- LCIS must create, publish, and socialize standards for "New Domain" justification.

6.2.4 Multiple Domain User Perspective

- **DNS Administrators.** The DNS aspect of multiple domain architecture is very complex and very invasive to the existing DNS Infrastructures. Today, most DOE sites' DNS servers are running on UNIX systems and wish to keep their stable DNS structure. Microsoft's DNS requirements may be viewed as invasive to the existing stability. Expect resistance from the DNS managers. It is critical that the AD design team engage the DNS management team as early as possible in the design cycle. (Note, the DNS issues are a major reason why enterprises are late in deploying Active Directory).
- **Departmental Users.** Expect some confusion on the part of the "typical user." They may wonder why their PC is pc1.physics.lcis.gov, where their UNIX computer is unix.lcis.gov. However, this model still has a single Global Catalog, and therefore searching for directory information is as easy as the single domain model.
- **Systems and Network Managers.** The domain administrators will have a sense of empowerment and autonomy. The domain is a natural administrative and security boundary. The system administrators of the centralized domain (a catch-all domain) will have to coordinate their administrative policies and group policies with each OU administrator also. Disaster recovery planning and test will require a coordinated effort.
- **Security Managers.** They have much more control over the security of their critical systems with their ownership of an autonomous domain. They can identify and protect critical systems and sensitive information without coordination of a domain admin, as would be required if managing security at the OU level only. Some security managers have expressed concerned with the power of the Enterprise Administrator and Schema Administrator accounts, as the Enterprise Administrator can take ownership of any object in the entire forest and the Schema Administrator has control to all schema modifications.

6.2.5 Multiple Domain Concluding Remarks

Currently department managers are responsible for the security of their system and the protection of sensitive information stored on these systems. It would seem that the security requirement alone could justify a domain.

Some of the additional costs of managing multiple domains is due to the lack of tools that should be provided by Microsoft. Microsoft has promised "grafting and pruning" tools for managing multiple domains, and "Resultant Set of Object Permissions" (RSOP) tools

for managing and analyzing the security of the domains, trees, and forests. Microsoft's Active Directory Programming Interface (ADSI) facilitates scripting solutions for problems such as migrating users from one domain to another. Also expect better domain-to-domain management tools from Microsoft and third parties.

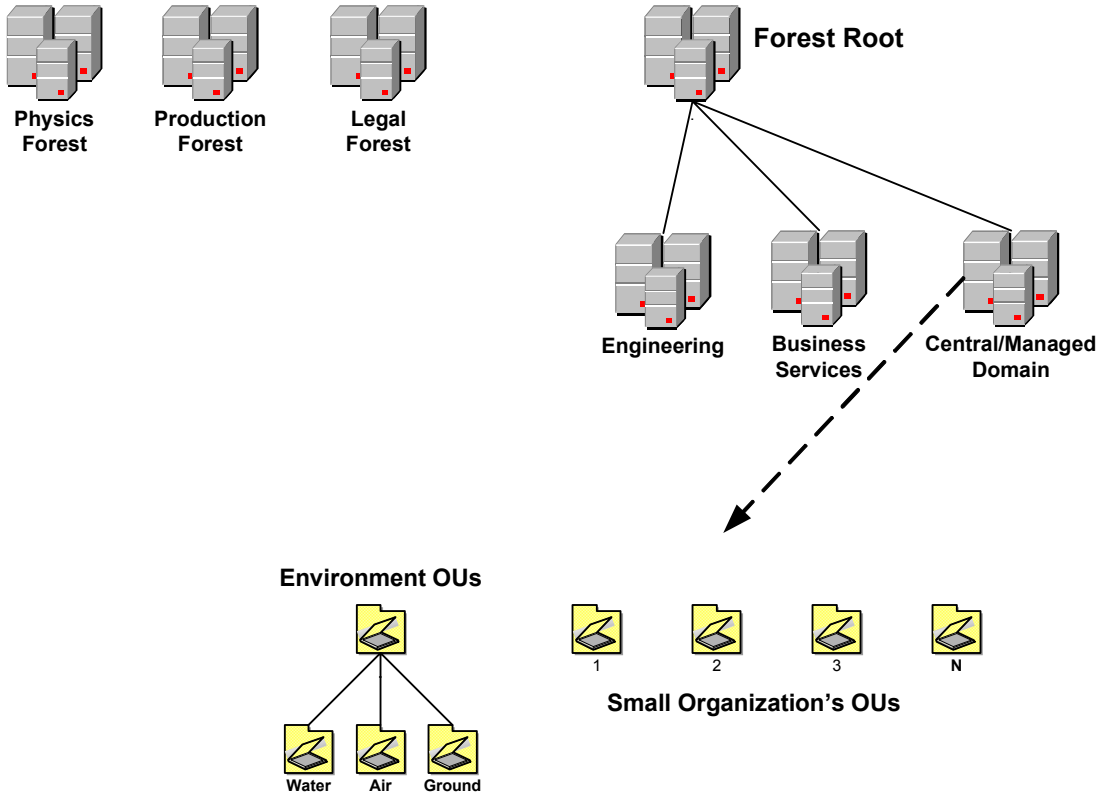
The integration of Microsoft domains to DNS domains creates a very large hurdle to accomplish a multiple domain implementation. See Appendix A. DNS Options.

6.3 Multiple Forests

The multiple domains with a single forest seemed like a good balance between departmental security and ease of sharing information between departments (the DNS managers did not think so!) It surprised many of LCIS' upper management when Physics, Production, and Legal demanded their own forests! These three important departments claimed that a single forest was not adequate assurances against the Enterprise Administrator, Schema Administrators, and the potential of a Group Policy being applied at the site level. They quoted page 22 of "Hacking Exposed: Windows 2000" (Scambray).

> ***"The boundary of security in Windows 2000 is the forest, not the domain as it was under NT."***

The diagram below depicts this situation.



Now there are four forests to implement and support. Is there any real justification for such a drastic move by Physics, Production, and Legal departments? Consider the quote below from Iseminger page 127(AD Technical Reference).

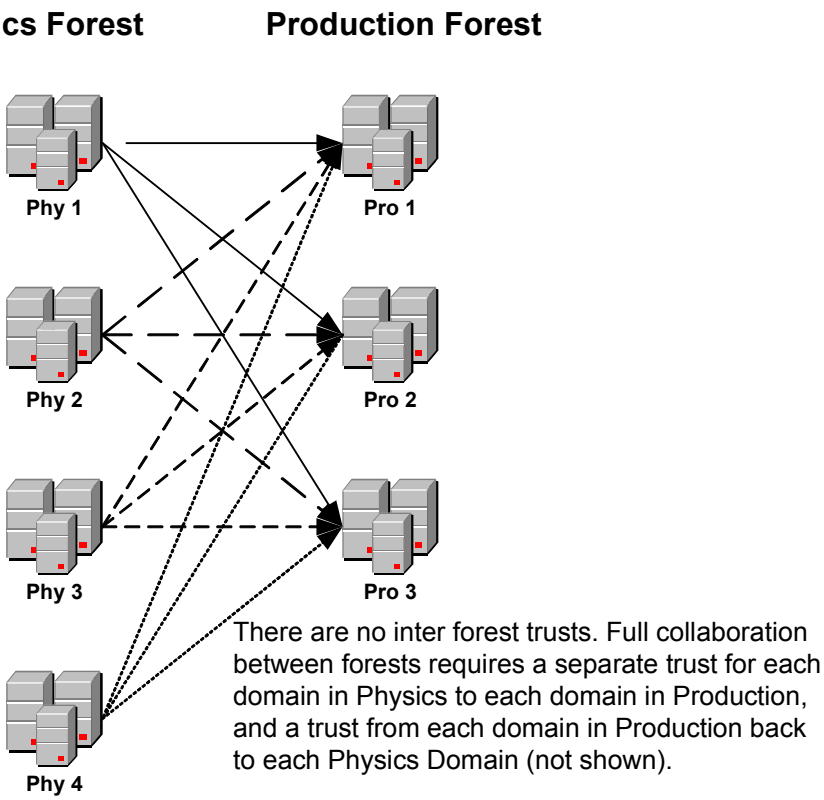
> "Real World"
>
> The same reasons I've identified for not having more than one forest might be perfectly good reasons for you to create more than one forest. If you have reasons for keeping certain users from viewing certain resources, or if you have a very segregated organizational structure in which trust relationships must be separated or private, a multiple-forest environment might be just the thing you've been looking for. Every deployment is different, and turning what I've described as drawbacks into great tools for privacy and security can be as simple as changing your perspective (your security-minded perspective, that is).

Well, in contrast to what Microsoft marketing and training espouses, there seems to be real justifications for multiple-forests.

6.3.1 Multiple Forest Description

LCIS' main forest still contains some major departments and most of the small departments. However, two major programs, Physics and Production, now have separate forests. Legal also has a separate forest but they do not have much collaboration with internal departments.

No doubt that Physics has enhanced its security potential but Physics will now find it much more difficult to share information with its collaborators. Let's examine the trust requirements for Physics to collaborate with Production. Assume also that Physics has implemented 4 domains within their forest and Production has deployed 3 domains. Since the current AD technology does not provide inter forest Kerberos Trusts, these trusts must be created explicitly. Also, the explicit trusts are NTLM trusts and therefore are not transitive and are unidirectional, which leads us back to the unruly NT4 trust model. See the diagram below.



6.3.2 Multiple Forest Benefits

- Total control of departmental security
- Control of the Schema
- Enterprise and Schema Administrators now report directly to the department
- Total control of information published in the Global Catalog
- Does not automatically have a trust relationship with all organizations at LCIS
- Save

6.3.3 Multiple Forest Drawbacks

- The cost of administration has increased
- The complexity of DNS has greatly increased
- Creates islands of technologies (large islands)
- Makes collaborations very difficult
- Complicated forest disaster recovery
- Explicit NTLM trusts are not as secure as Kerberos Trusts
- If institutional applications are rolled out that requires a schema change, each forest must perform the modifications
- The explicit trusts can be difficult to manage and troubleshoot
- At some point, the deployment of multiple forests defeats the concepts and all the benefits of directories
- Save

6.3.4 Multiple Forest User Perspectives

- **DNS administrators.** They are really alarmed at the prospect of 4 distinct and separate DNS name spaces. They are still not sure how they would support the single forest multiple domain models. This is the DNS administrator's worst-case scenario.
- **Departmental Users.** They are confused about how to search the directory for resources. It is likely that many resources they need are not located within their forest. This forces the users to learn how to search multiple global catalogs.
- **System Administrators.** They appreciate the power the have over their forest. Enterprise Administrator is the ultimate AD account. They will soon be faced with the collaboration network trusts. They also feel they need the highest levels of training in DNS, networking, scripting, and security.
- **Security Managers.** They have control over the security of their forest. However, they feel pressure from the internal scientists and managers to better support their mission and to especially streamline the collaboration process.

6.3.5 Multiple Forest Concluding Remarks

Earlier I explained the Burton Group's perspective on deploying a single forest. Recall that Burton Group claimed that Microsoft earlier marketing efforts definitely advocated a single forest and a single domain. Maybe The Burton Group was right considering that Microsoft's next release of AD (Server.Net) provides transitive inter forest Kerberos trusts. (Did Microsoft advocate a single forest/domain deployment because this is the least intrusive model to the existing stable DNS infrastructures? And what model will Microsoft advocate when they finally ship the necessary tools to manage a directory? Stay Tuned).

As we worked through our three designs it seems that we can make a general rule:

> "As the need for autonomy and security increases, the complexity of system management increases, and the impact to DNS increases."

This statement should not be considered profound; no it is merely a hint to achieve a balanced design.

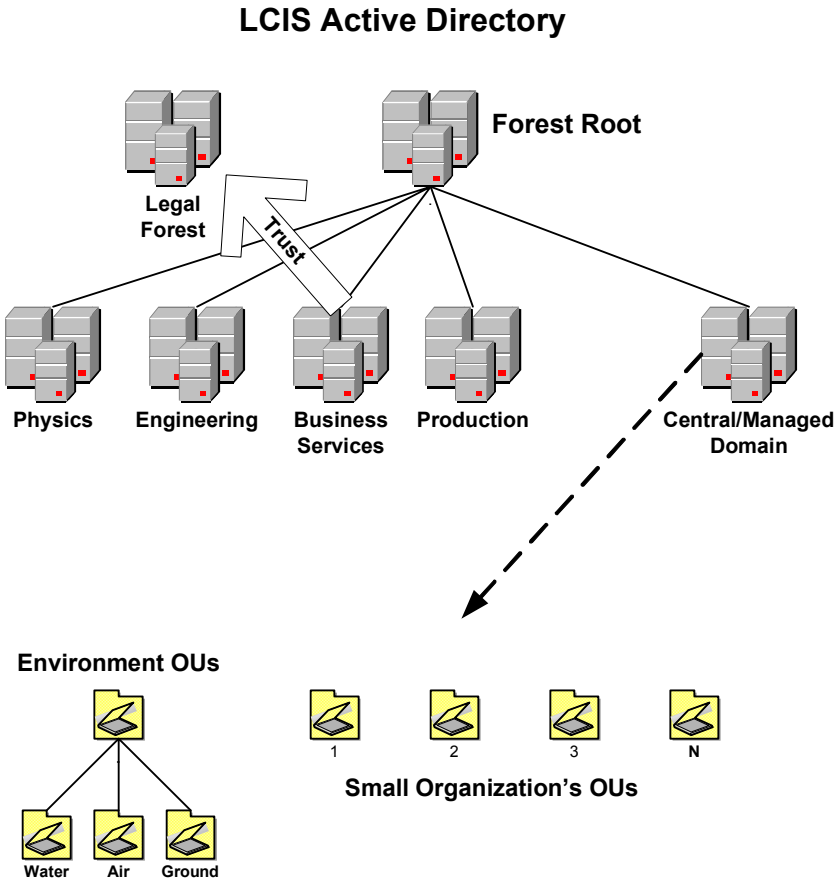
From the design discussions above, let's briefly recap where we are.

- **Single Domain**. Simple, yet there does not seem to be enough administrative and security autonomy.
- **Multiple-Domains.** More complex, more security autonomy, all but the most security conscious favors this model.
- **Multiple Forests.** Very complex to use and manage. Only favored by the departments who do not trust any other department.

The next section (Section 6.4) describes a solution that represents LCIS' final Active Directory Design. It is the multiple domain model with one extra forest, that of the Legal Department.

6.4 LCIS' Active Directory Design

After many meetings and debates, both Physics and Production decided to join LCIS' Active Directory Forest. Legal however, decided to build a separate forest. Below is a diagram showing the current AD design.



What changed Physics and Productions stance? Why would they join the forest when they have stated that they did not trust other organizations with any aspects of their security? The answer came in the form of a managerial control of the Enterprise and Schema Administrators, and the realization of the complications of collaborations and trust when maintaining their forest.

The Forest Root Domain being "empty" (see Best Practice #3), made the managerial control possible. The empty root limits the number of Domain Administration functions required for the forest root. Since the Domain Administrators of the forest root are also Enterprise Administrators (EA), their numbers can be reduced to one or two accounts. Physics and Production strongly insisted that the use of these accounts be limited and that all the EA or Schema Administrators (SA) work be conducted under the supervision of selected programs (a committee to be determined later).

To enforce these controls, smart card readers were installed on the forests root domain's domain controllers. Members of the managerial oversight committee know the PIN numbers. The smart cards for the EAs and SAs are locked in a safe located at the Central IT department. The only way the EAs or SAs can log on to the domain is via a smart card on the domain controllers of the forest root. The DNS admin group and other sub administrator accounts can log into the domain without smart cards, and may also logon via a network connection, allowing normal operational procedures (procedures that do not impact the other domains) to continue as normal.

These safeguards can be traced backed to the Forest Plan in Section 4. The Schema Change Control Policy and the Configuration Change Policy (see Section 4.1.3), should document the control on these powerful groups.

The Legal department did not agree to these managerial controls as the proper level assurance to their security posture. This was an easy decision for Legal, as they do not have any electronic collaboration with the internal programs. However, their personnel must have access to some internal business systems for time cards, HR, etc., and these systems are accessible to Legal by just a few one-way-trusts. (The Business Domain must trust the accounts in the Legal Department; Legal does not have to trust the business domains).

The discussion above points out that your design is not necessarily limited by the technologies; good management practices can overcome most limitations of the technology.

Part III: Active Directory Best Practices

7.0 Best Practices for Active Directory Design

If you have reached this point of this paper, you have the background to understand the list of best practices listed below. Still, you are not an Active Directory expert, but you can intelligently question any major deviations from the list below. If you are presented with an AD design that does not adhere to most of the below best practices, insist on reference sites where the design feature in question has been successful.

Best Practice Number 1
The Active Directory is a hierarchical structure. Best practices warrant that "complexity" is pushed down the hierarchy. To clarify, our design goal should be a simple forest structure, a simple site structure, and a simple domain structure. Therefore, any "complex" hierarchical structures should be designed into the Organizational Units (down the hierarchical tree), such as security groups and group policy. Also, these complex structures will be unique for each domain.

The justification for Best Practice #1 is simple. The active directory will have to respond to change, as the business practices, organizations, and the technologies are expected to undergo changes. The easiest unit within the AD to change and move is the OU. All other elements are very difficult to change given the current tools. Changing the OU structure or moving an OU within a domain, is just a matter of "point and click" and drag and drop.

Best Practice Number 2
It is important that the forest plan requires a minimum amount of restructuring as your organization evolves.

Best Practice Number 3
Create a dedicated domain to serve solely as the forest root.

By definition, the first forest created is the root of the forest. It is the immutable nexus of the entire hierarchy. The Schema and Enterprise administrator groups are contained in the forest root, as these groups are forest-wide.

Using a dedicated domain as the forest root has the following benefits.

- The Domain Administrator in the forest domain will be able to manipulate the membership of the Enterprise and Schema Administrator's groups. You might have administrators who require domain administrators' privilege for some part of their duties, but you do not want them to manipulate the forest-wide administrators groups. By creating a separate domain, you avoid having to place these administrators into the domain administrators group of the forest root domain.

- Because the domain is small (no unnecessary user or computer accounts), it can be replicated anywhere on your network to provide protection against geographically centered catastrophes.
- A small domain can be restored rapidly by backup.
- The forest root domain never risks becoming obsolete because its only role is to serve the forest.

A dedicated Forest Root Domain is highly recommended.

Best Practice number 4
User Accounts need to be on a Domain Controller that is located in the same site as the user. The objective of partitioning is to put physical copies of directory objects near the users that use the objects (resource kit 276).

Best Practice number 5
Limit the number of Domains.

Best Practice number 6
Limit the number of trees.

Best Practice number 7
Keep the top-level programmatic domains "static" as domains are difficult to change, move, or rename.

Best Practice number 8
Design Group Policy at the domain level. GPO's at the site level should be limited to services that are institutional in scope. A distributive file system for the entire institution is an example of a GPO at the site level; roaming users is another example. (The AD comes with "default policies" for a domain, and has no default policy for a site, as site GPOs have very limited applications).

Best Practice number 9
Do not cross link GPO from one domain to another. There will be serious performance issues due to the cross-domain linking of GPO.

Best Practice number 10
"The best approach that you can take with OUs is to create them based on your IT administrative structure and not on your organization's management structure (or any other structure, for that matter)." The above is an excerpt from Microsoft's AD technical reference, but you can find similar statements in most of the works listed in the bibliography.

Appendix A. DNS Options

Below are some options that are available for Active Directory and DNS. The sections below will describe the options as technologies and then point out the negative aspects of each option. At this time there seems to be no perfect solution to this problem. Option 5 is an example of a DNS design based on a short-cut.

This section really belongs to the DNS deployment phase of the design process and not the Domain Name Space planning. However, this section is included because the DNS designers may insist on changes to the Domain Name Space design to facilitate the implementation of DNS. Also, do not be surprised if a cultural war breaks out between the DNS managers (UNIX people) and the AD design people (Windows people), as DNS has been stable running on UNIX since 1984!

Microsoft's integration of DNS to the Active Directory Domains is truly invasive to the existing DNS and probably to the local IP address registration process. It is imperative that the DNS management team be engaged in the AD design process as early as possible. The AD design effort cannot advance without the cooperation of the existing DNS team.

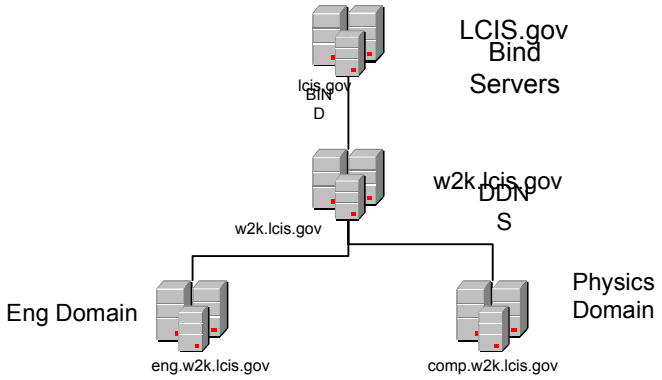
The options that follow describe the common DNS architectural options available to the design effort. These descriptions are brief, and therefore incomplete. DNS expertise is not necessary to understand this section, as these options are designed as a roadmap to the DNS design process. The first four options are typical and most designs will end up considering these options, whereas, Option 5 is included as a DNS short cut design that will have real ramifications in the Domain Name Space design, weaken the security, and complicate operations of the AD.

DNS Option #1
Replace the current lcis.gov BIND servers with Microsoft's Dynamic Domain Name Services (DDNS)

For large and stable DNS implementations, such as LCIS's DNS, this is the **poorest of the solutions**. While DDNS has some nice features, it is not RFC compliant. Also, this is the first release of DDNS and therefore, its stability, scalability, and security should be questioned.

DNS Option #2
Start the forest as a sub-domain of the current DNS domain

This option has been successfully implemented by industry and educational institutions, such as Leicester University (described in Lowe-Norris' book, see bibliography in Appendix B).



The above figure shows lcis.gov bind servers as the domain for LCIS. These servers have "delegated authority" to the w2k.lcis.gov DDNS servers. All the Active directory computers are registered and supported the w2k.lcis.gov DDNS servers; all other computers are registered in the lcis.gov domain. A nice feature of this configuration is that the root domain, lcis.gov, can run BIND without using dynamic updates, essentially running BIND in its current configuration.

This option, while viable, will create problems. First, current inventory systems must account for the extra sub domain. Second, this option will cause confusion in the programs. Consider Engineering, the UNIX computers are registered as machine-name.lcis.gov and the AD computers are registered as machine-name.eng.w2k.lcis.gov. The system managers will have two domains to consider when setting up shares and other services between the two platforms, and the users will have to understand the differences of these domain names.

DNS Option #3
A Separate Internal DNS root Domain

The National Security Agency (NSA) has recommended this configuration. See the NSA Guide to Securing Microsoft Window 2000 DNS. This guide recommends that you separate the AD DNS from the DNS server providing services outside the organization. This type of DNS deployment has historically been called Split DNS.

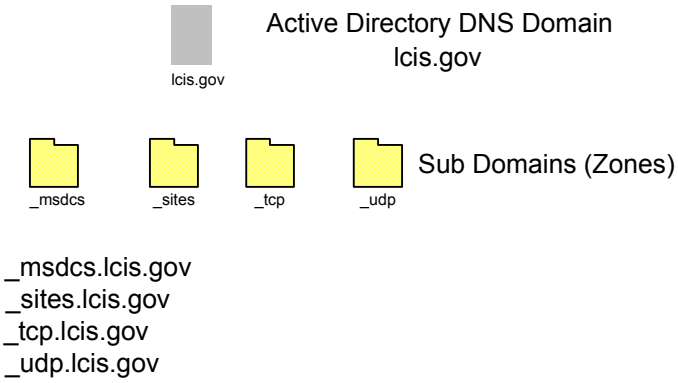
This configuration has a Security advantage. It prevents outsiders from querying the internal DNS service records, which are greater value to hackers than the host or pointer records. Another advantage is this configuration does not require any modification to the inventory registration process (the zones are Mirrored or Shadowed). However, it will cause some of the same types of confusion to the system managers and users as described in the sub domain configuration above.

DNS Option #4
Delegating the Microsoft DNS Service Domain's Only

This configuration delegates the DNS services domains from the domain that houses the Active Directory partition. All the service resource records are stored in four separate sub

domains of the DNS domain, which is the Domain name of the local partition. These domains are:

_msdcs (Domain controller services)
_sites (Windows 2000 sites)
_tcp (TCP-based services)
_udp (UDP-based services)



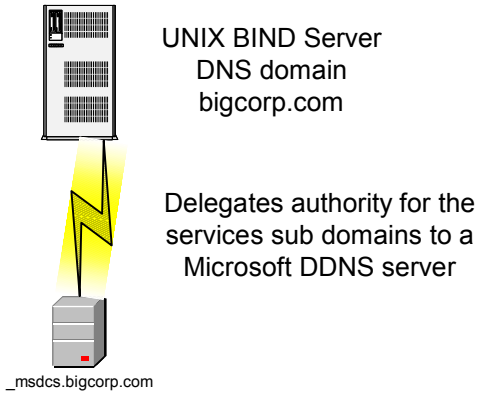
The figure above depicts these sub domains.

The interesting thing about these sub domains (or zones) is that they can be delegated to another DNS server. Some organizations have used this feature to cure all their DNS design issues. Since LCIS needs multiple domains, this option is not applicable to their design.

Here is a case where delegating the service domains can be applied. Imagine a large organization with a single DNS domain and call it bigcorp.com. The DNS services are all UNIX/BIND based. Imagine also that the organization of work and the IT organization is wholly centralized.

The Active Directory design team decides that a single AD domain is perfect for their organization. The DNS managers refuse to implement the required dynamic update BIND version 8.1.2. They also refuse to change their IP address registration process.

Here is the solution for bigcorp.com.



The delegation of the service domains to Microsoft DDNS server allows the BIND servers to remain intact, and also provides the dynamic update capability for the services records to the Active Directory. Note there is an obvious hazard to performing this delegation from the BIND servers. The Microsoft DNS Services sub zones all have an underscore character in the zone name. The underscore character is not RFC compliant and BIND does not support the underscore character. Here is a quote from page 68 of William Wong's DNS book, "Be aware that Windows 2000's use of the underscore character can cause problems with third-party DNS servers." By "third-party" DNS servers, he means RFC compliant DNS servers!

This approach is not very helpful for an organization that requires many AD domains. There is a misconception that the services sub domains can be delegated independently from the parent Domain. While DNS will allow delegations to any designated server, this does nothing for the Domain partition and the LDAP name space. It is impossible to partition the Active Directory simply by delegating the DNS services domains.

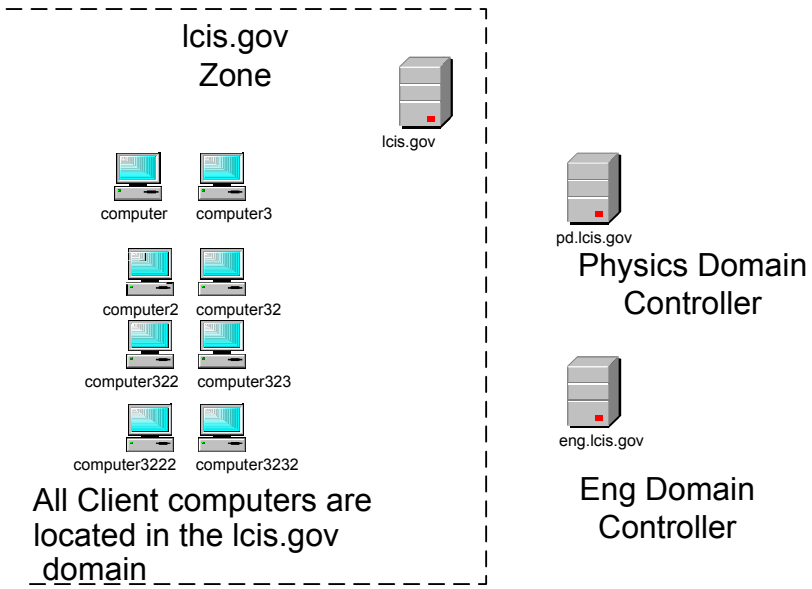
The author of this paper believes that Microsoft pushes a "single-domain model for AD" because they can reduce the intrusiveness of the DNS requirement merely by delegating the services domains from the existing DNS BIND infrastructure.

DNS Option #5 A Short-Cut Design
(Domain Controllers Only in the DNS Sub Domains)

The author of this paper has never seen this option described or listed in any of the books listed in the bibliography. However, the author of this paper tested this configuration as it was presented as an option to a design committee.

This example shows that a deviation from the design principles of Microsoft, best selling authors and design experience, only creates a convoluted design that will lead to a reduction of effectiveness for all users of the directory. This DNS design would place all the AD computers in the top level domain and only the programmatic Domain Controllers would be located in a DNS sub-domain. It was believed that this approach would;

- Minimize the impact to the current DNS infrastructure
- Protect the current IP Address registration process
- Prevent the addition of new fields in the current inventory database
- Reduce end user confusion as to what domain they are located in
- Program autonomy is provided by the program domain controllers, which are the only computers located in the DNS sub-domain



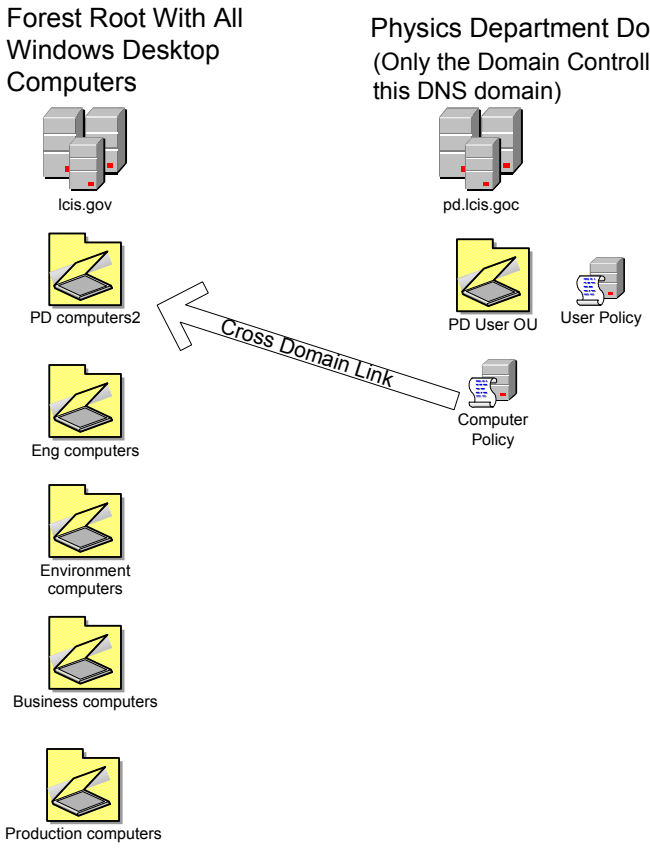
The figure above shows the lcis.gov domain, which contains all the client computers at LCIS. There are two sub domains also depicted, pd.lcis.gov and eng.lcis.gov. The key characteristic of the sub domain is that it only contains the records for the program's domain controllers. All the Windows 2000 clients are located in another domain, lcis.gov. This approach is advantageous to the IP Address process, in that only the registration of new domain controllers will constitute an exception to the current process.

This approach has some serious shortcomings. Reflect on the DNS and LDAP technical section above. All the clients will think their partition (which contains all the objects associated with the client) is the lcis.gov domain (partition). Even the user accounts will be located in a different domain than the user's own computer. Natively, the clients will locate the wrong resources and all LDAP queries will generate LDAP referral queries to another domain, which negatively impacts the performance of the entire directory!

However, there is a fix to the first problem. The clients have a registry setting that can change the client's default partition location. Clients will have to account for their domain location as a separate administrative task. (Actually, there is a GPO setting that will change the client's default partition, but this will not stop the LDAP referrals problem).

This approach is possible, but consider this quote from Lowe-Norris page 118, "Where the client is placed in the forest determines part of the name. Standalone servers and Domain Controllers will be placed in the individual domains they host. Clients can be placed anywhere, but **usually** are placed in the domain that the users of that client **normally** will log on to."

Let's consider the confusion and security concerns of this design. The diagram below shows the pd.lcis.gov domain controller in its DNS domain, and PD's end user's workstations that are located in an Organizational Unit located in the forest root, lcis.gov.



It is obvious that this is in opposition with Best Practice #3 (empty or simple forest root) and Best Practice #9, (avoid cross linking of GPOs from domain to domain). Notice also that the Enterprise Administrator has allowed the linking of a GPO to the forest root domain. Other than weakening the overall security of the forest root, the trouble shooting of GPOs will be more complicated than is the case of the computer accounts and user accounts located in the same domain.

Here is another aspect of this design that will **devastate the security of the forest root**; each domain administrator will require write permission to the forest root's system volume, in order to utilize startup and shutdown scripts on the workstations. (The start up scripts must be able to replicate to each domain controller in the domain, the forest root in this case).

This reduction in utility, weakening of security, and complication of management, is the by-product of this DNS short cut design!

DNS Option #5 demonstrates the hazards of Active Directory design. A decision to support a short-cut design can be an extreme risk to any design project. Make sure that any and all design features, including DNS, can be technically justified. Also, insist that the technical designers provide references to any design that deviates from the Documentation.

Appendix B. Bibliography

Windows 2000 Active Directory
Alistair G. Lowe-Norris / *O'Reilly & Associates* / *2000* / *1565926382*

Active Directory Services for Microsoft Windows 2000 Technical Reference
David Iseminger / *Microsoft Press* / *2000* / *0735606242*

Building an Enterprise Active Directory Notes from the Field
Microsoft Consulting Services (Edt) / *Microsoft Press* / *2000* / *0735608601*

Windows 2000 Active Directory Design and Deployment
Gary Olsen / *New Riders Publishing* / *2000* / *1578702429*

Implementing Directory Services): Microsoft Active Directory, Novell Nds, Netscape Nds, and Cisco/Microsoft Directory-Enabled Networks (Enterprise com)
Archie Reed / *McGraw Hill* / *1999* / *007134408X*

MCSE Training Kit: Designing Microsoft Windows 2000 Network Security
Microsoft Corporation / *Microsoft Press* / *2001* / *0735611343*

MCSE Training Kit: Designing a Microsoft Windows 2000 Directory Services Infrastructure
Microsoft Corporation / *Microsoft Press* / *2001* / *0735611327*

Mastering Windows 2000 Server
Mark Minasi, et al / *Sybex* / *2001* / *0782128726* / *3rd CD*

MCSE Training Kit: Migrating from Microsoft Windows NT 4.0 to Microsoft Windows 2000
Microsoft Corporation / *Microsoft Press* / *2001* / *0735612390*

Microsoft Windows 2000 Professional Resource Kit: Comprehensive Resource Guide and Utilities for Windows 2000
Microsoft Corporation (Edt) / *Microsoft Press* / *2000* / *1572318082*

Microsoft Windows 2000 Server Resource Kit
Microsoft Corporation (Edt) / *Microsoft Press* / *2000* / *1572318058*

Configuring Windows 2000 Server Security
Syngress / *1999* / *1928994024*

Microsoft Windows 2000 Security Technical Reference
Internet Security Systems (Edt) / *Microsoft Press* / *2000* / *073560858X*

Windows 2000 Security Handbook (Network Professional Library)
Tom Sheldon / *McGraw Hill* / *2000* / *0072124334*

DNS and BIND 4th Edition
Paul Albitz, et al / *O'Reilly & Associates* / *2001* / *0596001584* / *4th*

Windows 2000 DNS Server
William Wong / *McGraw Hill* / *2000* / *0072124326*

The Concise Guide to Windows 2000 DNS
Andy Ruth, et al / *Que* / *2000* / *0789723352*

Windows 2000 Design & Migration
Rand Morimoto / *McGraw Hill* / *2000* / *0072122056*

MCSE: Windows 2000 Migration Exam Notes
Todd Phillips / *Sybex* / *2001* / *078212769X*

Understanding and Deploying Ldap Directory Services
Tim Howes, et al / *New Riders Publishing* / *1998* / *1578700701*

Windows 2000: Group Policy, Profiles, and IntelliMirror
Jeremy Moskowitz / *Sybex* / *2001* / *0782128815*

Admin911: Windows 2000 Group Policies
Jennings / *McGraw Hill* / *2000* / *0072129484*

Hacking Windows 2000 Exposed
McClure, Scambray / *McGraw Hill* / *2000* / 0072192623