# Purdue University

## Purdue University

## Microsoft Windows 2003 and Active Directory Server Architecture Plan

**Document Version # .01**

Brent McCarthy

12/4/2002

Microsoft Consulting Services

# Contents

# Overview

As businesses compete in a constantly changing landscape, companies are searching for new and exciting ways to bring their products and services to market. Taking advantage of these opportunities is paramount to a company's growth. However, prior to taking on added resources, it is important to have an infrastructure that can readily respond to market forces. The infrastructure needs to offer efficient management, ease of use, and support for the latest advances in networking and server hardware.

To accelerate the deployment and maximize the return on investment in Microsoft® Windows® 2003, a rapid design effort has been undertaken with the goal of producing an Architecture Plan while at the same time delivering knowledge transfer. Through several focused design sessions, the team has been able to rapidly execute and deliver a detailed Architecture Plan.

This architecture document describes the use of foundation components in a Windows 2003 infrastructure, focusing on the "high risk" elements of the design. The "high risk" elements are categorized as the components that describe the most fundamental behaviors and attributes of the environment. Examples include the number of forests, the root name of Microsoft Active Directory™ directory services, and the number of domains that will be the foundation of the enterprise.

The focus on simplicity, global services infrastructure, and efficiency in operations has been core to the discussion.

The purpose of this design effort is to provide recommendations on the base components of the Windows 2003 infrastructure. The design team has examined many areas of the current operating environment and how they can be mapped into a simplified consistent infrastructure.

The design team has sought to preserve two main themes:

- Provide a consistent and secure operating environment that increases efficiency.
- Provide flexibility to business units for the management of information and services.

This architecture planning effort includes information on each of the following components:

- **Microsoft Active Directory**. Active Directory provides a structured yet flexible framework for the organization and management of objects ranging from users and computers to servers and databases.
- **Member Server**. This section addresses design and management of member servers with particular attention to core file and print strategies of Windows 2003.
- **Application Compatibility**. Application compatibility addresses planning the migration of various applications to Windows 2003, including accessing current hardware and software inventory, roles and responsibilities, and the testing process.
- **Operations**. The operations section describes Microsoft Operations Framework (MOF) and how these standards can benefit the customer when using Windows 2003. Additionally, some core recommendations for monitoring and managing the health of Windows 2003 Servers have been included in some detail.

## Vision and Scope

This section summarizes the vision and scope for the Windows 2003 Server migration platform design and deployment effort.

The University's vision of the Windows Active Directory Project is to achieve a Windows Active Directory environment that will meet all of the universities business needs for information technology management as well as decrease the total amount of time and money that is spent managing these technologies.

The scope of this project will address Active Directory needs for the entire university. The Active Directory will form a contiguous name space for Purdue University, but sill allow for the configuration of Departmental name spaces within the environment. The Active Directory will have a domain based network model, which meets the Universities needs. Centralized management to manage University resources is also envisioned with in this project, along with improved security and improved access to those resources within Active Directory. Lastly, the vision is for Active Directory to provide authentication and authorization of all university resources within Active Directory.

## Goals and Benefits

- *Migration Transparency.*

  Ensure that existing Windows 2003, NT 4.0, and Novell users will be able to login and work transparently after the domain has been migrated into the new Forest. Existing services provided by these areas will be integrated and maintained without excessive administrative overhead.

- *Account Management.*

  Streamline, secure, and simplify the creation, management, delegation, and use of accounts and resources across all campuses.

- *Single Sign On.*

  Encourage Single Sign On for all of Purdue University students, staff, and faculty at all campuses of Purdue University.

- *Cost Efficiency.*

  Maximize value by minimizing administrative labor and hardware costs. Administration can be completely and securely delegated to any department, campus, group, or individual on campus. This delegation of permissions cannot be overridden without leaving an un-erasable security trail. Greater integration with other services through LDAP or other protocols to enhance compatibility and reduce the need for custom coding to ensure cross-platform interaction.

- *Open Internet Standards.*

  Utilize existing open Internet Standards and protocols to ensure interoperability with existing and future systems. Windows utilizes RFC-compliant DHCP, DDNS, SMTP, LDAP, Kerberos v5, and TCP/IP.

## Risk Management

During the course of this design engagement, we noted several items that put the success of the subsequent deployment project at risk. These items require more detailed analysis and must be addressed to ensure a successful deployment.

| Risk and recommendation | Severity | Impact (if not mitigated) |
|---|---|---|
| Risk: Effective Communication of project information to University Schools and Departments<br><br>Recommendation: Implement a specific process for disseminating all project information to the different Schools and Departments (Extranet?) | High | Possible department pushback and resentment of project. Slow decision process. |
| Risk: Lack of School or Department involvement in Design and Implementation committee meetings.<br><br>Recommendations: Inform Schools and Departments of the importance of their involvement in the planning process. Reiterate the fact that without their involvement their needs may not be sufficiently met. | High | End design result may not reflect the needs or requirements of all of the Schools or Departments that are going to be a part of the University Active Directory. |
| Risk: Inability to implement a standard strong password requirement for Active Directory and all connected systems.<br><br>Recommendation: Implement and enforce a strong password policy according to the specific recommendations in the Password portion of this document | High | Degrades the overall security of the Active Directory and all resources attached to it. Allows for possible un-authorized intrusion to Active Directory resources. |
| | | |
| | | |
| | | |

## Next Steps

The outcome of this planning effort includes the key decisions made through the interactive design sessions. This architecture plan can serve as a foundation for creating a detailed functional specification for the pilot and deployment phases.

Next steps might include:

- Finalizing the design plans (functional specification).
- Creating detailed migration and deployment plans.
- Preparing for a beta or pilot.
- Creating detailed operations procedures.
- Developing user education and training procedures.

## Future Enhancements

The following is a list of items determined out of scope for the first pass design but tracked for possible future enhancements to the design:

### Microsoft Systems Management Server 2003

# Existing Environment

Documentation of the existing environment is used as a foundation for creating the first pass design for the new system. The location for the documentation used is found here:

### *Overview*

Purdue University is made up of four campuses; West Lafayette is the main campus, with three other Regional campuses. The three other campuses are North Central, Calumet, and Fort Wayne. The West Lafayette campus has approximately 38,000 students enrolled, with 13,000 faculty and staff. The North Central campus has approximately 3,000 students enrolled. Calumet campus has 9,000 students enrolled. Fort Wayne campus has approximately 5,000 students enrolled. All the faculty and staff combined on the three regional campuses amount to approximately 3,000. This gives us a total of 52,000 students, and 16,000 faculty and staff.

Purdue University is made up of multiple schools, divisions, and departments. Some of the schools and departments are responsible for clients that are physically located throughout the state of Indiana, for example the school of Agriculture has clients and offices in all 98 counties, and the School of Technology has 16 Statewide Technology sites.

### *Current Domain Environment*

Currently all faculty, staff, and students have unique accounts in the Central Directory maintained at the West Lafayette campus. Many NT 4.0 and Windows 2003 domains continue to maintain their own set of accounts.

The current configuration for ITCS is a single NT 4.0 Domain, ADMIN_DOMAIN, with 80 Member Servers serving a variety of services. Some of those services include IIS Web Servers, File, Print, SQL Database, Oracle Database, and SMS 2.0, to all clients. This environment has Novell servers currently providing file services to a large number of constituents.

Within Purdue University there are other NT4.0 and Windows 2003 Active Directory Domains, which will be collapsed into the new Windows 2003 Active Directory Forest. Some of those domains currently provide messaging, file, print, and database services to their respective clients. Other Domains provide authentication for students in order for them to access client and server resources, such as login ability to labs.

### *Current Network Infrastructure*

West Lafayette site has 1 Gbps to Indianapolis for all campus connections to Internet2 and ResNet connections to the commodity Internet, along with 155 Mbps to Indianapolis for non-ResNet to the commodity Internet. For each of the Regional campuses a T1 connection is made for intercampus connectivity only. Each of the Regional campuses has their own connections to the Internet. On the West Lafayette campus connections can range from 10Mbps Ethernet up to GigEthernet.

### *Current Network Service Infrastructure*

At the West Lafayette campus, DNS services are provided by IT Telecommunications department, for those areas that wish to use it. A number of departments through campus maintain their own Secondary, Caching, or even some Departments run as start of authority for their own name space. The IT Telecommunications

DNS is provided on a UNIX system running Bind 9.xx. This server does not except Dynamic updates for security reasons.

Exchange 5.5 servers, providing messaging services to many faculty and staff on the West Lafayette campus. Some of the network services currently being provided to clients are DHCP services to 18 subnets, also providing WINS services to those subnets. Caching Name Servers are also being provided to some of those subnets, provided by Unix servers.


17 ' RP DLQ
,7&6
: LQGRZV ' RP DLQ
3K\VLFDODFFHVVHV
: LQGRZV ' RP DLQ
$JUHHPHQWV
: LQGRZV ' RP DLQ
,QVWUXFWLRQ&RP SXWLQJ
: LQGRZV ' RP DLQ
7HFKQRORJ\
: LQGRZV ' RP DLQ
( QJLQHHULQJ
: LQGRZV ' RP DLQ
/ LEHUDO$UWV
: LQGRZV ' RP DLQ
&RPSXWV6FLHQFH
: LQGRZV ' RP DLQ
0 DQDJHPHQW
: LQGRZV ' RP DLQ
<HVWEHGLVFRYHUHG
17 ' RP DLQ
< HVRE HGLVFRYHUHG

# Design Summary

Based on the goals of the project and the discovered requirements of Purdue University a Windows 2003 Active Directory design is recommended that is consistent with Universities needs and well-established best practices.

The overall Active Directory Design for Purdue University consists of a single Windows 2003 forest, a dedicated root domain, a large shared domain, and a single Exchange 2003 organization which would be defined centrally.

Schools, and departments will be encouraged to participate as organizational unit containers within the shared domain. These containers are boundaries for administrative delegation and will provide essentially the same functionality present in an NT 4 domain or NDS context. This implementation will provide the greatest cost and feature benefits, including Kerberos and user data synchronizations, as well as security and reliability improvements over the current environment.

## Forest

A forest is a collection of one or more Windows 2003 Active Directory trees, organized as peers and connected by two-way transitive trust relationships between the root domains of each tree. All trees in a forest share a common schema, configuration, and global catalog. When a forest contains multiple trees, the trees do not form a contiguous namespace.

The forest has three main purposes:

- Creates the boundary of Trust
- Creates the boundary of Authentication
- Creates a common directory and schema

The design of the forest was a decision between either a single forest or multiple forest implementation. A single forest will provide a single directory of resources and a single location from which to manage security for the PC infrastructure, as well as provide an improved level of PC services in a manner that is more efficient and more robust than was possible in Windows NT 4. A single forest will represent a new way of thinking about and managing PC and client/server systems on an enterprise level. A multiple forest design will limit the ability of groups to share resources and applications, and will perpetuate Purdue's current system of independent PC infrastructures. A multiple forest implementation would severely restrict Purdue's ability to evaluate and take advantage of many of the feature and cost benefits of Windows 2003.

*Recommendations*

- Create a single Windows 2003 forest across the entire university.

## Schema

The Active Directory schema is a new feature in Windows 2003 and will need a new administrative model than has not been present in the environment previously. The schema operations and functionality will impact a broad range of Purdue administrators, developers and users and must be carefully protected from inappropriate modifications.

*Recommendations*

- Designate the Root Domain Administrator user account as the Enterprise and Schema administrator that will be utilized only for forest level changes, such as schema modifications, domain creation, and management of the forest. Apply the highest level of security and auditing to these user or group accounts.
- Implement a set of policies and procedures regarding proposing, testing, and implementing schema and design changes.

## Domain

An Active Directory forest is a distributed database, where domains define the database partitions, as well as the context for uniqueness.

The right technical reasons for creating more than one domain can be reduced to these issues.

- Password length and restrictions, such as complexity and history, are enforced on all users and are domain-wide; therefore, all participants in the domain must agree on these parameters. However, if an administrator decides to change password requirement restrictions, synchronization with the Kerberos realm will not be possible.

- WAN links cannot handle the inter-site replication traffic; therefore the Active Directory database needs to be partitioned into smaller pieces.

*Recommendations*

- Implement an Active Directory Forest Root domain and a separate account domain to establish the forest and namespace. This configuration would differentiate the Enterprise/Schema Administrators from the account domain administrators.
- All groups should be encouraged to evaluate an organizational unit within the shared domain, and should create additional domains only for one of the technical reasons specified above.

## Namespace

A Windows 2003 namespace is synonymous with its DNS namespace, and unlike Windows NT 4 is dependent on DNS for its operations. A Windows 2003 forest provides the ability to span multiple discontinuous namespaces or domain trees while providing trust and resource access among them. The choice of namespace is critical due to the effort required to modify the choice at a later time.

The dependence of Windows 2003 on DNS SRV records makes the practice of using the same DNS namespace that is currently used and accessible to the Internet a non-recommended security risk, as crucial server roles (domain controllers, global catalogs, LDAP servers), would be identified in the public DNS.

*Recommendations*

- Establish a single domain tree Purdue.LCL as a separate DNS namespace from the Purdue.Edu domain name.

## DNS

Purdue currently is using a Bind version of DNS, which is capable of supporting Windows 2003 and the Active Directory. However, at this point in time dynamic host record updating of the DNS zone file is not enabled. The extensive use of DNS records by Windows 2003 servers and clients make dynamic updates a virtual necessity to insure stability.

*Recommendations*

- Implement Windows 2003 Active Directory integrated dynamic DNS for the delegated zone "Purdue.LCL".
- Establish a line of demarcation between internal and external DNS resources, thus preventing the Windows 2003 DNS infrastructure from being accessible from the internet.
- Configure all Windows 2003 DNS servers with DNS forwarders to the Forest root then to the Public DNS servers for all DNS resolution outside of the Active Directory.

## Sites

A site is any given mask-able IP subnet range with fast, reliable connectivity (greater than or equal to 10 megabytes in most situations). Sites are the boundaries to control the replication parameters and are used to control the direction, schedule and frequency of replication for the active directory, global catalog, and SYSVOL information. In addition to replication traffic, sites are used to find resources that are closest to the requestor.

*Recommendations*

- Establish a comprehensive site design.
- Test and evaluate the Active Directory client extensions for legacy clients, such as Windows NT and Windows 9x.

## Organizational Units

Organizational units are containers used to organize the objects within a domain. Important points to consider about the use of organizational units:

- Organizational units can be nested.
- Organizational units are used to delegate administration.
- Organizational units are not security principals.
- Group policy is applied to an organizational unit.
- Users will typically not navigate the organizational unit structure.

The creation of organizational units should be done for the delegation of administration and the application of Group Policy objects. It is NOT necessary to create an organizational unit hierarchy that is aesthetically pleasing or that mirrors the organizational structure of the company. If security doesn't need to be delegated, or if GPOs aren't being applied to a collection of objects, there is no need to create an organizational unit.

*Recommendations*

- Create a top level OU design for the consolidated domain that has high level OUs for each of the schools or departments so that delegation can be configured easily for entire departments.

- Allow each of the schools, or departments, the ability to organize their resources as they see fit, and construct OU hierarchies based on the individual groups needs.

## Policy

Group Policy is the "active" part of the Active Directory and it defines and controls the behavior of directory objects, applications, and network resources. There are hundreds of policy settings that can be managed, which can be combined with inheritance / override restrictions and security group filtering, to generate a nearly unlimited number of possibilities.

*Recommendations*

- Apply only the absolute minimum security policies at an enterprise level. As part of the provisioning process for an OU container, rights and permissions will be established that will allow departmental level policy application within a given school, or department.

## Migration

Purdue will have a wide range of migration needs and solutions based on the various infrastructures in place. The differing needs of each organization as well as independent timetables and resource availability will require custom migration solutions that are developed to accommodate each of the variations. In addition to migrations, there will undoubtedly be several coexistence scenarios that will need to be addressed.

*Recommendations*

- Implement a pristine forest build followed by a series of moves, migrations, and in-place upgrades in order to achieve the desired end state environment.

- Establish a team of focused technical resources that can provide migration assistance to groups throughout the university.

## Microsoft Exchange 2003

Microsoft Exchange 2003 depends on the Active Directory and presents some additional migration considerations. A new Exchange 2003 organization, as well as forest and domain preparation will need to be implemented in the new Active Directory environment prior to any group with an existing Exchange 5.5 implementation from moving into the new environment or implementing Exchange 2003 in the new environment.

Currently Purdue's central IT provides an enterprise-wide messaging infrastructure, which runs on a number of Exchange 5.5 servers, and provides functionality, such as shared calendaring, contacts, etc, the various departments and schools at Purdue. have employed various messaging solutions including applications such as Microsoft Exchange 5.5, and Exchange 2003.

*Recommendations*

- Establish a centralized Exchange 2003 organization in the Windows 2003 forest.

Established a team of focused technical resources that can provide migration assistance to groups throughout the university.

# Active Directory

Microsoft Active Directory, new in Windows 2003, plays a major role in implementing an organization's network and accomplishing business goals. A better understanding of Active Directory provides a comprehensive and efficient ability to fully utilize Microsoft Windows 2003 Server.

Active Directory includes various structures, such as forests, domains, organizational units, and sites. This document covers many of these structures; however, for additional information, review the Directory Services Web site, "Exploring Directory Services," at:
http://www.microsoft.com/windows2003/technologies/directory/default.asp

## Guiding Principles

The first chapter describes the principles to use when planning and designing the migration.

- **Simplicity is the best investment**. Simple designs are easy to explain and maintain.
- **Aim for the ideal design**. Review the components and decide where you want to go.
- **Explore design alternatives**. The merit of a design is often seen as a comparison to another design.
- **Anticipate change**. How will change or reorganization affect your design? An employee changing departments, adding a branch, reorganizing the company.

### *Simplicity Is the Best Investment*

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

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

### *Aim for the Ideal Design*

In your first design pass, design what you consider to be the ideal structure, even if it does not reflect the current domain or directory infrastructure. It is useful and practical to understand what is ideal, even if the ideal design is not currently attainable. Weigh the migration costs against the long-term savings of the ideal plan and refine the design appropriately.

### *Explore Design Alternatives*

Make more than one pass at each design. The value of a design becomes more evident when you compare it to other design ideas. Combine the best of all designs into the plan that you implement. Exploring design alternatives is also required when performing a comparative cost analysis of differing designs.

### *Anticipate Change*

The normal changes that occur within any organization, ranging from employee moves to enterprise-wide reorganizations or acquisitions, will affect your Microsoft Active Directory structure. When designing the structure, consider how these potential changes will affect the user and administrator interaction with the directory. For example, consider the impact that your last major business reorganization would have had on the structures you have designed. What changes would be necessary if you add a new location or branch office? Would they be significant and expensive changes to the structure in Active Directory? Make sure your design is general and flexible enough to accommodate constant and significant change. Anticipating change is an important aspect of the design, but do not allow all the potential future changes to dominate the design process.

## Logical and Physical Overview

Active Directory planning encompasses several structural components with both logical and physical considerations.

The forest in Active Directory is a distributed database in which the domains define the database partitions. Organizational units are containers within the domain partitions that provide a mechanism for grouping elements of the database.

The next four sections describe the construction of the four basic elements of an Active Directory design:

- Forest structure
- Domain structure
- Organizational unit
- Site structure (physical structure)

## Forest Structure

A forest is a collection of one or more Windows 2003 Active Directory trees, organized as peers and connected by two-way transitive trust relationships between the root domains of each tree. All trees in a forest share a common schema, configuration, and global catalog. When a forest contains multiple trees, the trees do not form a contiguous namespace.

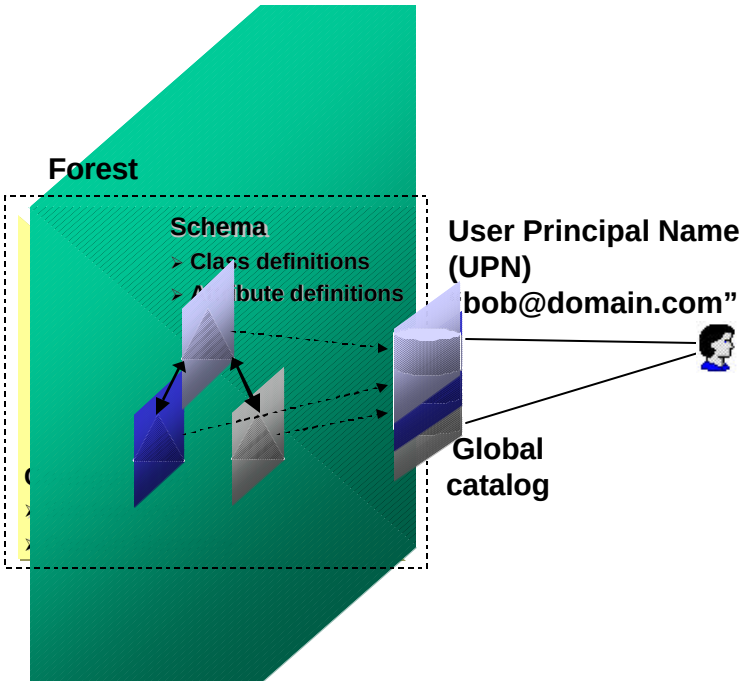



**Figure 1. Purpose of a Forest**

The forest has three main purposes. They are to:

- Make management easier by acting as a container for multiple domains.
- Make the structures transparent to the user.
- Create a common global catalog (enterprise-wide directory).

### Naming Contexts

There are three naming contexts used in Active Directory. There is also a fourth database in Active Directory known as the global catalog.

#### *Domain Naming Context*

The domain naming context (NC) maintains information specific to Windows 2003 domain objects, including users, groups, computers, and organizational units, as well as information about Group Policy objects (GPOs). The entire domain naming context is replicated to all domain controllers within its domain and to global catalog servers, if the update is made to an attribute that is marked for replication to the global catalog.

Changes in Active Directory are replicated on a per-attribute basis. Consider that group membership is a multi-valued attribute and is replicated as a whole. Group Policy objects not only trigger Active Directory replication, but are also reflected in increased File Replication service SYSVOL replication traffic.

#### *Configuration Naming Context*

The configuration naming context is replicated to every domain controller in a forest and contains forest-wide configuration information such as: sites, site links, authorized DHCP service, and others.

#### *Schema Naming Context*

The schema naming context defines the object classes and the attributes of object classes that can be created in the directory. The schema naming context is replicated to every domain controller in the forest.

**Note** The partial list of attributes replicated to global catalog servers is defined in the schema. If an attribute is added to the set of attributes replicated to global catalog servers, that change triggers a full replication of the global catalog.

### Behavior of Forest Objects

The following list describes various forest object behaviors:

- Share common schema.
- Share configuration container (for example, site topology kept on all domain controllers in the forest for calculating replication).
- Share global catalog (provides abstraction of the forest structure—for example, user principle name.
- Automatic two-way transitive trust between domains in the forest.

## Selecting Forest Model

Simply stated, the enterprise is contained in a single forest or multiple forests. Forest planning should begin with the assumption that the entire production enterprise is housed in a single forest. This assumption requires that all participants in the forest can agree about a change control policy for the forest. Specifically, there must be agreement on the changes made to the schema and configuration containers. The security group Schema Administrators owns the schema container, and the Enterprise Admins owns the configuration container. Note that all containers in Active Directory can be delegated so that other users and groups can be granted rights to operate on any containers in Active Directory.

In most organizations, individual domain owners should not object to central schema and configuration management considering that rights within a forest can be fully delegated to allow full control over the respective domains. Examples include: schema changes, such as the addition of another user attribute (in addition to the dozens defined by default), or the addition of a new site link to accommodate a change in the wide area network (WAN).

Creating multiple forests effectively hides the contents of one forest from another and severely restricts the interoperability of resources in those forests. A primary example is that multiple forests do not share a single global catalog, which restricts the discovery of resources across forests and the use of the user directory in forest-aware applications such as Microsoft Exchange 2003, which uses the global catalog as the global address list.

Multiple forest designs provide complete autonomy, separate schemas, global catalogs, and distinct security boundaries.

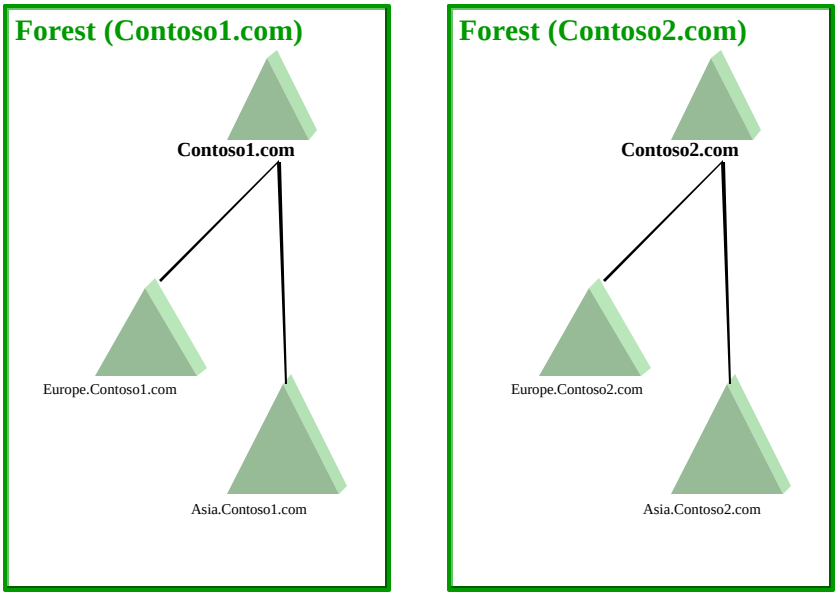



**Figure 2. Multiple Forests**

Test forests can be set up with manually created trusts between domains in the production forest, as necessary.

**Note** Forests cannot be merged or split, and domains cannot be moved between forests. Use the following tools to move individual objects between forests.

- ClonePrincipal is used to move users and groups
- Netdom.exe is used to move computers
- Ldifde.exe is used to move bulk import and other objects.

### *Uses and Considerations for Multi-Forest*

The following section describes various reasons for using multiple forests.

- Cannot agree on forest change control.
- Legally must not share resources.
- Must trust all administrators equally, since all copies of shared naming contexts contain writing privileges.
- Has no need to share the global catalog.
- Does not desire complete trust.

### *Examples of Multiple Forests*

Examples of multiple forests are:

- Acquisition
- Specialized directories (extranet, development)
- Conglomerate
- Grass-roots
- Armed forces
- Hosting (ASP/ISP)

### *General Implications of Multiple Forests*

The following information describes general implications of multiple forests:

- Additional management overhead, caused by:
  - Multiple schemas
  - Multiple configurations
  - Manual external trusts
- External trust is one-way, non-transitive
- No cross-enterprise security context
- Global catalog query only sees objects in one forest
- User principle name logon limited to default user principle names
- One Exchange organization per forest

## Extending the Schema

The directory schema serves the entire forest and can be the foundation for application development and services delivery in Windows 2003. Like any database or directory, the definition of the underlying schema is critical to its usefulness and flexibility. Microsoft built Active Directory to be extensible, flexible, and secure while providing a standard based stable foundation for enterprise infrastructures. Windows 2003 ships with several tools for managing, manipulating and modifying the schema, schema snap-in, ADSIEdit, and Ldp. However, the schema does form the basis for the operating system services and therefore is not completely free to manipulation. Mandatory attributes of basic objects, object classes, or the renaming of built-in classes, and so on, are not permitted to prevent destabilizing of the operating system or potential security violations. Unlike a directory or database that is intended to serve a singular function, Active Directory schema must have comprehensive change control policies implemented to ensure proper operations and functionality.

### *Change Control*

Changes to Active Directory schema should be infrequent, but well tested. Object and attribute additions are not reversible; objects can be disabled but not deleted once created. Microsoft explains the naming standards at: http://msdn.microsoft.com/certification/NamingRules.asp. These standards enable software vendors and enterprise customers to register schema modifications and naming standards, and allow the coordination of schema changes between Microsoft, software vendors, and customers. Internal coordination is also necessary and left to the Enterprise.

Distributed development teams will need a way to coordinate desired changes to Active Directory schema to prevent conflicts in naming, data type definitions, security, attributes of single or multi-values, and so forth. In addition, operation teams responsible for the network and operating system layer will need information on any desired changes that include size of fully populated object, expected frequency of change, size of changes, and so forth. Typically, the change control policy involves four stages: submission, developmental review, operational review, and implementation.

1) **Submission**. The publishing of proposed changes to a public repository available to developers and IT personnel. Commonly set up with a specific duration that a proposal must be made public. It is in this phase that development teams should coordinate naming and goals for an object or attribute. A Web site, Exchange public folder, or SharePoint portal are all possible tools to implement this functionality. The SharePoint server provides subscription services and revision history tracking as well as powerful indexing and search capabilities, making it an ideal choice for such a solution.

2) **Developmental review**. The period that a proposal is open for negotiation among development teams. May have a committee of development team representatives charged with ensuring that proposals are at the very least communicated and possibly edited.

3) **Operational review**. A period of time after the developmental review period in which no longer are developmental changes allowed but rather operational impact analysis can be completed on a known proposal. Often the group or individual proposing the change will have to submit typical sizing, rate of change, and size of change estimates for review by operational staff. A monthly or quarterly review of these submissions is common. Again the routing and approval functionality of the SharePoint server makes it an excellent solution.

4) **Implementation**. Upon approval of operations, the schema management team or individual will make the necessary schema modifications typically using the Ldifde.exe utility and a standard LDIF file. A monthly or quarterly cycle can be established that can adequately provide timeliness for developers as well efficiency for operations and schema managers.

This outline is only a sample to provide guidance for the implementation of a comprehensive schema change control management solution.

### *Obtaining Valid Object Identifiers*

An object identifier is a number that unambiguously identifies an object class or attribute in a directory. Object identifiers are issued by distribution authorities and form a hierarchy. An object identifier is represented as a dotted decimal string (for example, 1.2.3.4).

Object identifiers are guaranteed to be unique across all networks worldwide. They are used to ensure that objects defined by different entities do not conflict. The following authorities issue object identifiers: the International Organization for Standardization (ISO) or the national registration authority for various countries. The ISO recognizes national registration authorities and maintains a list of contacts on the ISO Web site at: http://www.iso.ch/.

> **Note** E-mail and Web addresses often change, to ensure you are using the correct address, use a search engine to find the Web site or sites mentioned here.

Most countries have an identified national registration authority responsible for issuing object identifiers. The national registration authority in the United States is the American National Standards Institute (ANSI).

The national registration authority issues root object identifiers. An enterprise can register a name for the object identifier as well. There is a fee associated with both root object identifiers and registered names. Contact the national registration authority of your country for details.

It is possible to receive object identifiers from Microsoft by sending e-mail to the addresses listed in the Web site mentioned below.

In the content of your e-mail, include your organization's naming prefix, and the following contact information:

- Contact name
- Contact address
- Contact telephone number

> **Note Schema extensions are not reversible.**

Once a new class or attribute has been added to the schema, it can be deactivated, but it cannot be removed. Deactivating a class or attribute prevents new instances from being created. You cannot deactivate an attribute if it is included in any class that is not deactivated.

Additional information can be found in Microsoft Windows 2003 Advanced Documentation at: http://www.microsoft.com/WINDOWS2003/en/advanced/help/sag_Adschema_13.htm

## *Situation and Requirements*

### ***Forest Situation and Requirements***

The existing domain environment at Purdue is decentralized with many different departments running their own domain. One of the primary goals of the Active Directory implementation is to centralize the domain efforts of the University as much as technically possible.

By consolidating all of the domain efforts, the University will realize large cost savings in hardware, administration and support of domain management overall.

Also, to leverage the cost savings, administration and performance capabilities of using a centralized Exchange Server environment only one Exchange Organization will be implemented.

### *Schema*

Purdue University has an extremely disparate environment with several projects occurring at the same in different schools and departments. There is often not a lot of communication occurring between the different schools and departments which makes making global changes to the Active Directory somewhat of a challenge.

Because the schema impacts every department and school within the University special consideration needs to be taken to ensure that if changes are going to be made to the schema that the proposed changes are reviewed and implemented based on the needs of the University overall to ensure that changes to the schema do not have a negative impact to any school or department within the directory.

## Design Decisions

### *Forest*

Purdue University will implement a single forest managed by ITCS to support the University wide Active Directory implementation. Having a single makes it possible to have a single directory that can be managed centrally. Using a single forest also supports the requirement of having a single Exchange 2003 Organization because an Exchange Organization cannot cross a forest boundary.

Having a single forest will also be the most cost effective solution from a hardware/software and administration standpoint.

### *Schema*

There process for implementing schema changes will occur in 4 phases:

1. Requesting department submits formal schema change request to ITC.

The requesting school should submit in writing to the ITC a formal schema change request which will contain the technical specifics containing all schema classes and class attributes that are being requested, as well as the functional requirements and architecture of the application that will leverage the implementation of these schema objects.

2. Request goes before a technical committee for approval.

A committee of knowledgeable representatives from each school or department must be formed to meet and discuss whether to recommend implementing a proposed schema change or not. By having a committee of representatives of all different schools or departments to approve proposed schema changes ensures that the schema needs of each area are taken into account.

3. Committee recommendation goes before ITCS executive management for final implementation approval.

Once the committee has determined the impact of modifying the schema with the proposed schema changes, they will submit the formal schema change request along with a formal recommendation to ITCS executive management for final approval of the requested change.

4. Test of schema changes and final implementation.

If ITC executive management approves the recommendation of the committee to implement the proposed schema changes then the ITC Schema Administrator will work closely with the requesting department in testing the proposed changes in a test Active Directory environment that matches the production directory as closely as possible.

Once testing of the approved changes has been determined to be successful then the ITC Schema Administrator will implement the approved changes into the production directory.

# Domain Structure

## Windows 2003 Domains

An Active Directory forest is a distributed database where domains define the database partitions. A *distributed database* is a database that is made up of many partial databases spread across many computers, instead of a single database on a single computer. Splitting a database into smaller parts and placing those parts on computers where the data is most relevant allows an administrator to distribute a large database efficiently over a large network. Database partitioning commonly occurs at geopolitical/continental boundaries where database segments "live" primarily on a continent or country. Examples include large databases divided into partitions such as "Americas," "Europe," and "Asia."

In situations where the network can support the replication of a single database, the ever-present local copy of the information can provide a consistent experience for the end user that is location-independent. This allows the administrators to design the best structural and logical models (such as groups and organizational units), independent of physical considerations.

### *Forest Root*

The first domain in a forest is the forest root. The root domain contains the Schema Administrator and Enterprise Administrator groups, and the Domain Administrator of this forest root domain controls the membership of those groups. For clear separation of these roles and other special forest-level roles in larger environments (explained in the following section on operations master roles), it is recommended that you create the forest root domain as a separate peer domain in the forest or as a tree root (both examples are illustrated in the following picture). The latter scenario is commonly referred to as dedicated forest root.

Dedicated forest root domains may in some cases cause too much unnecessary overhead. In a stable, highly centralized organization, a single domain forest in which the forest root domain and the main Account/Resource domain are the same would be appropriate. To execute a single domain configuration securely, the number of domain administrators (who could potentially add users to the Schema Admin or the Enterprise Admin groups) should be limited to only a few trusted administrators.

The section, "Special System Delegation", later in this document describes how to delegate control over single domain controllers using a delegation/GPO, eliminating the need for numerous domain administrators for rudimentary domain controller management.

## Design Alternatives

The following examples use Contoso, Ltd. as a sample company. The fictional organization Contoso, Ltd. has the ticker symbol Contoso and has a registered domain name Contoso.com.

### *Single Domain*

The single domain model represents the simplest design and should be the starting objective for domain planning.

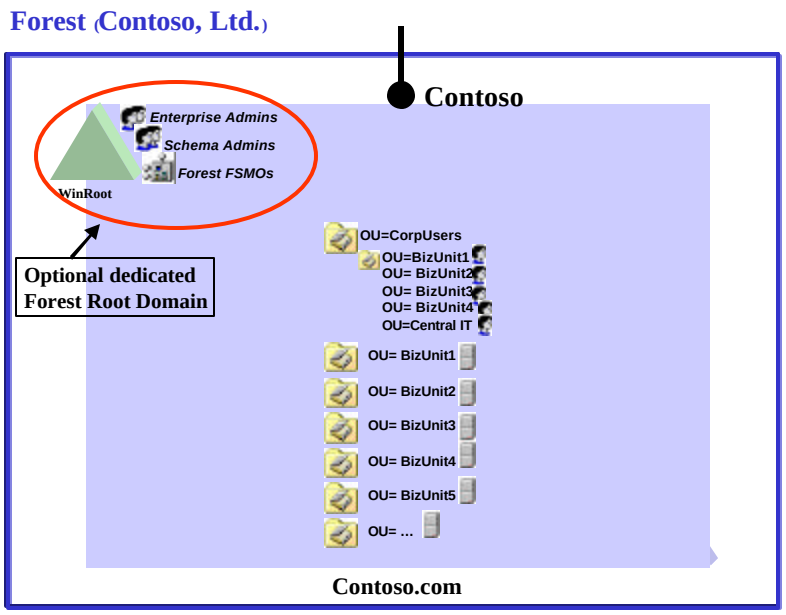



**Figure 3. A Single Domain Model**

The single domain model and the single domain model with a dedicated forest root provide the lowest total cost of ownership (TCO) implementation while at the same time providing for an administrative model that gives the autonomy that each department needs. Using the organizational unit delegation model, root-level containers can be established for each organization allowing the administrator full control to create users, groups, computers, sub-containers, and so on. In Windows 2003, organizational units have replaced the domains of Windows NT® 4 from delegated administration perspective. Windows 2003 domains are now used primarily for database partitioning. The single domain model is enormously beneficial to the enterprise because it allows the business units to have the rights to build what they need for their operations without being burdened by the overhead of maintaining "their portion" of the corporate directory. Now the unit can focus on its business objectives, saving the redundant effort and costs necessary to maintain a completely separate presence.

*Single Domain (+) & (-)*

The following are positive and negative aspects of the single domain model.

**(+)** It is very simple. Simple operations master management, replication, backup, recovery, Domain Name System (DNS), and so on.

**(+)** The database in Active Directory is everywhere. Any location with a domain controller can handle all authentication, GPO, certificates, and so on.

**(+)** A special dedicated forest root domain can be used to keep the Enterprise Administrators and Schema Administrators groups isolated in a separate domain (WinRoot) where the domain administrators of the main account domain (Contoso.com) do not have the rights to modify the membership of those two special groups.

**(+)** This model requires less hardware for Active Directory infrastructure. Hardware budget can be spent on application-type servers and not on authorization/directory servers. This setup also eliminates side-by-side domain controllers at each site for the different domains that are used there.

**(+)** The organizational unit delegation makes it easy to create/modify and to modify/delete.

**(+)** Users (and all objects) can be moved around very easily. (This action requires cloning domain-to-domain.)

**(+)** In the "true" single domain model (one without the dedicated forest root), the need for a separate global catalog database is eliminated because the single domain directory contains all the objects in the forest and can respond directly to all queries. Turning on Global Catalog for all Domain Controllers in a single domain can be done without incurring addition overhead, this is because the Active Directory itself contains ALL the objects in a single domain and serves a the sole source of information for the Global Catalog. This also provides Global Catalog (port 3268) responses for applications like Exchange that are explicitly looking to talk to a Global Catalog.

**(+)(-)** All data is replicated everywhere, making it readily available at any site. This replication may result in more local data than a small office needs to have at its site. (WAN replication calculations typically indicate that replication will be negligible.)

**(-)(+)** The design model is able to share common security settings, and forces all divisions to agree on parameters such as password length and complexity. This could actually be a positive aspect, depending on how security policies are viewed from a corporate perspective.

**(-)** This model exposes a single database to accidental or intentional harm. This risk is mitigated by the authoritative restore of individual objects and the administrative "walls" provided by organizational unit containers. In addition to being able to restore a pre-corruption copy of the database or some portion of it, there are background checks performed to prevent the introduction of an inconsistent database into the replication scheme. As part of scheduled backup and maintenance, a low-level integrity check can be run against the database to ensure the integrity, and semantics of the data in the recent backup (through the "integrity" command in the NTDSUTIL tool).

**(-)** Setup results in physical co-ownership of domain controllers (mitigated by security granularity and delegation). See the delegation section in, "Delegation of Administration," later in this document.

*Single Domain Situational Examples*

- Application developers only have to bind to one database for building enterprise–wide, directory-aware applications instead of having to add additional data to the global catalog or to run queries against multiple databases and build a concatenation process.
- Excessive numbers of hardware devices and configurations can become a long-term management issue with regard to BIOS upgrades, service packs, new releases, and cross compatibility. Maintaining the infrastructure with fewer, consistent configurations benefits everyone.
- Business units can focus effort that used to be spent on duplicate domain controller maintenance on building applications to enhance the productivity of their unit instead.
- While it is true that trusts within a forest are automatic and transitive, there are additional inter-domain operations, such as enterprise service account creations and security troubleshooting that makes multi-domain forests more complex to operate.
- Supporting applications that are put in place to route users to the right authentication mechanism can be eliminated, thus reducing the points of failure and the number of systems that need to be supported (remote access authentication is a common example).
- User experience is consistent throughout the world. The information technology (IT) team does not have to attempt to explain to users that the performance or feature availability is dependent on the local availability of one of their domain controllers.
- Efficient use of knowledgeable IT staff becomes very apparent when considering some of the esoteric topics the IT staff must know, such as operations master roles and site link management.
- Portions of Active Directory can be recovered without affecting the rest of the directory. This, for example, allows the authoritative restore of just a single container or leaf node.

### *General Summary*

While there are many examples that can be sited, and their individual weight may not be significant, the collective difference is very significant and directly impacts the efficient management of the base environment. This is not to say that multi-domain models are prohibitively costly. When a database needs to be partitioned for the right reasons, the consequences are naturally desired and accommodated. The right technical reasons for creating more than one domain can be reduced to two issues:

- Password length and restrictions, such as complexity and history, are domain-wide and are enforced on all users; therefore, all participants in the domain must agree on these parameters.
- WAN links cannot handle the inter-site replication traffic, and the database in Active Directory needs to be partitioned into smaller pieces.

Often, the reason for the creation of additional domains is political, commonly stemming from the current "ownership" of a Windows NT 4 domain or a misunderstanding of how the organizational unit structure and delegation model in Active Directory solves the problems that once had to be addressed with separate master account domains or resource domains.

### *Single Tree by Business Unit*

The single tree model adds some flexibility by decomposing the namespace and the database replicas into logical divisions, in Contoso's case, by using business unit boundaries. The tree shares a common root and a common schema (by being in the same forest). The Root Domain (Corp) provides a place to keep the forest roles.

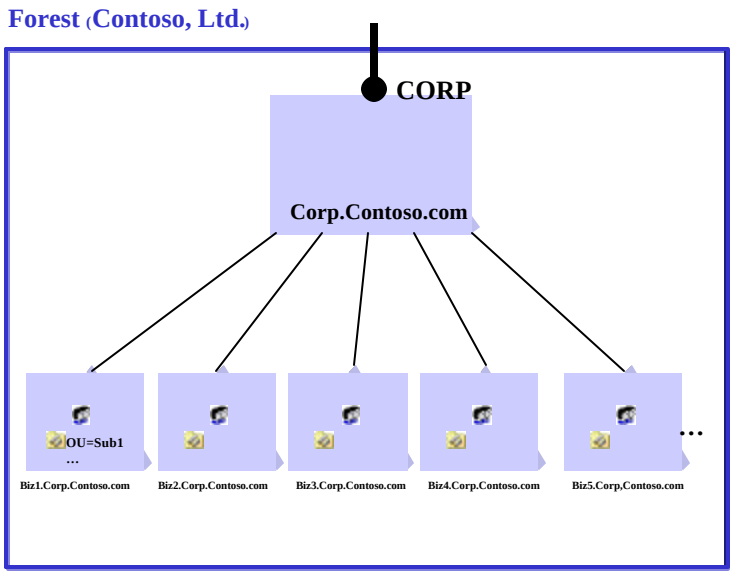



**Figure 4. The Single Tree Model**

**(+)** In this setup, the database is partitioned to accommodate small, poorly connected sites.

**(+)** It results in simple DNS delegation from existing corporate namespace to forest root.

**(+)** The root domain houses forest operations master, Schema Administrator, and Enterprise Administrator, while making a simple namespace root.

**(+)** Separate domain administrators for each domain give complete control for managing domain controller creation.

**(+)(-)** Each domain can have a separate security policy (password length, history, and complexity).

**(-)** This setup is not friendly to mobile users since it requires multiple domain controllers for local authentication at multiple sites.

**(-)** Frequent reorganizations require moving users between domains in a clone process, not a simple drag/drop. The reorganizations may also force costly domain restructuring efforts.

**(-)** More specialized administrators are needed (one for each domain that knows sites, operations masters, backup, and so on).

**(-)** It requires much more hardware for side-by-side domain controllers which are often used in sites where the user population is spread across multiple domains. This compares with the Single Domain that requires only a single domain controller to authenticate anyone in the forest.

**(-)** Default cross-realm authentication goes through the root domain. Shortcut trusts may be needed.

### *Single Tree by Geopolitical*

The single tree model adds some flexibility by breaking down the namespace and the replicas into logical divisions, by using geopolitical boundaries. The tree shares a common root, schema, and configuration container by being in the same forest.

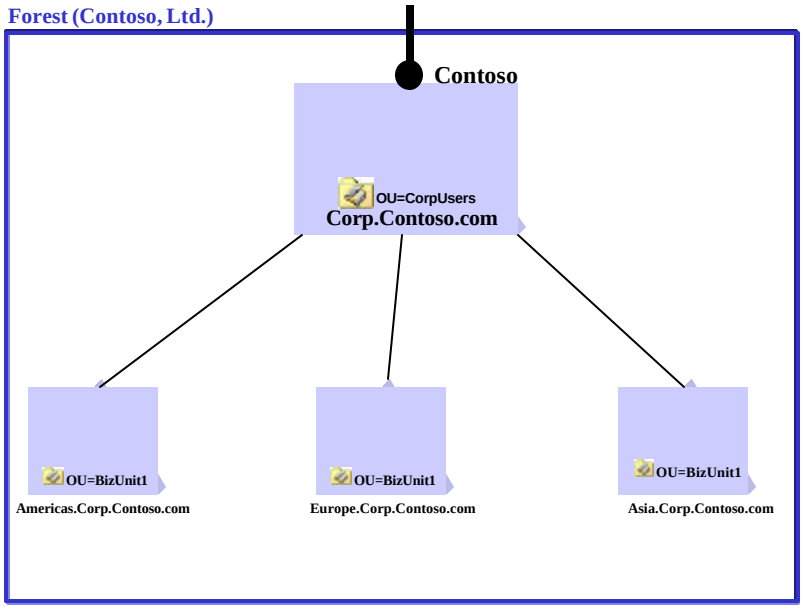



**Figure 5. The Single Tree using Geopolitical Boundaries**

The Root Domain (Corp) provides a place to keep the forest roles. Described below are positive and negative aspects of this model.

**(+)** The model partitions the database to accommodate small, poorly connected sites.

**(+)** It results in simple DNS delegation from existing corporate namespace to forest root.

**(+)** The root domain houses Forest operations master, Schema Administrator, and Enterprise Administrator, while making a simple namespace root.

**(+)** Stable boundaries, such as geopolitical boundaries, rarely change.

**(+)(-)** Each domain can have a separate security policy (password length, history, complexity, and so on).

**(-)** Organizational units have to be repeated for divisions that span continents.

**(-)** This setup is not friendly to mobile users, since it requires multiple domain controllers for local authentication at multiple sites. For example, if a BizUnit1 user from the Americas domain travels to Europe, he/she will have to have a local domain controller for Americas residing at the Europe site, or process the authentication across the WAN to a domain controller physically located in the Americas. This may give a different experience depending on which location the user is in, and the configuration of the infrastructure is no longer transparent to that user.

**(-)** Frequent reorganizations require moving users between domains which is a recreation of the user in the new destination domain, and not a simple drag/drop operation. Reorganizations may also force costly domain restructuring efforts.

**(-)** More specialized administrators needed (one for each domain that knows sites, operations masters, backup, and so on).

**(-)** It requires much more hardware for side-by-side domain controllers often used to accommodate users from multiple domains that may be sharing the same physical location. With a single domain, any Domain Controller can authenticate any user in the Forest.

**(-)** Default cross-realm authentication goes through the root domain when the request is between two trees in the forest. Default cross-real authentication passes through the parent domain within the individual tree structure. Shortcut trusts may be needed to eliminate the traversing between two "distant" domains that are actively sharing information.

*Multiple Tree*

The multiple tree model allows for a more distinct division, which may be used in cases in which separate business divisions want autonomy but also desire common schemas and easy security integration across the businesses.

In general terms, this model has the positives and negatives from the two single tree models and only really differs in the DNS delegation points from the existing corporate namespace. On the negative side, it is a slightly more complex DNS delegation from the corporate namespace require multiple points of delegation from the corporate root namespace.

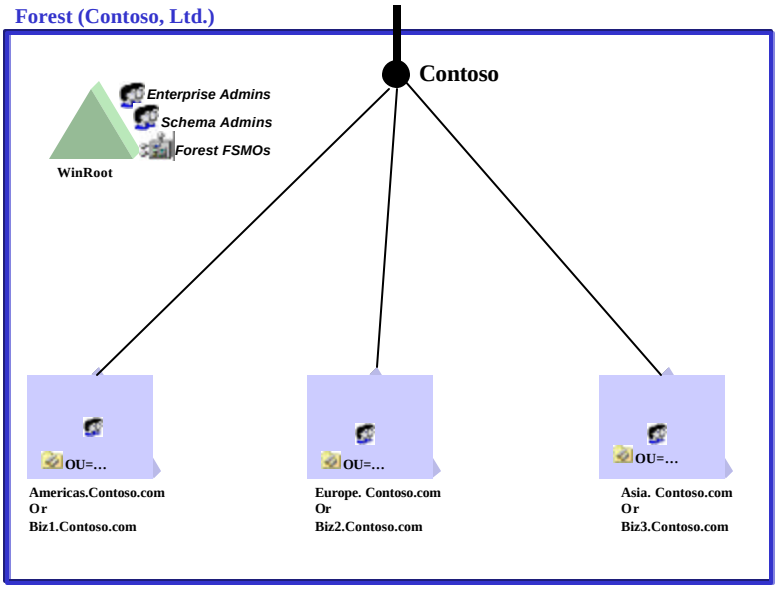



**Figure 6. Multiple Trees**

Domain administrators have three special roles that cannot be delegated through common delegation procedures.

1) **Create new domain controllers**. The administrator for the domain must be aware of all replicas created for the domain. Many other related functions are required when a new domain controller is created; for example, the proper site configuration must be established for the new domain controller. If this right was delegated, the environment may be exposed to security and configuration risks.

**Note** Domain controller replicas can actually be created by delegating the necessary rights and permissions on involved servers and directory containers.

This procedure is described in Chapter 11 "Delegating Tree/Forest Operations" in *Building Enterprise Active Directory Services: Notes from the Field*. The procedure described in that section involves direct sub-container security manipulation and should be approached with caution. Because of the reasons cited previously, the creation of new domain controllers is perhaps best managed by a central group of domain administrators. Once the domain controller is created, the administration and control of that server can be delegated.

The typical concern in the multiple tree design is the promotion of a new server at a remote site. This issue can be addressed in two ways:

- All domain controllers are created from a standard configuration and spares are ready to be shipped immediately.
- Administrators can connect to remote servers using Terminal Server Administrator Mode, and perform the Active Directory Installation Wizard remotely.

Secondary concerns involve shared applications on the server that is operating as a domain controller. This issue is addressed in many ways:

- Services like DHCP service and DNS have predefined administrative groups that can include non-Domain Administrative users.
- Individual services have security access control lists (ACLs), allowing permissions to be granted for stopping and starting individual services.
- The "server" aspect of a domain controller can be delegated separately from the "directory" aspect of a domain controller. (See the "Domain Controller Delegation" section later.)

2) **Create first level containers**. This is a desired effect so that the set of root containers can be established, and first-level delegation can be targeted to discrete containers.

3) **Take control of an object that it does not own**. This auditable right is required, since it is possible to "orphan" a container, or delegate exclusive rights to a user, then delete that user.

### *Situation and Requirements*

The Active Directory domain implementation at Purdue is to be as centralized as possible in a single domain with consideration taken for the specific needs of the different schools and departments in their need for their need for a dedicated domain within the forest.

Consideration for additional domains within the forest will be taken for the following situations:

1. Documented security requirement for stricter password requirements than the central domain.
2. Documented security requirements for stricter account lockout requirements.
3. Documented security requirement for a Kerberos authentication policy that differs from the central domain.

4. Documented Business requirements that would mandate the administrators of a specific school or department to have complete administrative control of their domain.

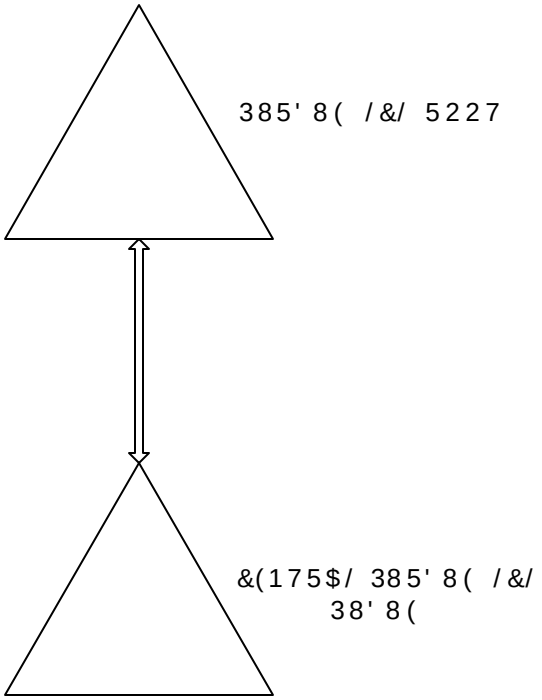
5. Physical network connectivity is extremely poor between the department location and the central domain. If the link is so poor that replicating central domain information cannot be processed over the WAN link between the two locations, a new domain may be required at the remote location so that only limited Global Catalog information is replicated.

### *Design Decisions*

The Purdue domain design will initially consist of a dedicated root domain called PURDUE.LCL with a NETBIOS name of ROOT. This domain will be used only for Enterprise level administrative tasks and services. Because of the secure nature of this domain it is the best place to host enterprise level services from such as Exchange 2003.

The central domain for all users and most computers and servers will be called CENTRAL.PURDUE.LCL with a NETBIOS name of PURDUE so that it can be easily identified as the central Purdue domain for all users.

Any other domains that are determined to be required will fall directly under the PURDUE.LCL domain to simplify the name space of the domain.



# DNS Namespace Design

A limited set of general alternatives is possible for connecting Active Directory to the Domain Name System (DNS) namespace. The first model involving the new-delegated node "Corp" provides the greatest feature set and has most the flexibility for the single domain and single tree domain designs.

## Design 1 - *Delegated Node to "Corp"*

The following diagram shows a sub-domain "Corp" delegated from the Contoso.com node in the corporate DNS namespace. The delegation uses the standard NS (name server) and A (host "glue") record to delegate authority to the Windows 2003 DNS server in the Corp.Contoso.com domain.

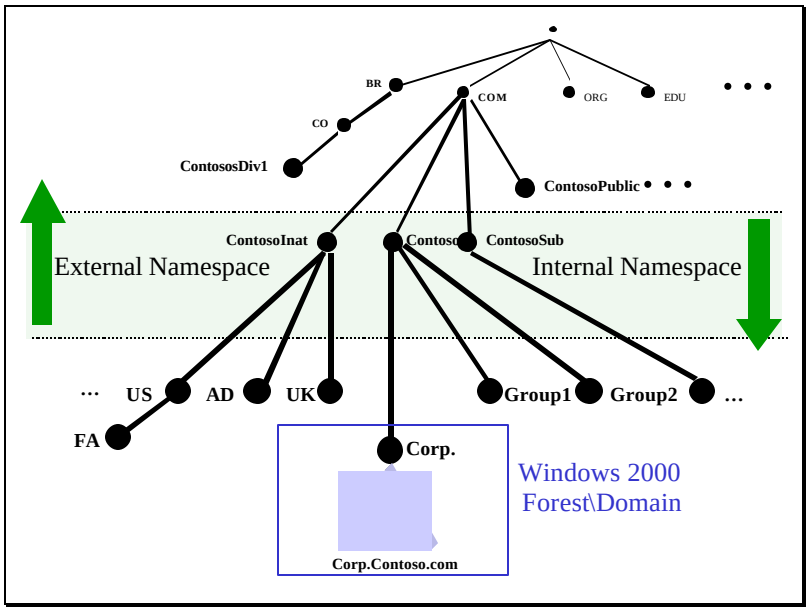



**Figure 7. Delegated Node to Corp**

The delegated node provides:

- Distinct separation between internal and external namespaces; that is, any host that is fully qualified with corp.Contoso.com is known to be an internal resource. This distinction assists in any proxy-type configurations in which logic is required to examine a fully qualified name and determine if it is an internal or external request.
- The ability to deploy a heterogeneous DNS environment where best-of-breed functionality can coexist at different points in the tree. For example, the multi-master replicated DNS database used in Active Directory can be used in the Corp namespace, and other legacy DNS systems can be used in the parent namespace.
- Reduction in risk for the overall environment. The delegated node demarks a point in which new components can be introduced without affecting the upper levels of the namespace. If the global user account domain is added to the Contoso node, then the entire DNS structure at that level needs to be upgraded before moving forward. This can impede progress on the Windows 2003 effort.
- Easier troubleshooting.
- A new neutral ground for all business units to join.
- The de-facto standard for the Microsoft Windows 2003 early adopters. This recommendation is also endorsed by other DNS vendors and can be found in their white papers on this topic.

*Common Issues*

The namespace will be deepened, and people are used to securing hosts with a fully qualified domain name (FQDN) that ends in Contoso.com.

- *Response*: The FQDN should be transparent to Windows users, since the directory will be used to locate resources, including a transparent routing to the replica that is nearest to the requestor. For resources that need to be accessed by non-Active Directory aware clients, those servers could be aliased in the Contoso.com domain, or some other business unit domain like ContosoInat or ContosoSub.
- *Response*: DNS suffix search order on the client should handle this.
- *Response*: People are also used to pursuing resources ending with other domains names like ContosoInat.com. Putting the domain at Contoso.com only solves FQDN issues to a point.
- *Response*: Creating aliases will not be too much overhead work. This involves only a manageable subset of Windows 2003 Servers that need to have special FQDNs. This level of management is being implemented today since dynamic update server registration is not in place yet.

### Design 2 - *Delegated Node to Contoso Tree*

From a DNS perspective, the single tree delegated to the "Corp.Contoso.com" has many of the same arguments as the previous example, but causes a further deepening of the namespace tree. The additional branches in this tree necessitate the installation of DNS servers that are authoritative for the new zones and could result in more DNS severs to handle the same number of records and requests, as compared with the single delegated node illustrated above.

**Note** The positives and negatives of this design from a Windows 2003 domain perspective are discussed in the preceding section. The DNS namespace design should follow from the Active Directory/Security design and not vice versa.

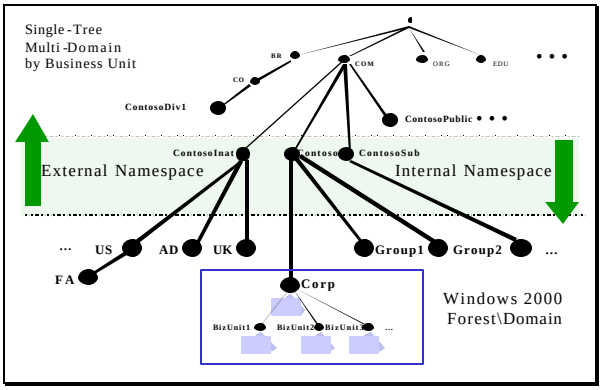



**Figure 8. Single Tree Multi-Domain by Business Unit**

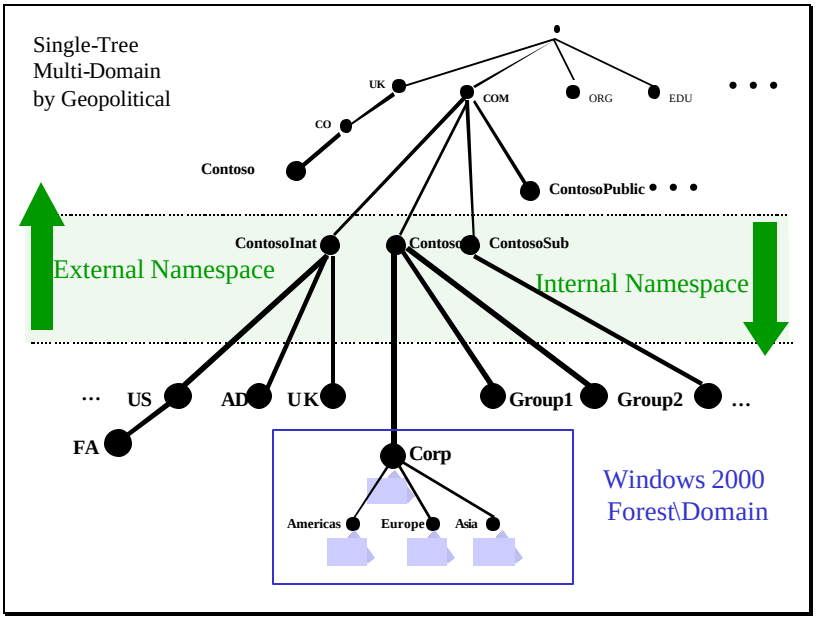



**Figure 9. Single Tree Multi -Domain by Geopolitical**

## Design 3 - *Rooted at Contoso*

This model mirrors Design1 earlier in this section, but instead of delegating a portion of the namespace to the Windows 2003 domain, the forest is moved to the upper section of the namespace, directly tied to the Contoso.com location.

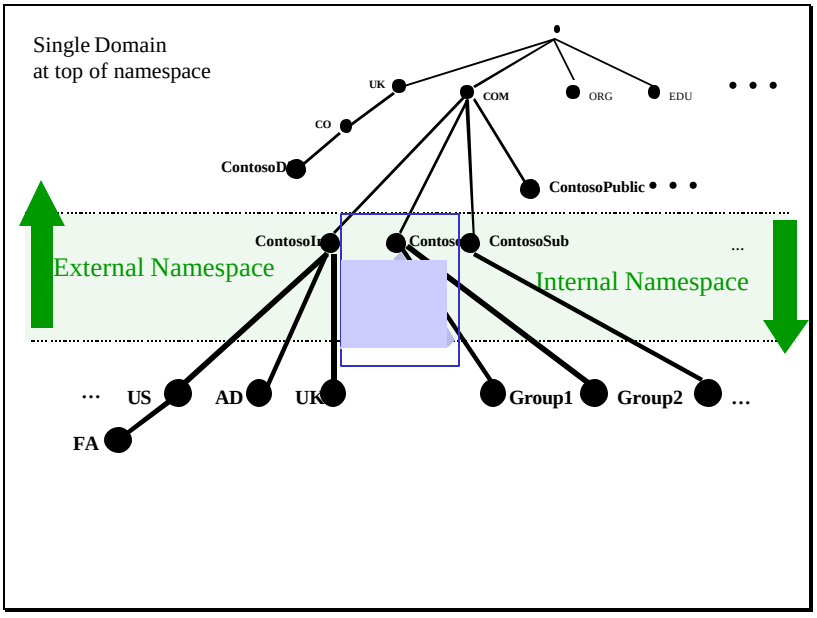



**Figure 10. Single Domain at the Top of Namespace**

The design depicted in the previous figure would require the upgrade of the entire Contoso.com DNS domain to accommodate the Windows 2003 requirements. If this upgrade cannot be performed, the deployment of Windows 2003 will halt. This design also prevents the best-of-breed approach where the Windows 2003 DNS server (providing multi-master, secure update and so on), could be used in a child domain while another DNS system could be used in the parent domain.

If Contoso.com can be upgraded without any issues to Windows 2003 DNS or another compliant DNS, then the risk is eliminated.

The delegated node design at the start of this section allows the design and deployment team to proceed with multiple options and clear fallback positions. For example, the child Windows 2003 domain could switch between a third-party BIND 8.2.2 implementation and Windows 2003 DNS without impacting the parent structure. The above configuration does not allow this and presents one of the greatest risks to a successful deployment.

See the preceding section, "Design 1 – Delegated Node to "Corp"," for contrasting points.

### Design 4 - *Delegated Direct to Business Unit*

The following two DNS models follow the same delegated node arguments as noted previously. The distinguishing points of these designs are the number of different sub-domains that are required to anchor each of the Business Unit domains. Although it is not pictured here, the geopolitical partitioning shown in the single tree model above could be applied in the same manner.

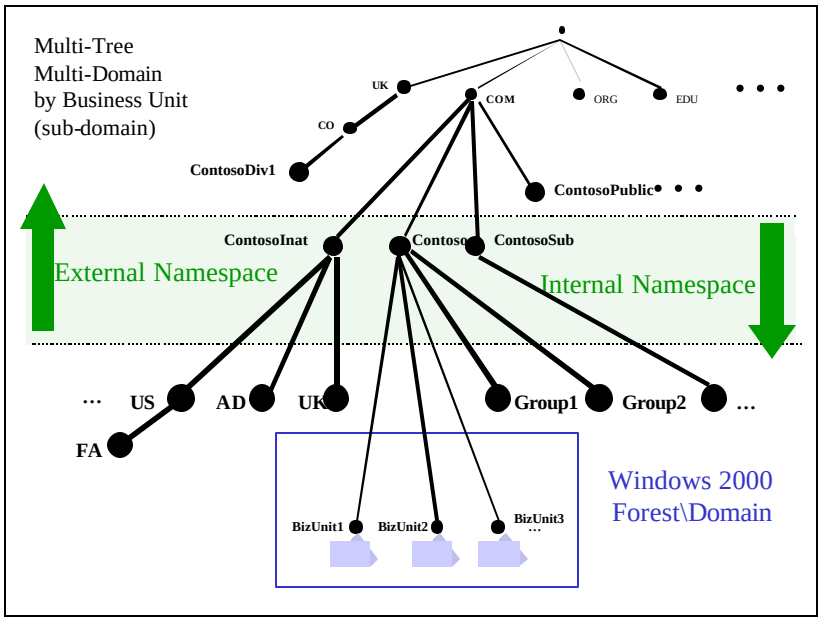



**Figure 11. Multi-Tree Multi-Domain by Business Unit (sub-domain)**

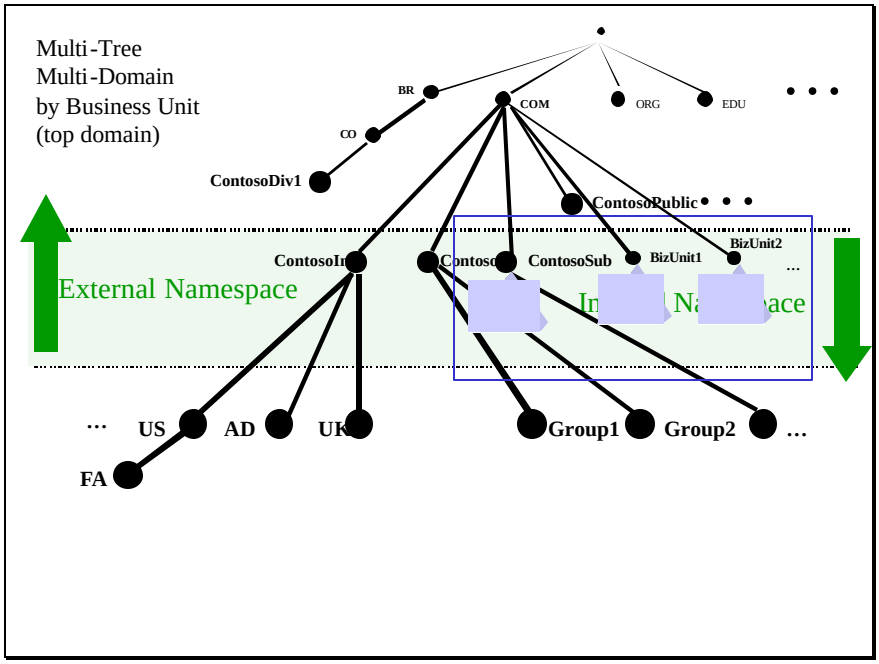



**Figure 12. Multi-Tree Multi-Domain by Business Unit (top domain)**

See the preceding discussion on "Rooted at Contoso" for objections to mixing with preexisting namespaces.

### *Design 5 – New Registered Name*

The "New Name" alternative requires that the name is available to be registered as a valid Internet name. In this case Contoso, Ltd. has registered Contoso.net. This name can be kept as an internal-only name, and this solution carries the same benefits as the Delegated Corp node in Design 1, with the only difference being that the FQDN is one node shorter. This solution requires that the name be registered and the necessary fees paid and maintained. In addition, the DNS administrators want to be sure to short-cut the natural hierarchical name resolution process because the natural flow for name resolution from Contoso.net to Contoso.com travels through the Internet "." zone, causing the undesired side effect of having the lookup request exit out to the Internet and re-enter the corporate network. This shortcut can easily be achieved with the addition of a name server (NS) record that publishes a name server for the internal peer sub-domain.

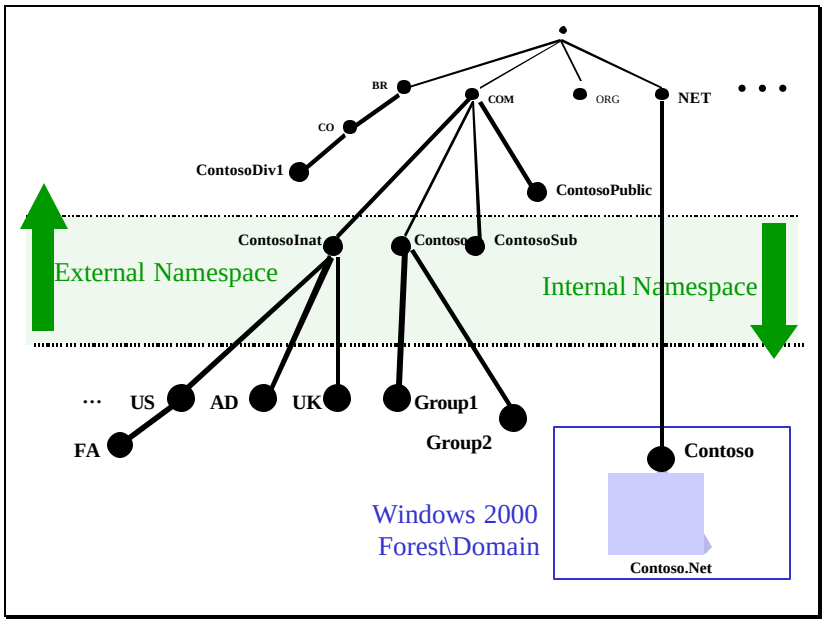



**Figure 13. New Registered Name**

### *Design 6 - Private Namespace*

The private namespace model is a complete departure from the other designs and demonstrates the ability to build the Windows 2003 environment on a DNS namespace that has no direct relevance to the corporate namespace. This abstract namespace would be nearly invisible to the end user. Through a simple zone transfer and forwarder, this independent namespace can be "hooked" to any parent namespace. This model is an interesting example, and may become a more widely used model as the concept of a private namespace evolves similar to the reserved names in RFC2606 (.test, .invalid, .example, or .localhost).

For additional information on RFC 2606, please visit the following Web site: http://ietf.org/rfc/rfc2606.txt?number=2606.

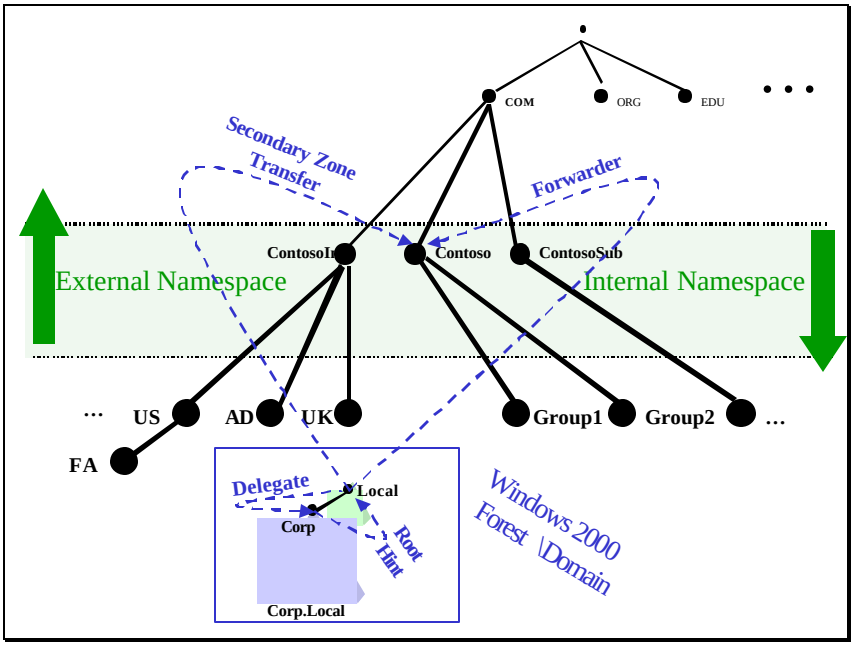



**Figure 14. Private Namespace Model**

### *Special Considerations for Locator Records*

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

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

If the DNS servers have persistent fast connections to the DNS servers authoritative for the _msdcs.<*DNS forest-name*> domain, then no special configuration is needed. If not, there are two options. You can replicate the entire forest root zone to all DNS servers in the enterprise using standard zone transfer, or you can create a separate zone called _msdcs.<*DNS-forest-name*> and replicate that zone to every DNS server. If you are using Active Directory-integrated DNS, you can place the primary copy of this zone in the forest root domain along with the <*DNS-forest-name*> zone. You can then replicate the zone to secondary DNS servers outside the domain using standard DNS replication. The domain controllers or DNS servers in non-root domains will host read-only copies of the source zone.

If a branch office is connected by dial-up lines to the hub sites, and the branch office contains either one global catalog server or at least two domain controllers for the same domain, then there should be a DNS server in the branch office which hosts a secondary _mscdcs.<*DNS-forest-name*> zone. The forest-wide records are then available locally, and there is no need to generate WAN traffic in this instance.

If a DNS server does not have a local copy of the _msdcs.<*DNS-forest-name*> zone and is not configured to forward queries, it must use DNS recursion to look up a name in that zone. For a DNS server to perform recursion, it contacts a DNS server that is authoritative for the root of the namespace (a DNS root server) and proceeds down the delegations in DNS until it finds the record in question. If there is no DNS root server or forwarder in a site, and the links between that site and other sites are down, a DNS server cannot perform recursion or forwarding. Thus, it will not be able to find any DNS servers that are authoritative for _msdcs.<*DNS-forest-name*>, even if those DNS servers are in the same site. The solution is to replicate the zone to two DNS servers in the site, and configure each to point to the other as its preferred DNS server and point to itself as the alternate.

### *The Island Problem*

The so-called “island" problem occurs when a domain controller that is the primary DNS server for the domain points to itself as the preferred or alternate DNS server for the zone _msdcs.<*DNS forest-name*>. The typical scenario for this configuration is a domain controller in the forest root domain, which is also a DNS server using Active Directory-integrated DNS. In the case of a configuration change, such as changing the IP address of the domain controller, the update of the forest-wide locator records only happens on the local domain controller. Normally, this change is replicated to all other domain controllers.

The CNAME record, (one of the domain controller locator records) is used for finding the IP addresses for replication partners. Replication internally uses the globally unique identifier (GUID) of domain controllers to find replication partners. This CNAME record of a domain controller uses the GUID as an alias name mapped to the fully qualified DNS name of the domain controller.

If a domain controller changes its configuration in a way that affects the CNAME and only updates this change in its own local DNS server, then this change never reaches the other domain controllers. The reason is that if Active Directory-integrated DNS zones are used, DNS relies on replication to update DNS zone information. However, if domain controllers cannot find replication partners due to changes in DNS, the changes cannot be replicated.

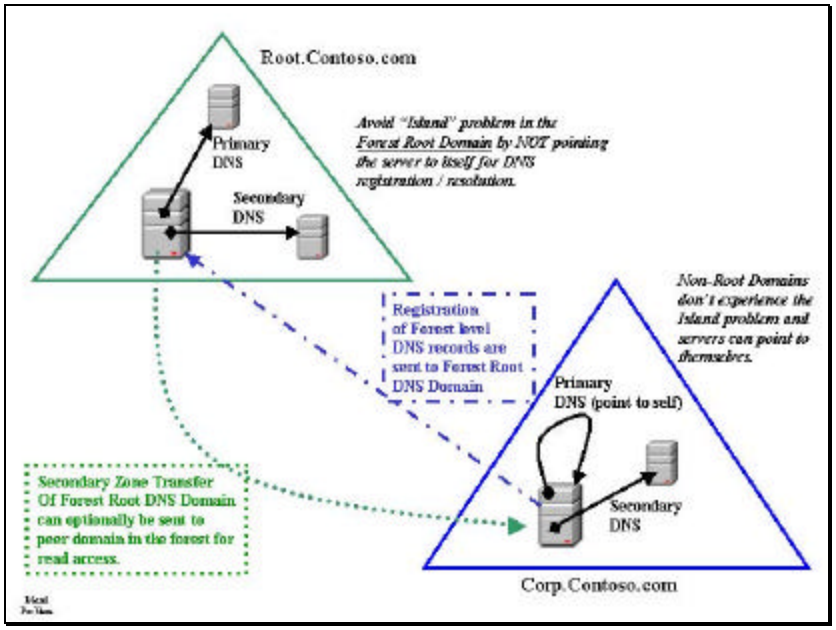



**Figure 15. An example of an Island Problem**

To avoid this scenario, a primary server in the forest root domain should not point to itself as the preferred DNS server if the forest root domain controllers are DNS servers authoritative for the _msdcs.<*DNS forest-name*>. The primary server should point to another forest root DNS server. Domain controllers in all other cases will not be affected by the "island" problem, because even if they point to themselves as preferred DNS servers, they will always update the records needed for replication on a DNS server that is authoritative for the root zone.

### *Configuration to Avoid "Island" Problems*

If a domain controller points to itself as preferred or alternate DNS server, and if the local DNS server is the primary server for the DNS domain _msdcs.<*DNS forest-name* >, (that is, it is in the root domain), it will be necessary to reconfigure its preferred DNS server prior to restarting it. After promotion, but before restarting, an administrator should configure the domain controller with preferred and alternate DNS servers that are primary for the domain _msdcs.<*DNS forest-name*>, but not with itself. Any other domain controller in the forest can be configured to point to it as a DNS server.

### *Configuration of DNS SRV Records Published by Net Logon*

Clients search for a nearby domain controller using site-specific domain controller locator records. Only if there are no domain controller entries for a client's site will the client query for and accept other domain controllers. By default, each domain controller publishes its site-specific and generic service resources records (SRVs).

### *Finding a Domain Controller in the Hub*

In the branch office scenario, it is important for clients that cannot find a domain controller in their own site to find a domain controller in their hub sites and never a domain controller in another branch or hub. In many deployments, clients from one branch cannot connect to computers in another branch office because the network is not fully routed (for example, one-way dial-up lines are used). Even if connectivity is possible, however, it is still undesirable to initiate network connections between branches. Such network traffic would always go through the hub site; therefore, it is better to restrict the traffic to branch-to-hub only.

To avoid the situation where clients in one branch contact a domain controller in another branch, the Net Logon service on all branch office domain controllers must be configured to publish only site-specific locator records, not generic domain controller locator records. The result is that only the hub domain controllers publish the generic locator records in addition to their site-specific records. Clients that cannot find a domain controller in their own sites only find generic domain controller locator records for hub domain controllers. For details on branch office deployments, see the white papers on the Resource information folder or visit the Active Directory Branch Office Planning Guide Web site at: http://www.microsoft.com/windows2003/techinfo/planning/activedirectory/branchoffice/dply09.asp

## DNS Delegation

### *Complete Sub-Domain*

Typically, organizations have an existing DNS structure. That structure may be a Windows-based DNS structure, or it may be a basic BIND implementation. The following models show a delegation from a "foreign" system. The same principles apply if it were a complete Windows 2003 DNS structure as well. The following scenarios depict the configurations of common heterogeneous environments.

The delegation model can be used to enable the use of heterogeneous best-of-breed DNS services that may differ between the upper half and lower half of the namespace. Third-party products have been developed that provide an enterprise management system that will encompass different back-end systems.

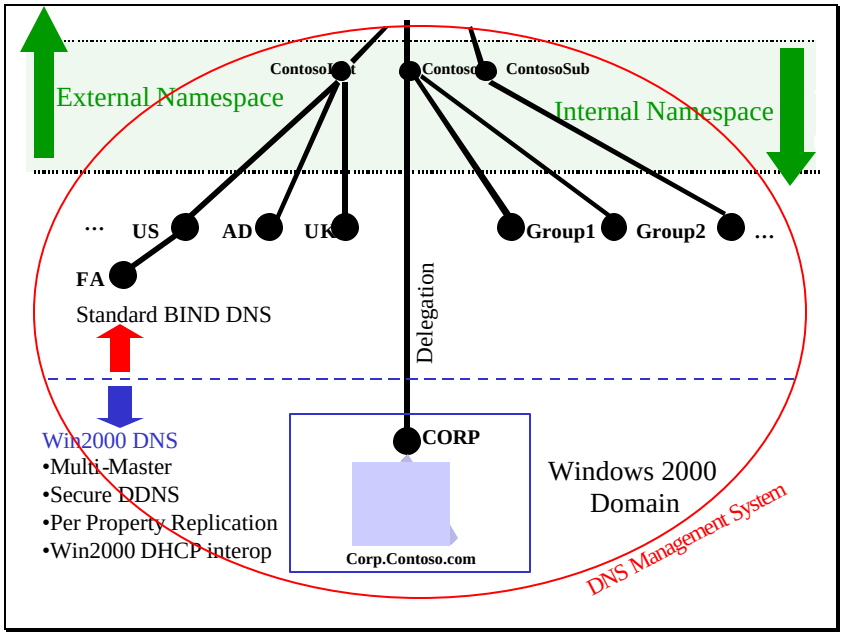



**Figure 16. DNS Management System**

*Delegation of Locator Record Domains*

The following model shows the delegation of the sub-domains handling the server locator record sub-domains. This model allows for the use of Windows 2003 multi-master, widely distributed DNS servers for the management of locator records used to find services in the domain. In this example, the A resource records would still need to be registered in the Corp.Contoso.com domain.

While this model may seem interesting to organizations that do not want to delegate a node to the Windows 2003 DNS, it does add complexity and possible confusion to the namespace.

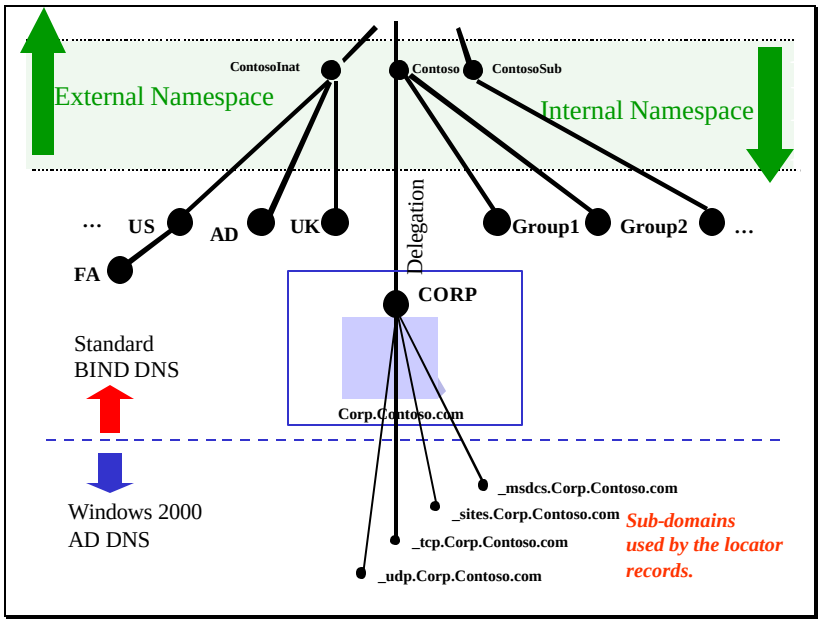



**Figure 17. Delegation of Locator Record Domains**

## *Physical Implementation*

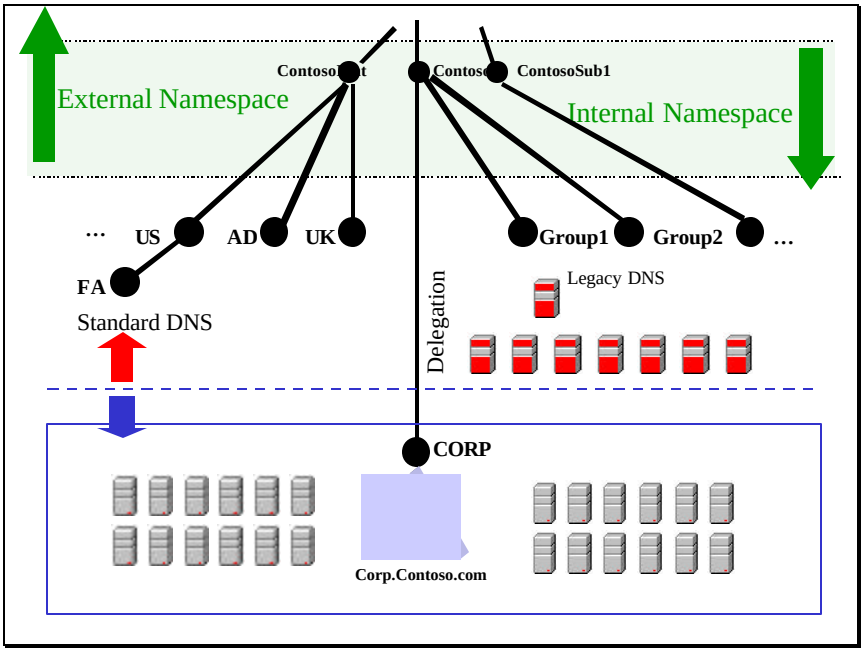



**Figure 18. Physical Implication**

Physical implementation positions the Microsoft Active Directory-integrated DNS as the name service for the Windows 2003 environment, taking advantage of the fact that every Windows 2003 domain controller can become a master DNS server, since Active Directory integration automatically distributes the DNS records along with the other Active Directory information. This allows DNS simply to be turned on at any domain controller.

The designers of the Microsoft Windows 2003 operating system chose DNS as the name service for the operating system. Windows 2003 Server includes an Internet Engineering Task Force (IETF) standard-based DNS server. Because it is RFC (request for comment) compliant, it is fully compatible with any other RFC compliant DNS servers. The use of the Windows 2003 DNS server is not mandatory. Any DNS server implementation supporting Service Resources Records (SRVs, as described in the Internet draft *A DNS RR for specifying the location of services (DNS SRV)*) and dynamic update (RFC2136) is sufficient to provide the name service for Windows 2003-based computers. However, because this implementation of DNS is designed to take full advantage of the Windows 2003 Active Directory service, it is the recommended DNS server implementation for any networked organization with a significant investment in Windows or extranet partners with Windows-based systems. For example, while conventional DNS servers use single-master replication, Windows 2003 DNS can be integrated into Active Directory service, so that it uses the Windows 2003 multi-master replication engine. (Note that the directory service supports multi-master replication.) In this way, network managers can simplify system administration by not having to maintain a separate replication topology for DNS.

For additional information, refer to RFC 2606 at: http://www.ietf.org/rfc/rfc2606.txt?number=2606.

DNS in Windows 2003 provides a unique DNS server implementation that is fully interoperable with other standards-based implementations of DNS server.

### *Active Directory Storage and Replication Integration*

In addition to supporting a conventional way of maintaining and replicating DNS zone files, the implementation of DNS in Windows 2003 has the option of using Active Directory services as the data storage and replication engine. This approach provides the following benefits:

- DNS replication will be performed by Active Directory service, so there is no need to support a separate replication topology for DNS servers.
- Active Directory service replication provides per-property replication granularity.
- Active Directory service replication is secure.
- A primary DNS server is eliminated as a single point of failure. Original DNS replication is single-master; it relies on a primary DNS server to update all the secondary servers. Unlike original DNS replication, Active Directory service replication is multi-master; an update can be made to any domain controller in it, and the change will be propagated to other domain controllers. In this way, if DNS is integrated into Active Directory service, the replication engine will always synchronize the DNS zone information.

Thus Active Directory service integration significantly simplifies the administration of a DNS namespace. At the same time standard zone transfer to other servers (non Windows 2003 DNS servers and previous versions of the Microsoft DNS servers) is still supported.

For complete details on the Windows 2003 DNS server and RFC compliances see the *Windows 2003 DNS White Paper* at:
http://www.microsoft.com/windows2003/techinfo/howitworks/communications/nameadrmgmt/w2kdns.asp

### *Allocation and Registration (Mixed Clients)*

Windows 2003 DNS and DHCP (Dynamic Host Configuration Protocol) servers work together in a mixed environment to provide a standards-based dynamic host IP environment for diverse client types.

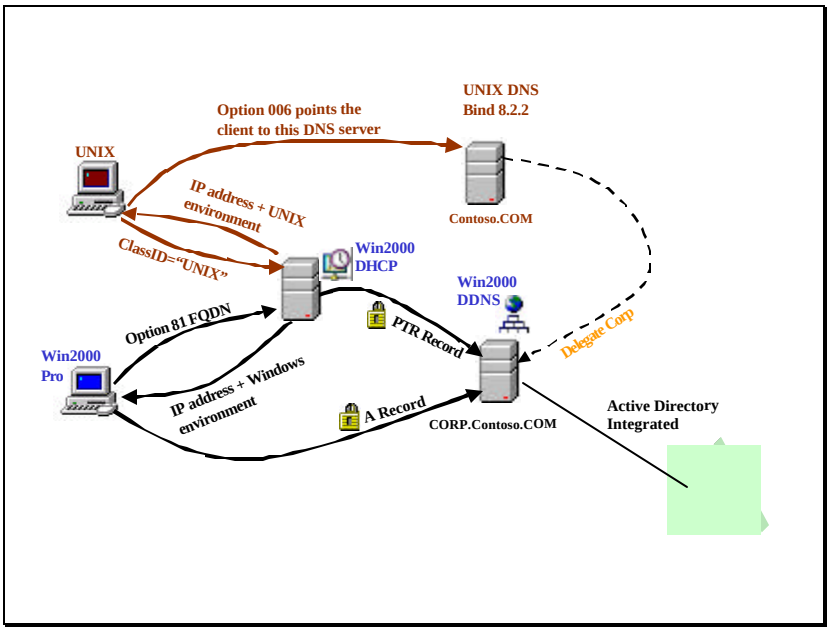



**Figure 19. Mixed Client Environment**

The previous configuration describes the allocation and registration process for a mixed client environment in which UNIX workstations belonging to another DNS domain are able to get distinct information by ClassID, thereby redirecting their updates and queries to a different DNS server and domain. Windows 2003 performs the standard "A" resource record registration while the DHCP server manages the registration of the Pointer (PTR) resource record.

### *Allocation and Registration (Back-Level Windows Clients)*

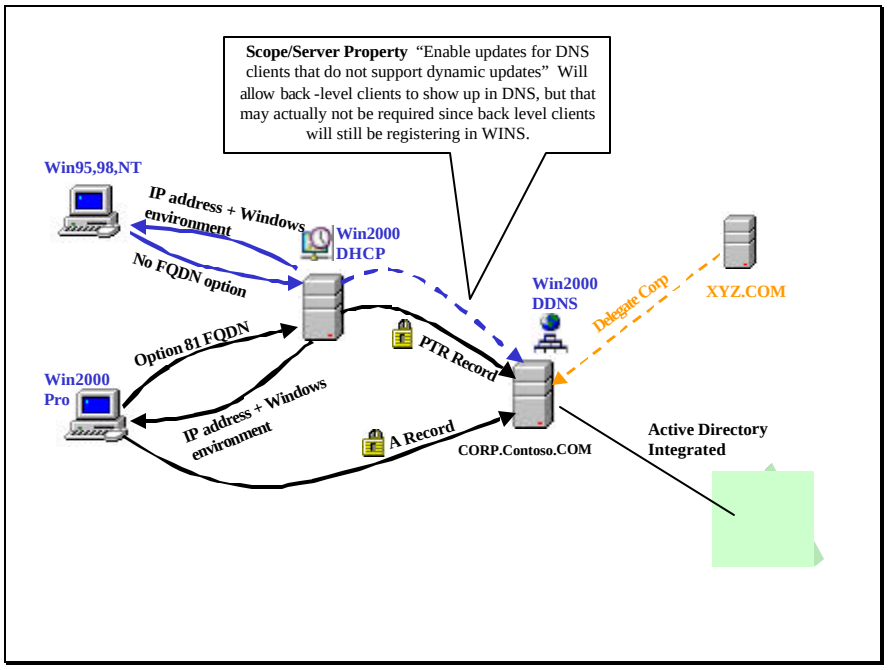



**Figure 20. Scope / Server Property**

This configuration shows the allocation and registration processes for Windows 2003 Professional and back-level windows clients. The Windows 2003 workstation performs the default registration process where the workstation registers the "A" record and the DHCP server registers the "PTR" record. The back-level client, lacking the Option 81 information, leaves DHCP to perform both record updates ("A" and "PTR"). For more information on Option 81, see *Dynamic Host Configuration Protocol for Windows 2003 Server* on: http://www.microsoft.com/windows2003/techinfo/howitworks/communications/nameadrmgmt/dhcp.asp

### *Dynamic Update Detailed*

Windows 2003 clients send dynamic updates for three different types of network adapter configurations: DHCP adapters, statically configured adapters, and remote access adapters. Regardless of which adapter is used, the DHCP Client service sends dynamic updates to the authoritative DNS server.

> **Note:** The DHCP Client service runs on all computers regardless of whether they are configured as DHCP Clients.

By default, the dynamic update client dynamically registers its "A" resource records and possibly all of its "PTR" resource records every 24 hours or whenever any of the following events occur:

- The TCP/IP configuration is changed.
- The DHCP address is renewed or a new lease is obtained.
- A Plug and Play event occurs.
- An IP address is added or removed from the computer when the user changes or adds an IP address for a static adapter. (The user does not need to restart the computer for the dynamic update client to register the name-to-IP address mappings.)

By default, the dynamic update client automatically de-registers name-to-IP address mappings whenever the DHCP lease expires. The client can be configured not to register its name and IP address in DNS. If you configure the client not to automatically register name-to-IP address mappings, the DHCP server is running Windows 2003, and it is configured to register DNS resource records on behalf of clients that are running versions of Windows earlier than Windows 2003, the DHCP server attempts to update the mappings instead.

**Note** Re-registration can be forced by running **ipconfig /registerdns**.

### *Aging and Scavenging*

With dynamic update, records are automatically added to the zone when computers and domain controllers are added. However, in some cases, they are not automatically deleted.

Having many stale resource records presents a few different problems. Stale resource records take up space on the server, and a server might use a stale resource record to answer a query. As a result, DNS server performance suffers.

To solve these problems, the Windows 2003 DNS server can *scavenge* stale records by searching the database for records that have aged and deleting them. Administrators can control aging and scavenging by specifying the following:

- Which servers can scavenge zones?
- Which zones can be scavenged?
- Which records must be scavenged if they become stale?

The DNS server uses an algorithm that ensures that it does not accidentally scavenge a record that must remain, provided that you configure all the parameters correctly. By default, the scavenging mechanism is disabled. Do not enable it unless you are absolutely certain that you understand all the parameters. Otherwise, you might accidentally configure the server to delete records that it should retain. If a name is accidentally deleted, not only do users fail to resolve queries for that name, but also any user can create that name in DNS and then take ownership of it, even on zones configured for secure dynamic update.

You can manually enable or disable aging and scavenging on a per-server, per-zone, or per-record basis. You can also enable aging for sets of records by using Dnscmd.exe. Keep in mind that if you enable scavenging on a record that is not dynamically updated, the record will be deleted if it is not periodically refreshed, and you must recreate the record if it is still needed.

If scavenging is disabled on a standard zone and you enable scavenging, the server does not scavenge records that existed before you enabled scavenging. The server does not scavenge those records even if you convert the zone to an Active Directory-integrated zone first. To enable scavenging of such records, use Dnscmd.exe.

### WINS

Windows Internet Name Service (WINS) is necessary for an extended period of time until the entire environment is upgraded, and all NetBIOS (network basic input/output system) dependencies are eliminated. Various existing or "legacy" applications may require NetBIOS for proper operation. The current WINS configuration works well and doesn't need to be changed. The WINS servers can be upgraded to take advantage of new WINS features, performance, and stability enhancements.

> **Note** Using the Windows DNS server, the WINS database can be "hooked" into DNS, allowing the DNS database to be augmented by the host records registered in the WINS database.

### DHCP

Windows 2003 includes an enhanced implementation of DHCP. This includes integration of DHCP with domain name system (DNS), enhanced monitoring and statistical reporting for DHCP servers, new vendor-specific options and user-class support, multicast address allocation, clustering, and rogue DHCP server detection. DHCP for Windows 2003 is open and based on industry standards, supporting Requests for Comments (RFCs) 2131 and 2132.

For complete details on Windows 2003 DHCP see, *Dynamic Host Configuration Protocol for Windows 2003 Server* on:
http://www.microsoft.com/windows2003/techinfo/howitworks/communications/nameadrmgmt/dhcp.asp

## Situation and Requirements

### *DNS*

Secure DNS is extremely important to Purdue University, so all DNS servic es that support Windows 2003 Active Directory will be implemented in a secure fashion. There are currently BIND DNS servers which host the public name space of Purdue.edu which the Windows 2003 DNS architecture will not replace.

There is a perception at Purdue that all Dynamic DNS (DDNS) is not secure. Because of this perception there has been great push back against using DDNS to support Active Directory. While it is true that DDNS technology by itself is not secure and any host could potentially create host records, in Windows 2003 Active Directory there is a specific feature of securing DDNS by creating Active Directory Integrated DNS Zones so that only specific computers can write dynamic records.

In Windows 2003 there are three types of DNS zones:

1. Standard Primary
2. Standard Secondary
3. Active Directory Integrated

**Standard Primary Zones**

Standard Primary DNS Zones in Windows 2003 DNS are the same as Primary DNS zones in any other type of DNS server. The Windows 2003 DNS server hosting the Primary zone is the only server that can make modifications to this static DNS zone. The DNS records are kept in a dedicated DNS database that is on the server. This zone can be designated to allow Secondary copies of the zone by replication to other DNS servers.

**Standard Secondary Zones**

Standard Secondary Zones in Windows 2003 DNS are the same as Secondary zones in any other type of DNS server. The Windows 2003 DNS server hosting the Secondary zone will replicate this zone from the designated Primary zone server, and can act as a Name Server for this zone. The DNS records

for these zones are kept in a dedicated DNS database that is on the server. A server hosting a Secondary zone cannot make any record changes to the zone.

**Active Directory Integrated Zones**

Active Directory Integrated Zones are unique to Windows 2003 and Active Directory. When a zone is designated as being Active Directory Integrated, the records for this zone are not kept in the DNS database on the server, but are instead moved into the Active Directory, and are tracked as Active Directory Objects. This gives the DNS servers in a domain the ability to have a Multi-Master DNS zone where changes to the zone can be made by any Active Directory Domain Controller that is running the DNS service in the domain.

Because the DNS records in an Active Directory Integrated zone are in fact Active Directory objects, access control lists (ACLs) can be applied to the zone which gives you the ability to secure the zone and allow only specific computers access to it. When leveraging this ability with DDNS you can ensure that only the machines that you have designated will be able to add new or manage existing records dynamically.

Using an Active Directory Integrated zone also provides automatic replication of the zone to all of the domain controllers because the DNS records are actually Active Directory objects and will replicate based on the existing replication topology so that no other DNS replication management needs to occur.

### *WINS*

Even though DNS will support all aspects of Windows 2003 Active Directory, the Windows Internet Name Service (WINS) is still required to support down level clients as well as line of business applications that have not been designed to use Fully Qualified Domain Names (FQDN) when accessing network resources.

### *DHCP*

Central management of DHCP services throughout Purdue University is currently not in the scope of the design, and will continue to be managed in the same way that it is currently being managed.

In the areas that DHCP is being run on Windows 2003 Servers, special consideration needs to be taken to the DHCP configuration in how it relates to DNS. By default, Windows 2003 DHCP in Active Directory is configured to create dynamically updated DNS records for its DNS clients. Because of the secured nature in which DNS will be implemented at Purdue, this feature should be disabled at the DHCP server so that only authorized servers are allowed to update records in DNS.

## *Design Decisions*

### *DNS*

In order to reduce the administrative overhead and possible errors in manually entering all DNS records to support Active Directory, DDNS will be implemented in a secure fashion to ensure that only authorized machines can update their corresponding host and Active Directory service records.

#### Root Domain

The root domain will consist of at least 2 domain controllers. Each of these domain controllers will run the DNS service and host the PURDUE.LCL as an Active Directory Integrated DNS ZONE. The Primary DNS server IP setting on each of these servers will point to the other server in the root domain to prevent DNS "Islanding" of the root domain.

A Universal group will be created which will be applied to the DNS zone PURDUE.LCL so that only members of that group will be allowed to make dynamic changes to the DNS Zone. The DNS zone

PURDUE.LCL will be configured to allow "Only Secure Updates" so that authentication MUST take place between the domain controller and the computer requesting the update.

The Enterprise Administrator will add each of the Child Domain "Domain Controllers" Global Group into the Universal Group that is applied to the PURDUE.LCL zone so that each of the child domain controllers can update site and global catalog service records in the root domain.

The ROOT domain DNS servers will have their DNS forwarders configured to point to the central DNS servers that host the PURDUE.EDU name space. This will cause all DNS requests that the DNS servers are not authoritative for to be resolved by the University DNS servers.

### Child Domains

Each child domain will consist of at least 2 domain controllers. In order to support DNS for the domain the domain administrators must decide whether to have the ROOT domain host their DNS, or whether they want to host the DNS zone for their own domain.

1. Hosted at the ROOT domain

If the domain administrators choose to have the DNS hosted in the ROOT then before the domain is implemented the Enterprise Administrator will pre-stage the new domains DNS zone on the ROOT domain controllers DNS. The Zone will be Active Directory Integrated and will be secured similarly to how the PURDUE.LCL zone is secured except that only the "Domain Controllers" group of the new domain will be allowed to make dynamic updates in the zone.

Each of the domain controllers in the child domain will configure their IP settings to point to the DNS servers in the ROOT domain.

2. Hosted in child domain

If the domain administrators choose to host their own Active Directory DNS then before the domain is implemented the Enterprise Administrator will pre-stage a "Delegated DNS Zone" in the ROOT DNS so that the child domains records can be resolved from anywhere in the forest.

Once the new child domain is brought online the DNS zone for that domain will be Active Directory Integrated and configured to allow "Only Secure Updates" and the ACL will be configured to allow only the required servers to be able to dynamically update the zone.

Each of the DNS servers must configure their DNS Forwarder to forward all DNS requests for which the DNS server is not authoritative for to the ROOT DNS servers. This will allow the child domain DNS servers to resolve DNS records in other zones in the Active Directory, as well as external or internet records.

### *WINS*

A single WINS architecture will be implemented to support all domains within the Active Directory. Initially the WINS service will run on 2 servers in the CENTRAL.PURDUE.EDU domain configured in a push/pull replication. Using two servers provides the redundancy required for this important name resolution service. Each computer in the Active Directory will be configured to register with these WINS servers. Because the WINS will be centralized, special consideration must be taken in assigning machine names as they must be unique throughout the WINS database even if they are in different domains.

# Organizational Unit Structure

## *Purpose of Organizational Units*

Organizational units are containers used to organize the objects within a domain. Important points to consider about the use of organizational units:

- Organizational units can be nested.
- Organizational units can be used to delegate administration.
- Organizational units are not security principals.
- Group Policy can be applied to an organizational unit.
- Organizational unit naming conventions and tree structures are flexible and relatively simple to change.
- Users will typically not navigate the organizational unit structure.

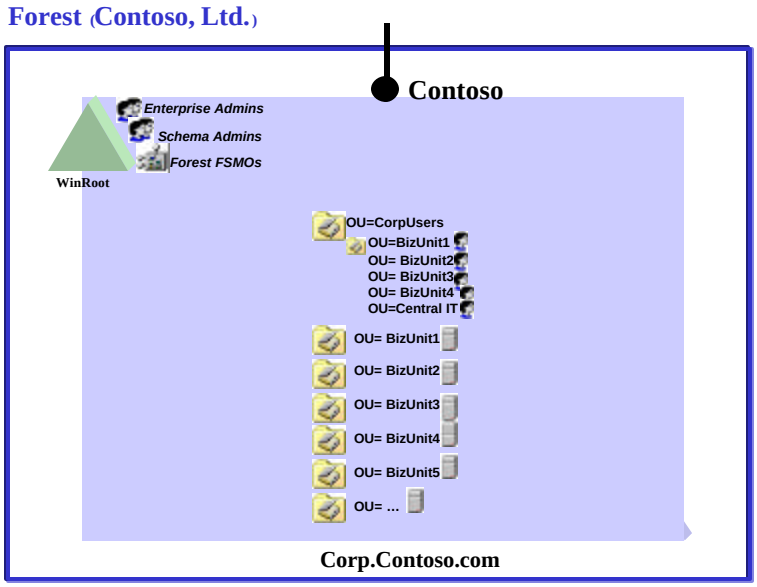



**Figure 21. Organizational Unit Hierarchy**

The creation of organizational units should be done for the delegation of administration and the application of Group Policy objects. It is not necessary to create an organizational unit hierarchy that is aesthetically pleasing or that mirrors the organizational structure of the company. If security doesn't need to be delegated, or if GPOs aren't being applied to a collection of objects, there is no need to create an organizational unit. Even though organizational units are relatively easy to change and restructure, it can become difficult in practice to restructure due to the organizational awareness that occurs with the use of a known naming convention or structure. (For example, a particular application is developed that relies upon a predefined organizational unit hierarchy.)

### Base Design

Geographic organizational units can, for example, contain server resources to allow for delegation to local devices, while users can be less geographically specific, using security groups to filter GPO processing. This approach allows for easy user maintenance, letting the organizational unit structure handle the major dividing properties while reusing security group affiliation to refine the GPOs being applied. It should be noted that computers could be added to security groups as well, allowing for their use of GPO filters.

### General Organizational Unit Design Examples

Contoso, Ltd. has a substantial presence in North America and a substantial International presence. While these geographies share common management schemes, the overall management structure differs significantly. The flexibility of organizational units allows for development of varied superstructures. One size does not have to fit all in organizational unit designs; they should be tailored to suit the way the business and management structure is organized. If the needs change, the structure can change with it. The following is an example of a Contoso North American organizational unit structure. This can be viewed as the general organizational unit superstructure.

The North American organizational structure begins with four top-level containers: Users, Servers, Security, and Workstations. The rationale for creating these containers is based on the current or future need to apply policy to these distinct objects. By allocating top-level containers for these objects, a single point in the tree can be targeted for policy application that will affect all resources of that type.

**Example:** Workstations throughout North America could have security settings, startup scripts, or applications deployed by implementing a single policy at the top of the container tree.

Extensive use of security group filtering or inheritance blocking can complicate policy administration and troubleshooting. By separating these objects into distinct branches, the need to filter policy or block inheritance is reduced. An additional benefit to separating the object types stems from the ability to apply User or Machine policies independently. This separation allows for faster processing by disabling one of the two policy elements in a given policy.

The placement of these four groups into a single North American (U.S.) container to allow administrative delegation and a simplified management interface is recommended and will be implemented once the impact of the change can be reflected in other processes that are now dependent on organizational structure.

#### *Users (Geographic and Functional Divisions)*

Geographic containers will be used as the highest level of security division. A stated requirement of the Contoso, Ltd. North American structure was to limit the scope of administrative privileges for both users and technical support staff. To provide this level of functionality, it required the creation of geographically based containers. By creating separate containers for each field site, support personnel could be delegated control over tasks for that container only.

**Example** A field site technician in San Diego will have the ability to reset passwords for users in the San Diego and San Francisco sites only.

By restricting the scope of privileges, the risk of accidental changes or intentional mischief is mitigated. Technicians who are responsible for covering multiple sites can be given permissions on those specifics sites without compromising global security.

Functional divisions by business unit provide the ability to target a division for software deployment or specific configuration.

**Example:** Sales employees may use a customer management application that could be updated through Group Policy.

**Note** The current implementation of nesting business unit containers inside location-based organizations requires the policy to be applied to multiple containers to affect changes to an entire business unit. An alternative would be to apply the policy to the top of the user's container and filter the policy based on a security group.

**Example:** A single policy object is created to deploy an application to sales personnel; this policy object is then linked to each of the Sales containers in all of the location containers.

Alternatively a single policy object is created to deploy an application to sales personnel; this policy is applied to the user's container, but only the sales security group would have rights to read the policy.

Reversing the nesting of function and geography results in similar solutions when dealing with the targeting of all users of a particular location. The current choice was prompted by the known need to delegate administration based on location compared to a potential need to deploy application based on business organization.

### *Servers*

Application and resource servers in North America are members of containers based on function and then location. By having no users in this branch of the tree, all policies can be implemented with the user portion of policy disabled, providing faster processing. The functional breakdown provides a single point in the tree to deploy applications or apply registry settings specific to that type of server.

**Example:** File and Print servers could have a policy applied that sets the application, security, and system log file size and overwrite settings different than those in Microsoft Exchange or Microsoft SQL Server™.

The additional categorizations by location allow the scope of technicians' rights to be limited to those servers in a particular site or group of sites. Higher-level support engineers can be granted privileges across location boundaries as needed.

### *Security*

Creating a container to hold administrative security groups offers the ability to separate the management of the security groups from the management of delegated organizational unit containers. The ability to make modifications to broad security groups or to modify the attributes of service accounts will likely be limited to very few individuals. The ability to delegate rights on a container dictated the creation of the top-level security container. This container is for the collection of objects that fall outside of other delegated containers.

**Example:** A container established for Customer Service to house servers, test accounts, and other objects specific to Customer Service will have security groups within it that are under the control of the Customer Service organizational unit administrators. However, the Customer Service organizational unit administrators group will reside in the Security container to allow corporate administrators the ability to monitor who has rights over the Customer Service container.

### *Workstations*

The Workstations container has been created for the same reasons as stated previously—the ability to apply computer policies independently to the top of the branch that will affect all computers within. Currently, workstations are categorized into only two groups, laptops and desktops. This structure was designed to allow for policy application different for machines that were known to be well connected to the corporate network versus those that potentially were not connected at all.

**Example:** A large application can be assigned to well-connected desktops through policies, while laptop users will be mailed CDs to perform the installation.

Furthermore, the grouping of workstations is entirely possible by either geographic or functional criteria, if required. Functional containers often provide the ability to use policy-based software deployment for varied configurations as well as to use different levels of security based on the machine's purpose.

**Example:** A manufacturing machine may have a more restrictive lockdown policy applied than a sales representative machine.

At this time, this level of structure has not been implemented and reasons to do so have not yet presented themselves.

## Delegation of Administration

The following diagram depicts the delegation of the organizational unit for BizUnit1 to the ***Biz1 Admin*** group, where that group can:

- Further delegate rights to other administrators.
- Create new objects, such as Computers and Groups in that organizational unit or any sub-organizational unit that is created.

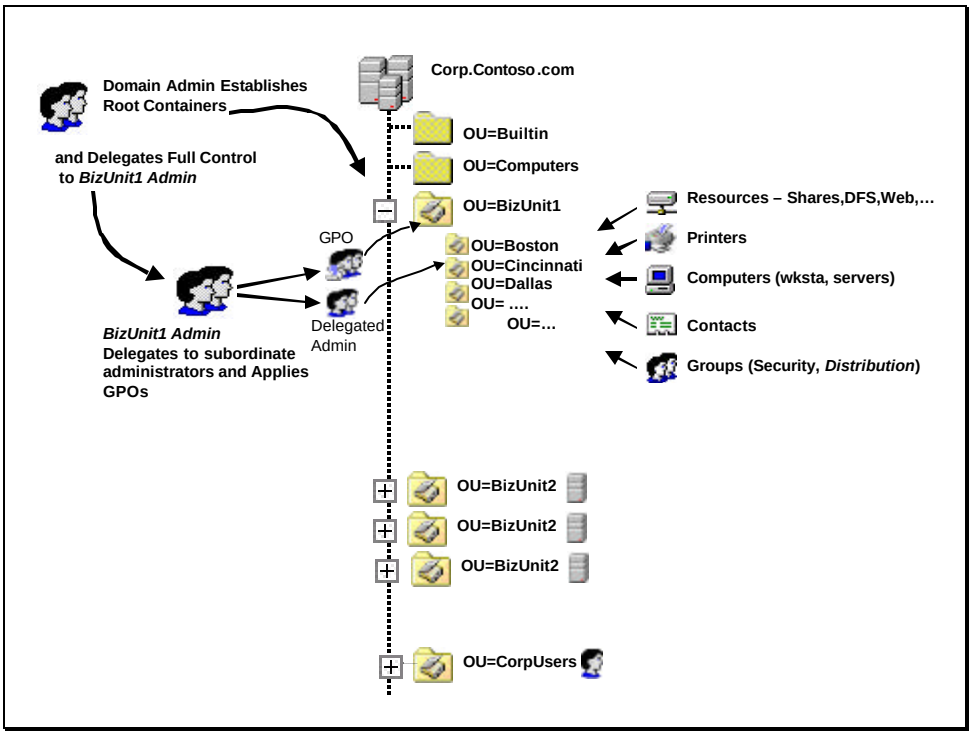



**Figure 22. Corporate Users**

### *GPO Applied to Corp User Container*

At the CorpUsers container, Central Security can control the adding and deleting of all corporate users, while the business units can have rights to control the behavior of those users through the application of Group Policy objects.

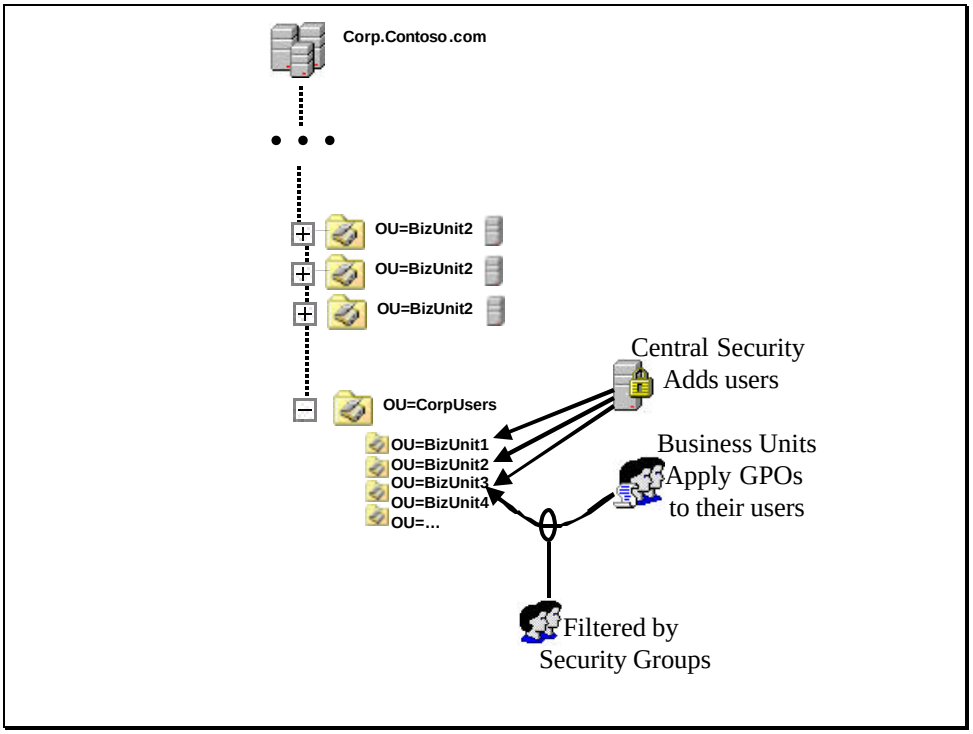



**Figure 23. Adding Corporate Users**

## *Special System Delegation*

### *Delegate Server Rights to a Domain Controller without Active Directory Permissions*

Domain administrators are configured by default to have permissions on the directory and user rights on the domain controller servers. Sometimes, just server level rights are desired for domain controllers. To accomplish this type of security, you must create a new global group called DC Server Admin, and make the following two security changes:

- Put the DC Server Admin group in the <domain>\Administrators group—this gives DC Server Admin the user rights to manage the physical domain controller device.
- At the top of the domain naming context (right-click **<domain>** in the Users and Computers snap-in), select **Deny Full Control for "DC Server Admin"**. Repeat this Deny Full Control setting on the Built-in and Domain Controllers Container. This denies all access to Active Directory.

This example points out the differences between *permissions* in Active Directory and *user rights* on computers.

Permissions in Active Directory come from:

- Default permissions set in the Active Directory Schema for a particular object type.
- Inherited permissions from parent containers.
- Explicitly defined permissions.

User Rights come from:

- Policies defined in Active Directory.
- Local policies defined on the server.

Using various combinations of permissions and rights, the previous example may be extended to allow DC Server Admin to have user rights only on a single domain controller, or to have some permissions on a particular organizational unit in the domain.

### *Delegate the Sites and Services Container*

Default control of the Configuration container is granted to the Enterprise Admins group. The Enterprise Admin group is an authoritative group with many computer rights and permissions on Active Directory. Often there are components of the forest configuration that need to be administered, but that group should not be given Enterprise Admins permissions.

An example is the creation of the network components like sites, site links, and subnets. To delegate this capability, create a new domain global group "Network Administrators." In the Sites and Services snap-in, navigate to the Security on Sites container. In the **Advanced** dialog box for NetServices, add **Network Administrators** with **Full Control** to **This Object and all Child Objects**.

This gives Network Administrators the rights to create sites, site links and subnets. If desired, the Sites, Inter-site Transports, and Subnets containers can be delegated individually, but typically, one group would handle those configurations, and further delegation may add complexity for no reason. The advantage of delegating rights is to allow someone who is not a Domain Administrator to build sites.

### *Delegate the Authorization of DHCP and RIS Servers*

As stated above in the "Delegate the Sites and Services Container" section, the default control of the configuration container is granted to the Enterprise Admin group. Often there are components of the forest configuration that need to be administered, but it is not desired to give that group full Enterprise Admin Permissions.

To delegate DHCP and RIS server authorization, create a new domain global group "Network Services Administrators," and go to **Security** on **NetServices** under the Services container. Click **Advanced** and add **Network Services Administrators** with **Full Control** to **This Object and all Child Objects**. This allows the Network Services Administrators group to add network services without needing to be an enterprise administrator.

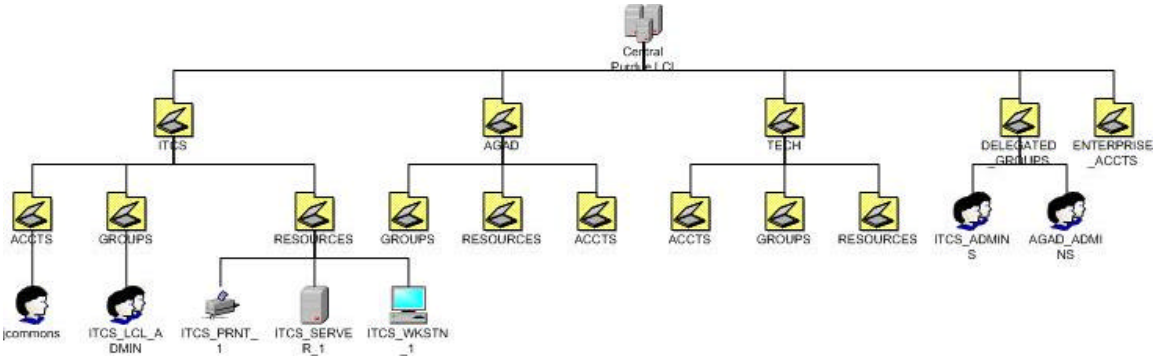
### *Situation and Requirements*

The OU design for the central domain should be organized in such a way that it can be mostly self maintained. This can be accomplished with a comprehensive OU delegation strategy giving each school or department control over their objects so that their designated administrators can maintain all Active Directory objects that they are responsible for.

### *Design Decisions*

The OU design for the CENTRAL Active Directory Domain will be very simple and will enable delegation of objects to their corresponding schools or departments. At the top level of the Domain an OU for each school or department will be created. There will be at least 3 sub OUs under these OUs

which will be used to hold Users, Groups, Computer and Resource objects within the Active Directory. Underneath these OUs the delegated administrator will be able to create additional sub OUs and arrange their Active Directory objects in a way that makes more sense for their environment if necessary.




At the top level there will also be administrative OUs created to hold User and Group objects that are used for Active Directory administration. The first of these OUs is the "Delegated_Groups" OU. The Delegated_Groups OU is used to store all of the Global and Universal groups that are created to give users delegated permissions. This gives the Enterprise and Central Domain administrators the ability to keep all groups that are being given special permissions in the domain in a single location where they can be tracked. Also permissions to modify these groups will only be given to the Enterprise and Central Domain administrators so that they can ensure that only authorized users are getting special permissions in the domain. The other top level OU "Enterprise_Accts" is used to store the Administrative accounts of all Enterprise and Domain Administrators. Special accounts will be created for these people in this OU so that they are not logging on to the network with their administrative account when doing user specific functions like Email Etc. This ensures that the administrative accounts are only being logged on to when their special permissions are required.

# Group Policy Objects

Group Policy defines and controls the behavior of applications and network resources for organizations users and computers. There are hundreds of policy settings that can be managed, which can be combined with inheritance / override restrictions and security group filtering, to generate a nearly unlimited number of possibilities. Group Policy object planning is closely linked to organizational unit planning, since organizational units are built for two basic reasons:

- Delegation of administration
- Application of Group Policy objects

Because of their breadth and complexity, GPOs are frequently under-used. This section describe basic uses that are essentially must-have GPOs. The focus in this document is on server configuration, which leaves out one of the largest areas of management—the management of the user workstation through Group Policy object.

When setting Group Policy objects, remember that GPOs are automatically applied to domain controllers every five minutes and every 90 minutes (plus a random offset) to Windows 2003 member servers and workstations. To force the application of Group Policy objects, the following command can be run from the command prompt:

```
SECEDIT /REFRESHPOLICY MACHINE_POLICY /ENFORCE
```

**Note** Group Policy objects only affect Windows 2003 or Windows XP computers.

## *Base Recommendations*

Every domain should set the following two Group Policy object settings in the Default Domain Policy GPO that appears by default at the domain level.

### *Location Tracking*

```
Found in: Computer Configuration – Administrative Templates – Printers
Policy: Pre-populate printer search location text
Description: Enables the physical Location Tracking support feature of
Windows 2003 printers.
```

Location tracking lets you design a location scheme for your enterprise and assign computers and printers to locations in your scheme. Location tracking overrides the standard method of locating and associating users and printers, which uses the IP address and subnet mask of a computer to estimate its physical location and proximity to other computers.

If you enable location tracking, a **Browse** button appears beside the **Location** field in the **Find Printers** dialog box. (To go to the **Browse** button, click **Start**, click **Search**, and click **For printers**.) The **Browse** button also appears on the **General** tab of the **Properties** dialog box for a printer. It lets users browse for printers by location without having to know the precise location (or location naming scheme). Also, if you enable the Computer location policy, the default location you type appears in the **Location** field.

Location tracking becomes enabled everywhere when this GPO is turned on. The **Location** tab on Subnet and Sites, for instance, now displays a browseable field where a physical tree can be constructed and reused by all objects in Active Directory. Rigorous use of this field in all objects adds a searchable physical dimension to Active Directory objects. The user will be able to find printers by their locations, servers, or subnets. Sites can also be sorted and found by location. It is possible to construct a simple query asking for all components in a physical location.

See "Publishing Printers" later in this document for more details.

### *Account Policy*

`Found in: Computer Configuration – Windows Setting – Security Settings`

`Description: Account Policy (Set in the "Default Domain Policy") defines account authentication configuration settings.`

For domain accounts, Account policy settings must be consistent across the domain. You cannot define individual account policy settings for users within a specific organizational unit. Account policy is set for the entire domain.

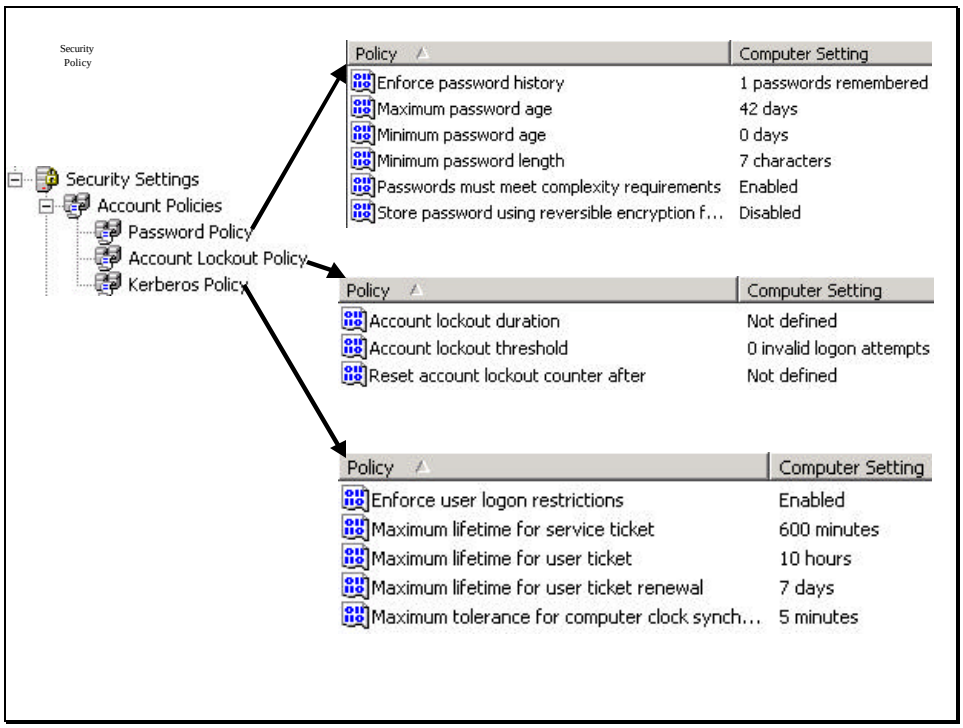



**Figure 24. Account Policy Settings**

**Note** Account policy settings applied at the organizational unit level affect the local Security Accounts Manger (SAM) databases but not the user accounts within the organizational unit. This is because the user accounts apply their Account policy settings from the default domain policy.

The Account policies listed in the previous picture should be set to match the Corporate Security policy. The Kerberos policies are usually accepted as listed above. A password length of seven characters and enabling required complexity are typical minimum settings for well-secured environments.

### Security Templates

A set of security templates is provided for common security scenarios. These can be assigned directly to a computer as is or modified to suit unique security requirements. Predefined security templates should not be applied to production systems without testing to ensure that the right level of application functionality is maintained for your network and system architecture.

The common predefined security templates are:

- Default workstation (**basicwk.inf**). Default security settings. User rights\restricted groups not included. (Windows 2003 Professional)
- Default server (**basicsv.inf**). Default security settings. User rights\restricted groups not included. (Windows 2003 Server)
- Default domain controller (**basicdc.inf**). Default security settings. Requires the environment variables %DSDIT%, %DSLOG% and %SYSVOL% . Must be joined to a domain in order to open. User rights\restricted groups not included. (Windows 2003 domain controllers)
- Compatible workstation or server (**compatws.inf**). Assumes clean install of the NTFS file system and registry discretionary access control list (DACL). Relaxes DACLs for Users. Empties Power Users group.
- Secure workstation or server (**securews.inf**). Assumes clean install of NTFS and DACLs. Secures remaining areas. Empties Power Users group.
- Highly secure workstation or server (**hisecws.inf**). Increases SecureWS settings. Restricts Power User and Terminal Server DACLs.
- Secure domain controller (**securedc.inf**). Assumes clean install of NTFS and registry DACLs. Secures remaining areas.
- Highly secure domain controller (**hisecdc.inf**). Assumes clean install of NTFS registry DACLs. Includes SecureDC settings with Windows 2003-only enhancements. Empties Power Users group.

By default, these templates are stored in the ***\systemroot*\security\templates** folder. Use the Security Templates snap-in to view them. These are common templates; others are listed in the snap-in.

In addition to the predefined templates, a security configuration and analysis snap-in can be used to evaluate the current local security settings against any desired configuration template.

The templates can be applied once as a default setting (secedit command) or loaded into a GPO to continuously maintain a set of specified security settings.

#### *Template For Domain Controllers*

The predefined template securedc.inf should be applied to the Default Domain Controllers Policy GPO on the Domain Controllers container. This policy should be enhanced as necessary to conform to Corporate Security guidelines, but in its default state, this template provides many basic enhancements, including auditing and even log properties.

### Situation and Requirements

Only the minimum requirements and domain only Group Policy Configurations will be configured in the Default Domain Policy. This will allow school and department administrators the ability to create Group Policies that meet their needs.

### Design Decisions

## Default Domain Policy

The following items will be configured in the Default Domain Policy:

**Security Policy**
The ITEP Security Group password requirements will be used as follows:

1. Password Policy
   (Computer Configuration\Windows Settings\Security Settings\Account Policies\Password Policy)

   Enforce Password History: Enabled
   Maximum Password Age: 1 Year
   Minimum Password Age: 1 Days
   Minimum Password Length: 7 Characters
   Password Complexity: Enabled
   Reversible Encryption: Disabled

2. Account Lockout Policy
   (Computer Configuration\Windows Settings\Security Settings\Account Policies\Account Lockout Policy)

   Account Lockout Duration: 15 Minutes
   Account Lockout Threshold: 3 Invalid Attempts
   Reset Account Lockout Counter: Not Defined

**Computer Location Policy: Enabled**
(Computer Configuration\Administrative Templates\Printers)

Specifies the default location criteria used when searching for printers.

This policy is a component of the Location Tracking feature of Windows 2003 printers. To use this policy, enable Location Tracking by enabling the Pre-populate printer search location text policy.

When Location Tracking is enabled, the system uses the specified location as a criterion when users search for printers. The value you type here overrides the actual location of the computer conducting the search.

Type the location of the user's computer. When users search for printers, the system uses the specified location (and other search criteria) to find a printer nearby. You can also use this policy to direct users to a particular printer or group of printers that you want them to use.

If you disable this policy or do not configure it, and the user does not type a location as a search criterion, the system searches for a nearby printer based on the IP address and subnet mask of the user's computer.

**Pre-Populate Printer Search Location Text: Enabled**
(Computer Configuration\Administrative Templates\Printers)

Enables the Location Tracking feature of Windows 2003 printers.

Location tracking lets you design a location scheme for your enterprise and assign computers and printers to locations in your scheme. Location tracking overrides the standard method of locating and associating users and printers, which uses the IP address and subnet mask of a computer to estimate its physical location and proximity to other computers.

If you enable location tracking, a **Browse** button appears beside the **Location** field in the **Find Printers** dialog box. (To go to the **Browse** button, click **Start**, click **Search**, and then click **For printers**.) The **Browse** button also appears on the **General** tab of the **Properties** dialog box for a printer. It lets users browse for printers by location without their having to know the precise location (or location naming scheme). Also, if you enable the Computer location policy, the default location you type appears in the **Location** field.

If you disable this policy or do not configure it, Location Tracking is disabled. Printer proximity is estimated based on IP address and subnet mask.

## Default Domain Controller Policy:

The Default Domain Controller policy applies only to the Domain Controllers in a domain. The following configuration changes will be made to the Default Domain Controller Policy in the Root domain and the Central domain:

**Additional Restrictions for Anonymous Users: Enabled**
**(**Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options)

Determines what additional restrictions should be placed on anonymous connections to the computer.

Windows 2003 allows anonymous users to perform certain activities such as enumerating the names of domain accounts and network shares. This is convenient, for example, when an administrator wants to grant access to users in a trusted domain that does not maintain a reciprocal trust. By default, an anonymous user has the same access that is granted to the Everyone group for a given resource.

This security option allows additional restrictions to be placed on anonymous connections as follows:

- **None. Rely on default permissions.**
- **Do not allow enumeration of SAM accounts and shares.**
  This option replaces "Everyone" with "Authenticated Users" in the security permissions for resources.
- **No access without explicit anonymous permissions.**
  This option removes "Everyone" and "Network" from the anonymous users token; thus requiring that "Anonymous" be given explicit access to any required resources.

This policy is defined by default in Local Computer Policy. By default, no additional restrictions are in place for anonymous connections.

# Authentication

Windows 2003 has adopted the Kerberos V5 security protocol as the default protocol for network authentication. As an emerging standard, Kerberos provides a foundation for interoperability while enhancing the security of enterprise-wide network authentication.

Windows 2003 implements Kerberos version 5 with extensions for public key authentication. The Kerberos client is implemented as a security provider through the Security Support Provider interface. Initial authentication is integrated with the Winlogon single sign-on architecture. The Kerberos Key Distribution Center (KDC) is integrated with other Windows 2003 security services running on the domain controller and uses the domain's Active Directory as its security account database.

## *Authentication Options*

Windows 2003 supports several protocols for verifying the identities of users who claim to have accounts on the system, including protocols for authenticating dial-up connections and for authenticating external users who access the network over the Internet. However, there are only two choices for network authentication within Windows 2003 domains:

- **Kerberos Version 5.** The Kerberos version 5 authentication protocol is the default for network authentication on computers with Windows 2003.
- **NTLM.** The NTLM protocol was the default for network authentication in the Windows NT 4 operating system. It has been retained in Windows 2003 for compatibility with down-level clients and servers. NTLM is also used to authenticate logons to standalone computers with Windows 2003.

All computers running pre-Windows 2003 Microsoft operating systems use the NTLM protocol for network authentication in Windows 2003 domains. Computers running Windows 2003 use NTLM when authenticating to servers with Windows NT 4 and when accessing resources in Windows NT 4 domains. However, the protocol of choice in Windows 2003, when there is a choice, is Kerberos V5.

*Benefits of Kerberos Authentication*

One of the design goals of Windows 2003 is to enable administrators to turn off NTLM authentication once all network clients are capable of Kerberos authentication. The Kerberos protocol is more flexible and efficient than NTLM, and more secure. The benefits gained by using Kerberos authentication are:

- **More efficient authentication to servers**. With NTLM authentication, an application server must connect to a domain controller in order to authenticate each client. With Kerberos authentication, the server does not need to go to a domain controller. It can authenticate the client by examining credentials presented by the client. Clients can obtain credentials for a particular server once and reuse them throughout a network logon session.
- **Mutual authentication**. NTLM allows servers to verify the identities of their clients. It does not allow clients to verify a server's identity, or one server to verify the identity of another. NTLM authentication was designed for a network environment in which servers were assumed to be genuine. The Kerberos protocol makes no such assumption. Parties at both ends of a network connection can know that the party on the other end is who it claims to be.
- **Delegated authentication**. Windows services impersonate clients when accessing resources on their behalf. In many cases, a service can complete its work for the client by accessing resources on the local computer. Both NTLM and Kerberos provide the information that a service needs to impersonate its client locally. However, some distributed applications are designed so that a front-end service must impersonate clients when connecting to back-end services on other computers. The Kerberos protocol has a proxy mechanism that allows a service to impersonate its client when connecting to other services. No equivalent is available with NTLM.
- **Simplified trust management**. One of the benefits of mutual authentication in the Kerberos protocol is that trust between the security authorities for Windows 2003 domains is by default two-way and transitive. Networks with multiple domains no longer require a complex web of explicit, point-to-point trust relationships. Instead, the many domains of a large network can be organized in a tree of transitive, mutual trust. Credentials issued by the security authority for any domain are accepted everywhere in the tree. If the network includes more than one tree, credentials issued by a domain in any tree are accepted throughout the forest.
- **Kerberos policy options**. In Windows 2003, Kerberos policy is defined at the domain level and implemented by the domain's KDC. Kerberos policy is stored in Active Directory as a subset of the attributes of domain security policy. By default, policy options can be set only by members of the domain administrators group.

Kerberos policy options include the following settings:

| Policy | Description | Default |
|---|---|---|
| Maximum user ticket lifetime | A "user ticket" is a Ticket-Granting-Ticket (TGT), an encrypted session ticket with the KDC itself. | 10 hours |
| Maximum lifetime that a user ticket can be renewed. | Total lifetime of ticket with possibly many session renewals. | 7 days |
| Maximum service ticket lifetime | The setting must be greater than 10 minutes and less than the setting for Maximum user ticket lifetime. | 10 hours |
| Maximum tolerance for synchronization of computer clocks | Acceptable time variance. | 5 minutes |
| Enforce user logon restrictions | When this option is enabled, the KDC validates every request for a session ticket by examining user rights policy on the target computer to verify that the user has the right either to log on locally or to access this computer from the network. Verification is optional because the extra step takes time and may slow network access to services. | Enabled |

## Security Groups

Windows 2003 supports four types of security groups: local, domain local, global, and universal.

### *Local Groups*

Local groups, which existed in Windows NT, can contain members from anywhere in the forest, in other trusted forests, or in a trusted pre-Windows 2003 domain. However, local groups can only grant resource permissions on the computer on which they exist.

Special cases for local groups in Windows NT are those created on a primary domain controller (PDC). The replication of the domain security accounts manager (SAM) among the backup domain controllers (BDCs) resulted in these local groups being shared between the PDC and the BDCs. In mixed mode, local groups behave the same in both Windows NT and Windows 2003. In native mode, local groups on a domain controller become domain local groups, which are described in the next section. Typically, local groups are used to grant specific access to resources on a local computer.

### *Domain Local Groups*

Domain local groups are a new feature of Windows 2003, though similar in concept and use to the local groups created on the PDC in a Windows NT domain.

Domain local groups are only available in native mode domains and can contain members from anywhere in the forest, in trusted forests, or in a trusted pre-Windows 2003 domain. Domain local groups can only grant permissions to resources within the domain in which they exist. Typically, domain local groups are used to gather security principals from across the forest to control access to resources within the domain.

### *Global Groups*

Windows 2003 global groups are effectively the same as Windows NT global groups. Windows 2003 global groups can only contain members from within the domain in which they exist. These groups can be granted permissions to resources in any domain in the forest or in trusted forests.

### *Universal Groups*

Universal groups can contain members from any Windows 2003 domain in the forest, and can be granted permissions in any domain in the forest or in trusted forests. Though universal groups can have members from mixed mode domains in the same forest, members from such domains do not have the universal group added to their access tokens because universal groups are not available in mixed mode. Though you can add users to a universal group, it is recommended that you restrict membership to global groups.

**Note** Universal groups are only available in native mode domains.

Use universal groups to build groups that perform a common function within an enterprise. An example of this is virtual teams. The membership of such teams in a large company could be nationwide, or worldwide, and almost certainly forest-wide, with team resources being similarly distributed. In these circumstances, universal groups could be used as a container to hold global groups from each subsidiary or department, with the team resources being protected by a single access control entity (ACE) for the universal group.

Universal groups and their members are listed in the global catalog. Though global and domain local groups are also listed in the global catalog, their members are not. This has implications for global catalog replication traffic. It is recommended that you use universal groups with care.

| Group Type | Membership From | Scope | Available in Mixed Mode? |
|---|---|---|---|
| Local | The same forest<br>Other trusted forests<br>Trusted pre-Windows 2003 domains | Computer-wide | Yes |
| Domain Local | The same forest<br>Other trusted forests<br>Trusted pre-Windows 2003 domains | The local domain | No |
| Global | Local domain | Any trusted domain | Yes |
| Universal | The same forest | Any trusted native mode domain | No |

### *Group Behavior*

It is recommended that you limit group size to 5,000 members, because the Active Directory store must be able to be updated in a single transaction. Because group memberships are stored in a single multi-value attribute, a change to the membership requires the whole membership list to be replicated between domain controllers and updated within a single transaction. Microsoft has tested and supports group memberships up to 5,000 members.

#### *Nesting Groups*

Nesting groups is an effective way to increase the number of members. Further, nesting groups helps reduce traffic caused by replication of group membership changes. The nesting options depend on whether the domain is in native mode or mixed mode. The following list describes what can be contained in a group that exists in a native-mode domain (these rules are determined by the scope of the group):

- Universal groups can contain user accounts, computer accounts, universal groups, and global groups from any domain in the forest.
- Global groups can contain user accounts and computer accounts from the same domain, and global groups from the same domain.
- Domain local groups can contain user accounts, computer accounts, universal groups, and global groups from any domain. They can also contain other domain local groups from within the same domain.
- Security groups in a mixed-mode domain can contain only the following:
  - Local groups that can contain global groups and user accounts from trusted domains.
  - Global groups that can contain only user accounts.

*Group Membership Expansion*

When a user logs on to a client or makes a network connection to a server, the group membership of the user is expanded as part of building the user access token. Group expansion occurs as follows:

- During interactive logon to a client, a workstation contacts the domain controller to verify user credentials and obtain a Kerberos TGT. The domain controller expands the list of all group memberships for the user for the following group types that are included in the TGT as authorization data:
  - Universal groups defined anywhere in the forest
  - Global groups
  - Domain local groups for the same domain as the user account
- When the client initiates a network connection to a server, if the server is located in a different domain than the user account, a cross-domain referral is used to get a service ticket from the KDC of the server. When the service ticket is issued, group expansion adds the domain local groups of which the user is a member to the domain of the server. These groups are added to authorization data in the service ticket along with the group list in the TGT. If the server is in the same domain as the user account, the domain local groups are already available in the TGT from the initial interactive logon.
- When the client connects to the server, expansion of the local groups occurs if the user account, or one of the groups of which the user is a member, is also a member of any local groups on the server.

When the user access token is being created, all the group membership information expanded by the domain controller or the resource server is used to identify the user.

*Upgrade Effects on Groups*

Upgrading a PDC to Windows 2003 has no immediate effect on groups: Windows NT local groups become Windows 2003 local groups, and Windows NT global groups become Windows 2003 global groups. The real change occurs when you switch the domain to native mode, at which point local groups on the PDC become domain local groups.

## Situation and Requirements

The group design for the University must follow Microsoft best practice for group management, but also be flexible enough to provide the appropriate delegation model.

## Design Decisions

**Domain Local Groups**

Domain Local Groups will be used to assign permissions to any objects within Active Directory. Because of the new capabilities of Domain Local Groups in Active Directory, these groups may also be used to give permissions on member servers or applications running on servers within the domain thereby simplifying the overall group permissions structure within the domain. This would remove the requirement of managing local groups on the application servers.

**Global Groups**

Global Groups will be used to group together user accounts for the purpose of applying permissions via Domain Local Groups. These groups will also be able to function as distribution lists once Exchange 2003 or 2003 is implemented. This will reduce the redundancy between security groups and distribution lists thereby reducing the overall total number of Global Groups

**Universal Groups**

Because Universal Group membership is also kept in the Global Catalog, the use of these groups will be kept at a minimum. A Universal Groups primary function is to contain user or computer objects from any domain within the forest, and be able to be applied to any resource within the forest. Because this domain design will be simple and the number of domains will be kept to a minimum, there will not be much need for them for most groups needs.

If a Universal Group is used, group nesting should be used as much as possible so as to keep the actual number of group member entries to a minimum. This will have a much smaller impact on the size that it takes in the global catalog.

Example:

I have an application running on servers in different domains. I need to be able to give a list of users that span multiple domains the same permissions on every one of these application servers regardless of the domain that the application server resides in. Without using Universal Groups, I would create a Domain Local Group in every domain that the application servers reside in and I would make the Global Groups of each domain that need access to this application a member of each Domain Local Group in each domain that I am using this application. This could be a large number of groups that I need to manage depending on the number of domains. If a new Global Group needs to be added to this application, then I would need to go to every Domain Local Group once again and modify the membership.

If I used a Universal Group, then I have a single group that I create and add the Global Groups of each domain that need access to the application. This Universal Group an then be given permissions on each of the application servers. If I need to add a new Global Group or remove one then all I need to modify the Universal Group, which would immediately change the permissions on every application server.

# Sites and Replication

Windows 2003 domain and forest-wide replication consists of two major components:

- Active Directory replication.
- SYSVOL replication, which utilizes the File Replication service (FRS).

The replication is governed by the configuration of sites and site links.

## *Active Directory and FRS Replication*

Active Directory replication and FRS replication used for SYSVOL are different processes that use the same replication topology, but run independently of each other. There are two major differences between how an available replication window is utilized by Active Directory and by FRS SYSVOL replication: start time and replication behavior. Active Directory replication chooses a start time randomly within the first 15 minutes of a replication window to distribute the concurrence factor across the window. FRS SYSVOL replication, on the other hand, starts the moment the window opens. This means that while Active Directory replication with multiple partners starts at different times within a 15-minute window, FRS SYSVOL replication with multiple partners starts at the same time for all partners.

| | Active Directory | SYSVOL – FRS |
|---|---|---|
| Scope | Forest | Domain |
| Type | Pull (notify/pull within site) | Notify/Push/Ack |
| Concurrent partners – Inbound | Serialized | Parallel |
| Concurrent partners – outbound | Parallel | Parallel |
| Threading model | Single thread per replication partner | Multiple threads per replication partner |
| Versioning | Per attribute version number | Per file timestamp |

### *Active Directory Replication*

Active Directory replication is always a one-way pull replication; the domain controller that needs updates (target domain controller) contacts a replication partner (source domain controller). The source domain controller then selects the updates that the target domain controller needs, and copies them to the target domain controller. Since Active Directory uses a multi-master replication model, every domain controller works as both source and target for its replication partners. From the perspective of a domain controller, it has both inbound and outbound replication traffic, depending on whether it is the source or the destination of a replication sequence.

The replication process in Active Directory encompasses all three naming contexts found in the Forest:

- Schema
- Configuration
- Domain (including global catalog)

*SYSVOL Replication*

FRS is used to replicate system policies and logon scripts stored in SYSVOL. Each domain controller keeps a copy of SYSVOL for network clients to access. FRS can copy and maintain shared files and folders on multiple servers simultaneously. When changes occur, content is synchronized immediately within sites, and by schedule between sites.

FRS is a multithreaded, multi-master replication engine that replaces the LMRepl service. Multithreaded means that several replication sessions can run at the same time to handle multiple tasks. This allows FRS to replicate different files between different computers simultaneously. Multi-master replication means that changes to the SYSVOL can be made on any domain controller, and this domain controller will then replicate the changes out to the other domain controllers using a store-and-forward mechanism. FRS SYSVOL replication uses the same Active Directory replication topology defined by connection objects. In contrast to Active Directory replication, FRS SYSVOL replication uses a timestamp on a file to determine which version is the newer version and should be kept on a domain controller and replicated out to partners.

The SYSVOL includes:

- SYSVOL Share
- NETLOGON Share
- Windows 95, Windows 98, and Windows NT system policies
- Windows 2003 Group Policy settings
- User logon and logoff scripts

### Sites

A site is any given maskable IP subnet range with fast, reliable connectivity. Sites are used to control the direction, schedule, and frequency of replication for Active Directory, the global catalog, and SYSVOL. In addition to replication traffic, sites are used to find resources that are closest to the requestor. For example, when a logon request is looking for a domain controller or when a Distributed File System (Dfs) replica is being chosen, the site affinity will be checked to find a resource that exists in or near a requestor's site.

Site replication (both inter and intra site replication) is generated by the Knowledge Consistency Checker (KCC). For inter-site replication, the KCC selects a bridgehead server in each site and uses that bridgehead to move data between the sites. Each site elects a Topology Generator among the servers in the site to create the replication topologies and acts like an operations master role, but the role flows automatically as necessary and no data needs to be transferred.

### Intra-site Replication

For intra-site replication, the KCC on each domain controller helps to automatically generate and optimize a replication topology among domain controllers in the same domain. To accomplish this, the KCC automatically creates connection objects between domain controllers. A *connection object* is an Active Directory object that represents a communication channel used to replicate information from one domain controller to another. Under normal conditions, Active Directory automatically creates and deletes connection objects. However, you can manually create connection objects to force replication if you are certain the connection is required and you want the connection to persist until you manually remove it.

Intra-site replication traffic is not compressed and is based on a notification process that replicates changes as fast as possible. A brief dampening interval is used to batch the changes efficiently so network sessions are not continuously built and destroyed while data updates are still active on a particular domain controller.

### Inter-site Replication

For inter-site replication to occur, you must customize how Active Directory replicates information by setting up site links. Site links are logical transitive connections between two or more sites that mirror the network links and allow replication to occur. Once site links are created, the KCC will automatically generate the replication topology by creating the appropriate connection objects. Site links are used by the KCC to determine replication paths between sites and must be created manually. Connection objects are what actually connect domain controllers together and are created automatically by the KCC. Connection objects may also be created manually when discrete control is required.

Inter-site replication uses data compression and scheduling to manage the impact on wide area networks. Compression is used when the payload exceeds 50 kilobytes (KB) and typically achieves a 10X compression ratio (1,000 KB of data would compress to 100 KB of data). The amount of compression achieved is impacted by the type of object and data within the object.

For every site, one domain controller, known as the Inter-Site Topology Generator (ISTG), is responsible for managing the inbound replication connection objects for all bridgehead servers in the site in which it is located. If the server holding the ISTG role is taken offline, the role is automatically transferred to another domain controller in that site by the system, unlike the transfer of the operations master roles.

### Site Links

A site link represents network connectivity between two or more sites. A site link allows the administrator to assign cost, a replication schedule, and a transport for replication. Cost is an arbitrary value selected by the administrator to reflect the relative speed and reliability of the physical connection between the sites; the lower the cost, the more desirable is the connection.

To get a feasible cost factor (not including the operating cost of the link) for site links that correlates to the available bandwidth you could use the following formula:

$$Cost = \frac{1024}{\log(\ AvailableB\ andwidth\ [Kb\ ])}$$

The following table shows some examples of the formula applied to various line speeds.

| Available bandwidth (kilobits/second) | Cost |
|---|---|
| 9.6 | 1042 |
| 19.2 | 798 |
| 38.8 | 644 |
| 56 | 586 |
| 64 | 567 |
| 128 | 486 |
| 256 | 425 |
| 512 | 378 |
| 1024 | 340 |
| 2048 | 309 |
| 4096 | 283 |

### *Link Costs*

Site links are used to control the flow of inter-site replication traffic. The configuration of site links typically follows the detailed network topology, and the desired paths for replication. The cost comes into play when there is more than one route to take in a replication scheme. This is important in the multi-path scenario and in the single-path scenario when a server is unavailable and the replication is bridged through the downed node.

### *Replication Schedules*

Replication schedules can be set to any frequency, from immediate notification to minutes to hours to daily depending on the data consistency required and the need to manage peak bandwidth usage.

### *Site Link Bridges*

A site link bridge is a collection of two or more site links and provides a structure to build transitive links between sites and evaluate the "least cost path." Site link bridges are only significant when the **Bridge all site links** option is not enabled. This may be necessary in scenarios where not all naming contexts are in fully routed sites. Bridging all site links implies that all site links are transitive. When this mode is on, bridges are ignored, and all site links are considered to be in one big bridge. This is the default behavior in Windows 2003.

### Creating a Site Plan

During an architecture planning process, one of the most important outcomes of working on a site plan is determining if the network can be configured to support the Forest/Domain plan. Working out the details of every site, site link and sub-net is not necessary to determine if the logical design is supportable. For example, if we had a three-main-office design with several thousand users at each location, and they were connected to each other through T1 lines with moderate available bandwidth and the logical design called for a single domain, we could safely say that the network could handle the logical design. It would then be a straightforward process of picking site names, links, costs, and frequencies.

Site topologies are flexible. Once configured, they can be adjusted and readjusted as necessary. At this point in the architecture process, it is imperative to determine only a few things:

- How many locations will have domain controllers? (This information necessitates the creation of a site.)
- What is the best, worst, and typical link to the sites determined above? (Determine the speed of the link and how much bandwidth is available.)
- What is the "deepest" piece of the topology? (For example, a single hub and spoke is only one level deep, and a tail circuit off of one of the spokes would create a two-level-deep site.) The "depth" of a site helps determine if there are any substantial latency issues.
- What is the highest number of links coming into one site? (This information helps identify bridgehead sizing issues.)

Special considerations for branch office deployments, sites, and replications are discussed in detail in the document titled Active Directory Branch Office Deployment Guide, located at: http://www.microsoft.com/WINDOWS2003/techinfo/planning/activedirectory/branchoffice/default.asp.

### Situation and Requirements

Design a comprehensive Active Directory Site and Replication topology which works well based on the existing Purdue physical network infrastructure.

### Design Decisions

#### *Sites*

Based on an analysis of the existing network infrastructure at Purdue University the Site Topology will consist initially of 5 sites. These site boundaries will act as the boundary for all authentication and DNS lookups in that site. All IP subnets on the campus network will be associated with the Active Directory site that it belongs to. The site configuration is listed as follows:

1. Main Campus

   The Main Campus site will encompass the entire West Lafayette campus as well as any location that connects directly to the West Lafayette campus and has very few concurrent users.

2. Fort Wayne

   The Fort Wayne site will encompass the entire Fort Wayne campus.

3. Calumet

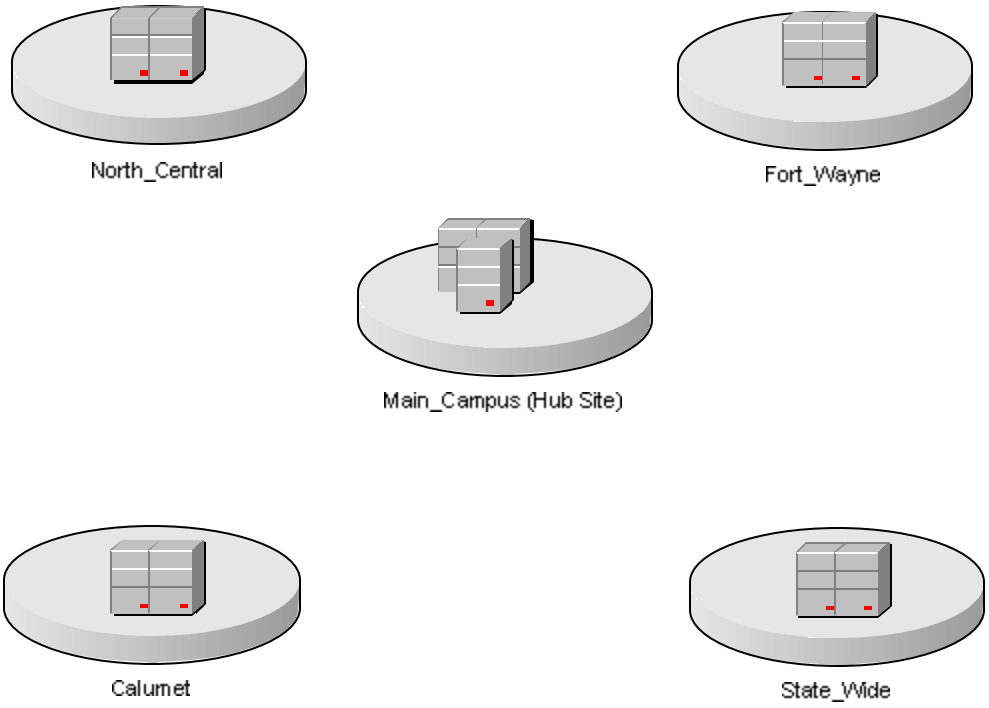
   The Calumet site will encompass the entire Calumet campus.

4. North Central

   The North Central site will encompass the entire North Central campus.

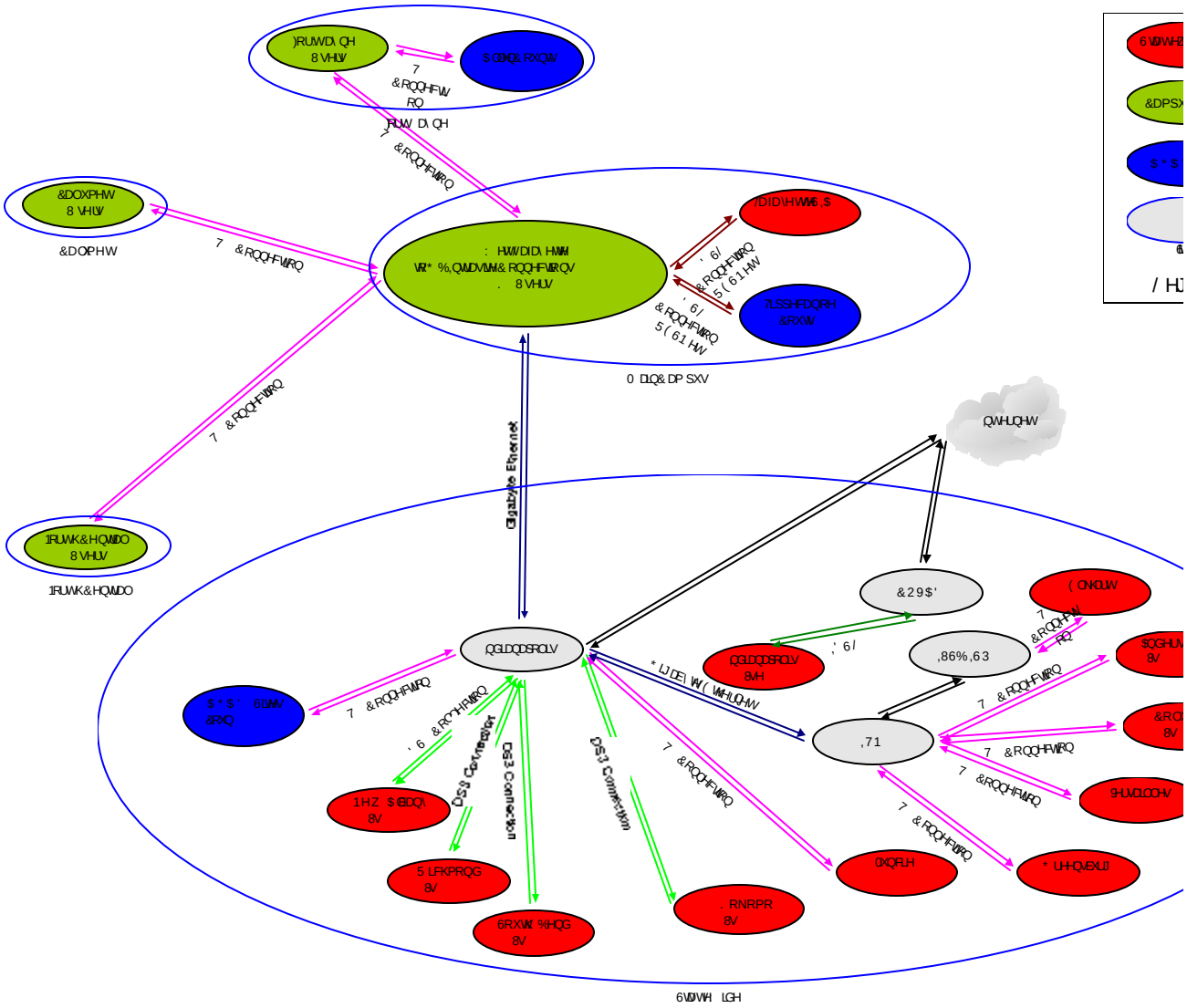
5. State Wide

The State Wide site consists of all outside locations that connect to the IHETS (<Acronym???>) or directly to Indianapolis campus which is routed to the Main Campus over a Gigabit Ethernet Connection.




University Site Diagram

This conceptual site diagram was based on the physical structure of the Purdue University network which is shown in the following diagram:



Purdue University Physical Network Infrastructure \ Site Diagram

### *Site Links*

In order to accomplish the desired hub and spoke replication model, Site Links will be created for each replication connection so that replication can be controlled by a cost factor and also so that each site replication schedule can be controlled differently.

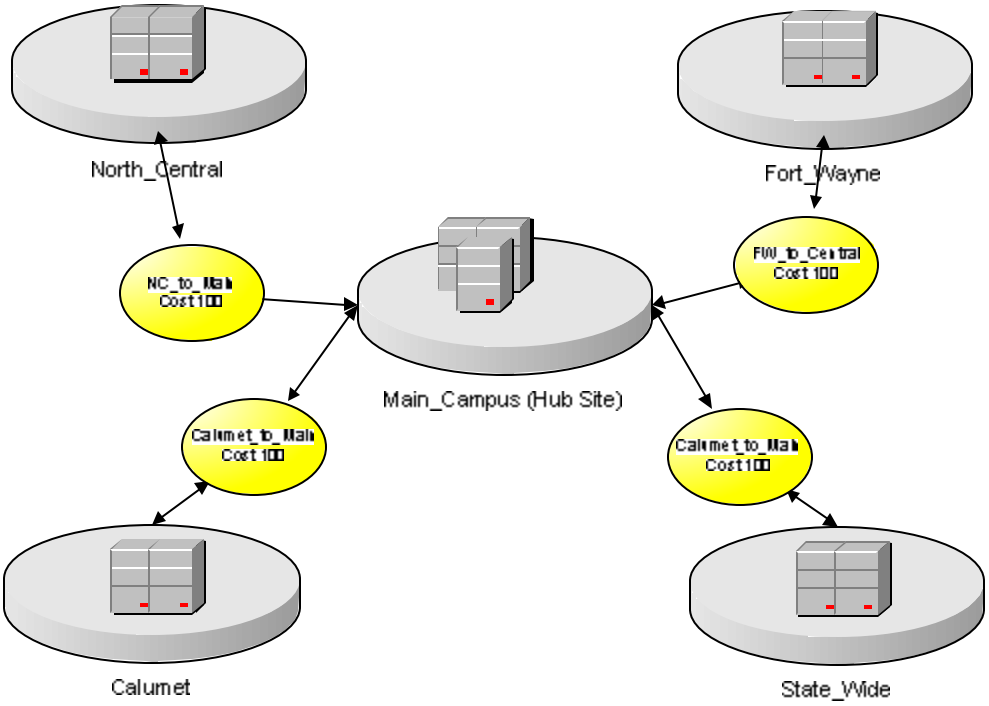
All sites will use the IP Inter-Site Transport to replicate active directory. The default "Bridge All Site Links" setting will remain enabled so that each of the sites may replicate with another site in the unlikely event that the domain controllers in the Main_Campus site are unavailable.

The site names and site membership of the Site Links will be as follows:

1. NC_TO_MAIN (Cost 100)
    - North_Central

- Main_Campus

2. FW_TO_MAIN (Cost 100)
    - Fort_Wayne
    - Main_Campus
3. Calumet_TO_MAIN (Cost 100)
    - Calumet
    - Main_Campus
4. SW_TO_MAIN (Cost 100)
    - State_Wide
    - Main_Campus




University Site and Site Link Diagram

### *IP Subnets*

Site membership is determined in Active Directory by IP Subnet. Because of this, each subnet on the Purdue University network must be entered into Active Directory Sites and Services and assigned to a specific site.

### *Replication*

The sites and site links must be configured with replication intervals and schedules so that the domain controllers know when they are allowed to make replication requests over the WAN link. Communication is expected to be similar between the Main_Campus site and the other four outlying sites. Because of this each site and site link will have the same replication schedule configuration unless after testing and implementation one of the WAN links to one of the outlying sites does not perform as expected.

- Each Site Link will be configured to be "always available"
- Each Site Link will be configured to replicate ever hour.
- The site replication interval will be left at the default settings (every 15 minutes)

# Services Locations

## *Domain Controller Placement*

For fault tolerance and best performance, at least one domain controller should reside at each site, providing users with a local computer that can service query requests without requiring slow-link traffic. Domain controllers at smaller sites can be configured to receive directory replication updates only during off-hours, and to help optimize network traffic utilization flow.

- A domain controller must be able to respond to client requests in a timely manner.
- The best query performance happens when you place a domain controller (at a small site) with a global catalog server, enabling that server to fulfill queries about objects in all domains on your network.

At a minimum, there must be at least one global catalog in the forest so that users can be authenticated.

**Note** The first domain controller you install in a forest is automatically configured as a global catalog.

### *Global Catalog Placement*

In a multi-site environment, place at least one global catalog at each site:

- This provides a backup resource that will reduce the possibility of disruption in service.
- A global catalog must be available in order to log on. If a global catalog is not available, the users (with the exception of domain administrators) will not be able to log on.

**Note** A registry setting can allow logon without being able to contact a global catalog server.

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\IgnoreGCFailures
```

For more information, see the Microsoft Knowledge Base article, Q241789: "How to Disable Requirement that a Global Catalog Server Be Available to Validate User Logons" at: http://support.microsoft.com/support/kb/articles/Q241/7/89.ASP.

- If user principal names (UPNs) are used for authentication, it may be necessary to place more global catalogs, which will spread the load of UPN lookups. When using UPNs for authentication, the domain controller the users connect to will contact a global catalog to find out which domain the users are in, then it will contact a domain controller in that domain to authenticate the users.

### *DNS Placement*

Using Active Directory DNS provides many benefits and allows the creation of multi-master replicas using Active Directory replication topology. This also allows the addition of a DNS server to each site for very little cost to the environment and virtually no administrative overhead. Benefits of per-site DNS servers include:

- Local name resolution and registration.
- Fault tolerance for over-the-WAN name look-up. This is particularly important when the site is sub-netted.
- Establishing the foundation for elimination of NetBIOS for finding local resources via broadcast.
- Eliminating possible zone transfer delays to a local-site Secondary Server.

*Forest Root Services*

The design principles behind the creation of a forest root domain are in the "Domain Structure" section earlier in this document. The servers that maintain the root have a number of special operations running on them that require particular attention. While the forest root servers have very little authentication and replication traffic, they provide base services to the forest. The forest root servers should be kept on highly fault-tolerant machines and be well maintained, which includes backup, recovery, and FSMO Flexible Single Master Operation management procedures. The following diagram depicts the services that the root computers offer to the environment.

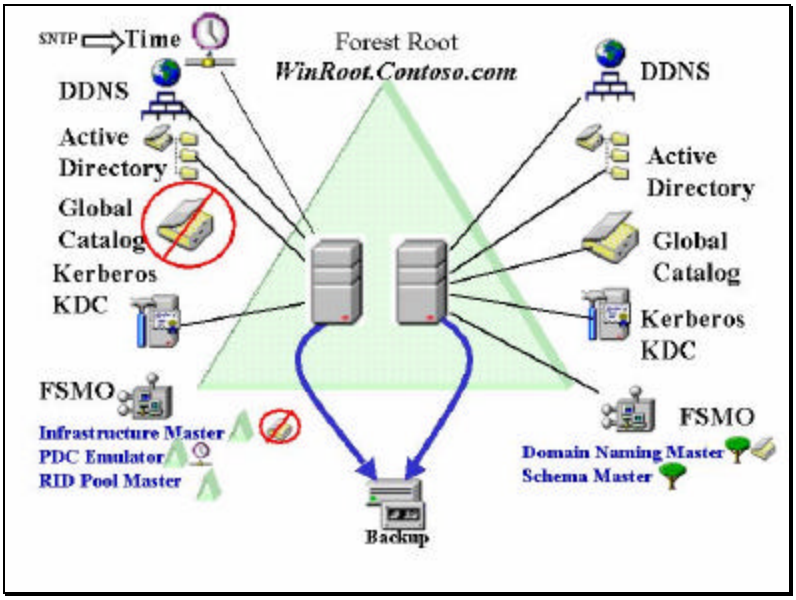



**Figure 25. Forest Root Services**

The operations that are unique to the forest root servers include:

- **Time Service**. The root domain controller maintaining the PDC Emulator role is designated as authoritative for the entire forest.
- **Forest FSMO**. The two forest level FSMOs, schema master and domain naming master, are maintained on the two root servers.
- **Root Domain FSMO**. The root domain like all domains has it's set of three FSMO roles as well, infrastructure master, PDC emulator, and RID pool master.
- **Backup**. Because of the importance of the root domain to the general functioning of the entire forest, the ability to recover this domain is of great importance. Both servers in the forest will be backed up.

The alternative to the above design would allow all roles to be put on one domain controller if all domain controllers were global catalog servers. This eliminates the need for the infrastructure master to perform foreign reference updates, since each domain controller would have direct reference to the object in the global catalog. For serviceability, a third server is commonly added to this configuration, ready to assume the roles of either one of the servers.

*Account Domain Services*

The Account Domain(s) is the center of activity in the domain, hosting nearly the entire authentication, replication, and publishing processes. A pair of servers should be designated as the infrastructure servers for the domain, providing domain level FSMO maintenance, backup, recovery, as well as authentication and query services for the headquarter(s).

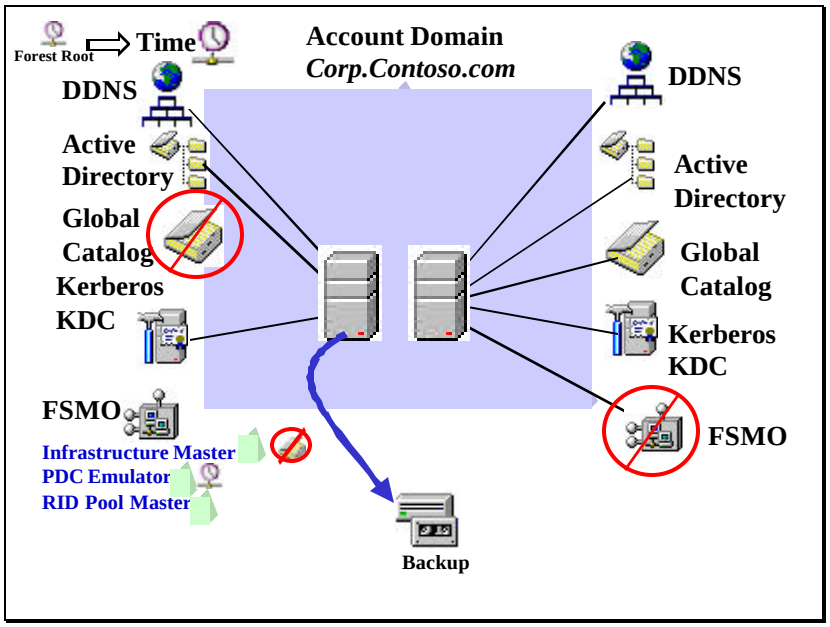



**Figure 26. Account Domain Services**

*Account Domain Replicas*

Account Domain Replicas are distributed to different physical sites to provide local authentication services. The reduced number of services and no requirement for backup allows for the placement of "low overhead" replicas in the enterprise.

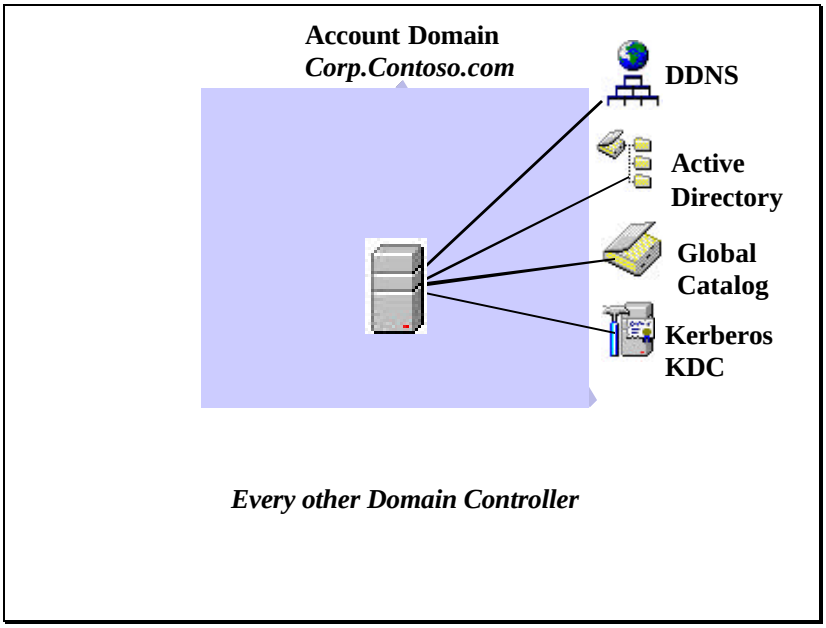



**Figure 27. Account Domain Replicas**

### *Situation and Requirements*

The domain services pictured above need to "just work" consistently and efficiently if ITCS is providing an infrastructure service that provides the foundation to all Windows 2003 interactions. The environment needs to be easy to manage and repeatable, so that domain controllers can easily be added or subtracted from the overall structure without having to do major adaptive work.

### *Design Decisions*

## FSMO Roles

**Forest FSMO Role locations:**

There are two Active Directory Forest specific FSMO roles (Schema Master, Domain Naming Master) which serve a function for the entire forest. Since there will be a dedicated Forest Root domain (Purdue.lcl), the domain controllers in the Forest Root domain will serve these FSMO roles. Both roles will reside on one of the Root Domain Controllers, and it will not host any of the domain FSMO roles for the Root Domain.

**Purdue.LCL (Root Domain) Domain FSMO Role locations:**

The three domain FSMO roles (Relative ID Master, PDC Emulator, Infrastructure master) for the Forest Root domain will be hosted by the Root Domain Controller that is not serving the Forest FSMO roles.

**Central.Purdue.LCL (Child Domain) Domain FSMO Role locations:**

The three domain FSMO roles (Relative ID Master, PDC Emulator, Infrastructure master) for the Central domain will be hosted by a single domain controller for the Central Domain which will be located in the Main_Campus Site.

FSMO roles are very flexible and can be moved at any time if the server that is hosting them needs to go offline or if it becomes unavailable.

### Domain Controller Placement

Based on an analysis of the number of users per site, the following is an explanation of the number of domain controllers required to support Active Directory and where they will be located.

Generally, the size of a domain controller (amount of processing ability) is driven by the number of possible user authentication requests as well as replication load. Domain controllers will be placed in the following sites as follows:

1. Main_Campus (40 – 45,000 Users)
   - Root Global Catalog1
   - Root Global Catalog2
   - Central Global Catalog1
   - Central Global Catalog2
   - Central Global Catalog3
   - Central Global Catalog4
2. Fort_Wayne (5,000 + Users)
   - Central Global Catalog5
   - Central Global Catalog6
3. Calumet (9,000 + Users)
   - Central Global Catalog7
   - Central Global Catalog8
4. North_Central (3,000 + Users)
   - Central Global Catalog9
   - Central Global Catalog10
5. State_Wide (3,000 + Users)
   - Central Global Catalog11
   - Central Global Catalog12

Note: The sizing of these domain controllers is outlined in the Database and Server Sizing of this document.

# Flexible Single Master Operation

Active Directory defines five operations master roles: schema master, domain naming master, relative identifier (RID) master, primary domain controller (PDC) emulator, and infrastructure master. The schema master and domain naming master are per-forest roles, meaning that there is only *one* schema master and *one* domain naming master in the entire forest.

## *Master Roles*

The five operations' master roles and their purpose are listed below:

- **Schema master (forest level FSMO)**. The domain controller that holds the schema master role is the only domain controller that can perform write operations to the directory schema. Those schema updates are replicated from the schema master to all other domain controllers in the forest. Only members of the schema administrator group are allowed to issue a schema modification.
- **Domain naming master (forest level FSMO)**. The domain controller that holds the domain naming master role is the only domain controller that can add new domains to the forest and remove existing domains from the forest.
- **Relative identifier (RID) pool master (per domain FSMO)**. A new security principal object (User, Group, or Computer) can be created on any domain controller. However, after creating several hundred security principal objects, a domain controller must communicate with the domain controller holding the domain's RID master role before creating the next security principal object. Then, another several hundred security principal objects can be created, and when this set of objects has been created, the process of contacting the RID master repeats. If a domain controller's RID pool is empty, and the RID master is unavailable, you cannot create new security principal objects on that domain controller.
- **PDC emulator (per domain FSMO)**. The domain controller holding the PDC emulator provides backward compatibility to down-level backup domain controllers (when running in Mixed Mode). The PDC emulator also serves other roles, including time synchronization and password latency control. Changes to security account passwords present a replication latency problem wherein a user's password is changed on domain controller A, (perhaps by an administrator at a hub site) and the user subsequently attempts to log on, being authenticated by domain controller B (in the local branch office). If the password has not replicated from A to B, the attempt to log on fails. Active Directory replication remedies this situation by forwarding password changes immediately to a single domain controller in the domain, specifically to the PDC emulator. If authentication fails at a domain controller, the authentication request is passed immediately to the PDC emulator domain controller, which is guaranteed to have the current password. The urgent replication of password changes to the PDC emulator occurs immediately without respect to schedules between sites on site links.
- **Infrastructure master (per domain FSMO)**. The domain controller holding the infrastructure master role for the group's domain is responsible for updating the cross-domain group-to-user reference to reflect the user's new name. The infrastructure master updates these references locally and uses replication to bring all other replicas of the domain up to date. If the infrastructure master is unavailable, these updates are delayed.

## Locating Roles

### *Infrastructure Master*

Infrastructure Master should not be a global catalog server.

When an object on one domain controller references an object that is not on that domain controller, it represents that reference as a record containing the GUID, the security identifier (SID) for references to security principals, and the distinguished name of the object being referenced. If the referenced object moves, its GUID does not change, its SID changes if the move is cross-domain, and its distinguished name always changes.

The infrastructure master for a domain periodically examines the references, within its replica of the directory data, to objects not held on that domain controller. It queries a global catalog server for current information about the distinguished name and SID of each referenced object. If this information has changed, the infrastructure master makes the change in its local replica and also replicates the new values to other domain controllers within the domain.

If the infrastructure master runs on a global catalog server, it will never update anything because it does not contain references to objects that it does not hold. This is because a global catalog server holds a partial replica of every object in the forest.

### *Domain Naming Master*

The domain naming master should also be a global catalog server.

When the domain naming master creates an object representing a new domain, it must make sure that no other domain has the same name. The domain naming master achieves this by running on a global catalog server, which contains a partial replica of every object in the forest.

### *Special Considerations in Mixed Mode*

In mixed-mode domains that contain backup domain controllers, the "Standby operations master domain controller" should be in the same site as the primary domain controller emulator. By keeping both domain controllers in the same site, the system can avoid performing a full synchronization with the backup domain controllers in case you seize the PDC emulator role to the standby operations master domain controller.

## Role Integrity and Recovery

### *Role Transfer Is Not Completed*

When a role transfer takes place, it updates the current role owner before it updates the desired new role owner. If the desired new role owner fails before making its update, it does not yet hold the role. The desired new role owner can gain ownership of the role in the following ways:

- Repeat the role transfer attempt.
- Allow replication to update the desired new role owner with the change made at the current role owner. (This typically does not require any action; however, it does take more time than repeating the role transfer attempt.)

*Backing Up Role Masters*

When backing up a domain controller, back up the roles it owns, so when a domain controller is restored from backup media, it restores the role it owns.

*Roles During the Active Directory Installation Wizard Demotion Process*

When removing Active Directory from the domain controller that owns the operations master roles, the domain controller attempts to abandon its roles. For each role the domain controller holds, it locates another available domain controller for the role and transfers the role to it. If another domain controller is not available during the demotion, the demotion process will not succeed.

Do not rely on the transfer feature when removing Active Directory from a domain controller. Instead, transfer any roles before you begin the removal process so that role placements are as they should be.

### Situation and Requirements

FSMO roles don't require active management; they are part of the infrastructure that needs to be present for a properly functioning environment. Typically, if a FSMO server were taken offline for a short period of time, it wouldn't be noticed. While these roles don't have an inherent time sensitivity to them, their management and understanding of function must be well known and managed by qualified staff members.

### Design Decisions

The FSMO locations described earlier in the "Services Locations" section, offer the best configuration for where the roles should be. A select subset of the infrastructure management staff should understand these roles and where they reside, and a clear set of how to manage procedures should be established.

# Database and Server Sizing

## *Database Files*

Active Directory™ directory services and related files are stored in the system directory winnt\system32\ntds.

- winnt\ntds\ntds.dit – Active Directory database
- winnt\ntds\edb.log – log file (default circular logging)
- winnt\ntds\edb.chk – checkpoint (pointer into log file of which transactions have been committed)
- winnt\ntds\temp.edb – temporary database used for searches and open transactions
- winnt\ntds\res.log – two pre-allocated log files for use in the case that logs can't be allocated

Windows 2003 Active Directory uses circular logging for maintaining transactions in the database (Ntds.dit). The log files are maintained until the data they contain is committed to the database. It uses these log files to recover transactions if the database is shut down in an inconsistent state. For more details on logging and transactions, see the Microsoft Knowledge Base article, Q247715: "Circular Logging for Active Directory," at: http://support.microsoft.com/support/kb/articles/Q247/7/15.ASP.

## *Database Size Estimates*

Listed below are size estimates for common Active Directory objects.

- Users – 4 KB
- Contacts – 1 KB
- Organizational unit – 2 KB
- Group – 2 KB*

  * empty + 300 bytes/member for 10 members, +100 bytes/member for 100 members, 75 bytes/member for 250 members.

## *Global Catalog Database*

The size of global catalog database can vary by the number of Active Directory attributes that are marked for global catalog replication and the use of universal groups, which are always written to the global catalog.

Many of the attributes that are marked as "mandatory" in Active Directory are replicated to the global catalog. Many of the optional attributes are not replicated to the global catalog, meaning that the global catalog will not necessarily grow directly proportional to Active Directory growth.

Stated as a percentage of Active Directory size, the global catalog may range from 25 percent to more than 50 percent of the size of the database in Active Directory, depending on the number of option attributes and universal groups used in the database.

### Replication Traffic

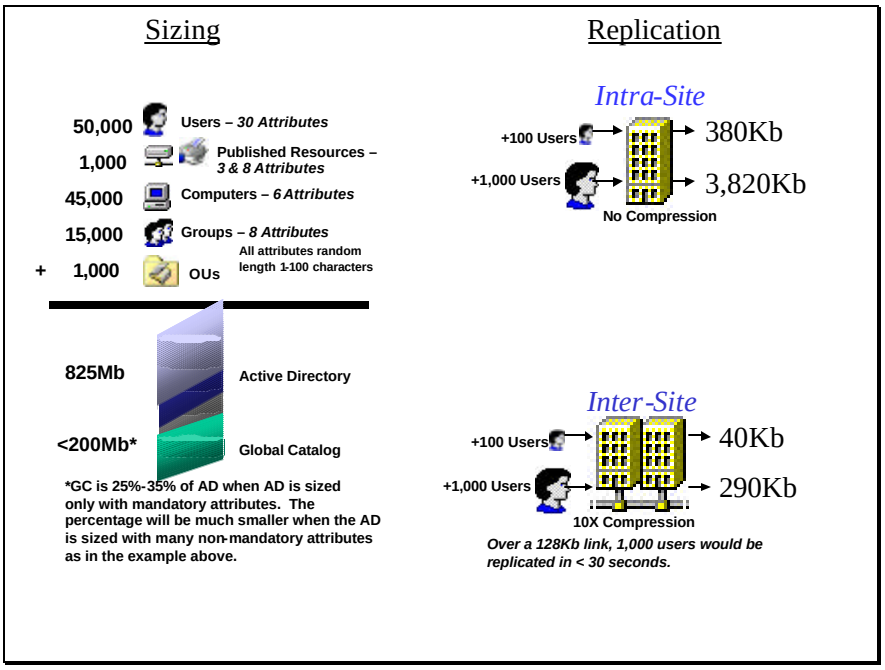



**Figure 28. Estimates of a "Real Environment"**

The size estimates above represent a "real" environment and not simply users with minimum attributes. The user counts include an appropriate number of organizational units, groups, workstations, and volumes. The replication estimates are general estimates intended to provide guidance when determining expected network replication traffic.

### Logon Traffic

For complete details of the logon and start up process, see *Windows 2003 Startup and Logon Traffic Analysis* at: http://www.microsoft.com/TechNet/win2003/win2ksrv/w2kstart.asp.

When thinking of logging on in the Windows environment, you typically think of a user pressing CTRL+ALT+DEL and entering his or her username and password (credentials) to gain access to a system. These credentials gives users access either to resources on the computer where they are working, or if the system is part of a network, these credentials give access to resources on the network such as applications, files, or printers that the user has been authorized to access.

The process that allows users to access resources does not start when the user logs on to the system, but begins well before that when the system is started. In a Windows 2003 domain environment, the computer needs to establish itself as a valid member of a domain before users are able to log on to that system and access other resources on the network.

### *Computer Startup Traffic*

Computer startup is typically less frequent than user logon traffic, since most computers are left running when they are connected to the corporate network. The startup process consists of several steps, with the most basic components outlined in the diagram below. For the tests documented here, and in the reference document above, the environment was a "pure" Windows 2003 configuration that had NetBIOS turned off ("Disable NetBIOS over TCP/IP" in the Advanced TCP/IP settings).

There are many things that can change the amount of traffic generated during the startup process, particularly Group Membership and Group Policy object processing.

Conservative estimates for a Windows 2003 computer startup is approximately 50 KB of traffic.

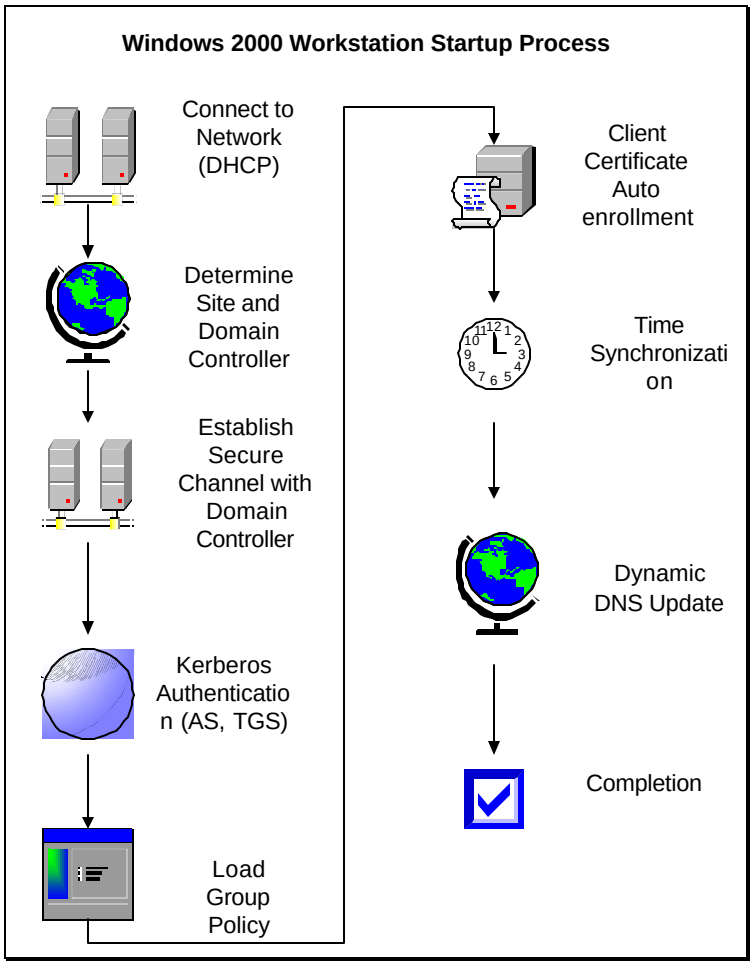



**Figure 29. Workstations Startup Process**

### *User Logon Traffic*

User logon traffic can vary widely depending on how well managed the user's environment is and how much processing Microsoft IntelliMirror® management technologies is taking place. Group memberships, roaming profiles, redirected folders, GPO changes, and many other components can dramatically increase the amount of data traveling to and from the user's workstation during logon.

As a starting point, the estimate of 30 KB per logon for essential logon processing is reasonable, but "real" implementations that take advantage of Active Directory may realistically be closer to 100-150 KB. See the complete trace of logon traffic in the reference information on *Windows 2003 Startup and Logon Traffic Analysis* at: http://www.microsoft.com/TechNet/win2003/win2ksrv/w2kstart.asp.

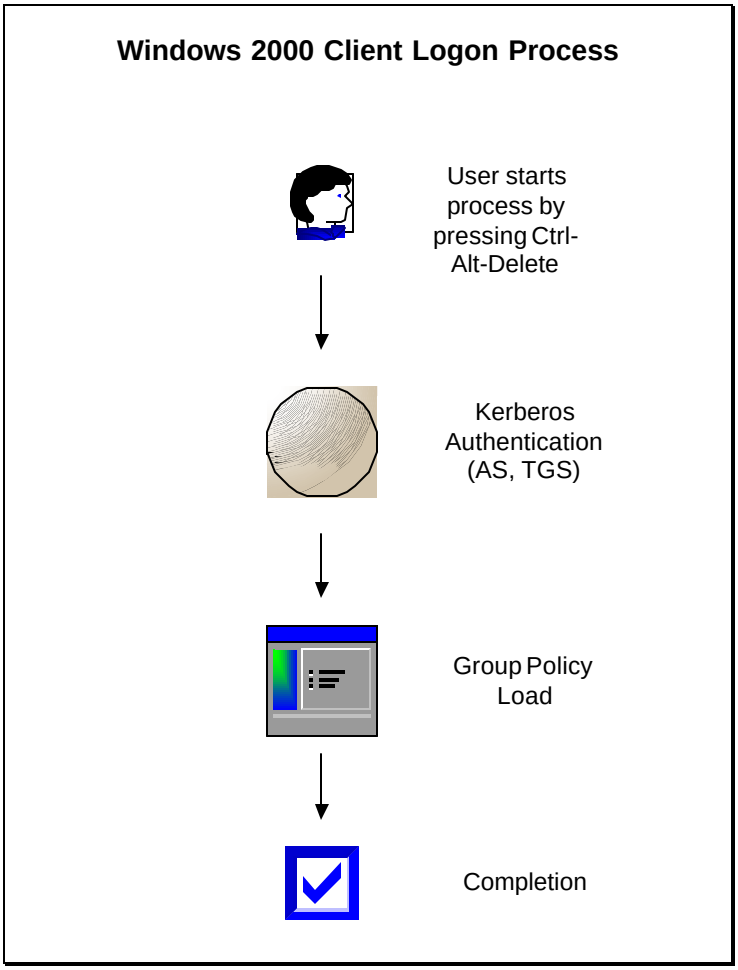



**Figure 30. Client Logon Process**

### *Situation and Requirements*

The sizing of directory infrastructure servers domain controllers and global catalogs must meet client performance expectations and provide adequate head room for anticipated growth and increased loads.

### *Design Decisions*

**Active Directory Database Sizing**

As explained above, different types of Active Directory objects increase the size of the Active Directory database by different factors. An estimation of the size of the Purdue Active Directory has been performed to ensure that the needs of the potential database size where taken into account during the Domain Controller server sizing requirements.

**Active Directory Server Sizing**

The server sizes required to support Active Directory be generally consistent across the Purdue Active Directory Forest. The number of servers required to support a particular site may vary based on number of Active Directory users/computers and other applications which depend heavily on Active Directory such as Exchange.

**Server Hardware Requirements**

Purdue uses HP (Comapq) Proliant servers as a general standard for most of their server needs. In keeping in line with these standards the server that has been selected specifically to support Active Directory at Purdue is the HP Proliant DL 380G3.

**Server Specs:**

- Intel® Xeon™ Dual 2.80 GHz or 2.40 GHz Processors with 512-KB level 2 ECC cache
- Five Peer PCI Architecture with 400-MHz Front Side Bus
- 2GB of 2-way interleaved capable PC2100 DDR SDRAM running at 200MHz, with Advanced ECC capabilities and Online Spare capabilities
- Integrated Smart Array 5i Plus Controller with optional Battery-Backed Write Cache (BBWC) Enabler option kit
- Two Compaq NC7781 PCI-X Gigabit NICs (embedded) 10/100/1000 WOL (Wake on LAN)
- Six 18GB, 15K rpm 1" Wide Ultra3/Ultra320 SCSI hot plug hard drives configured with one mirror (system partition) and one RAID 5 volume (database partition)
- 400-Watt Hot Plug Redundant Power Supplies.
- Sliding rails and cable management arm for easy serviceability and in-rack tool-less access to major components

**Hyper-Threading**

Hyper-Threading will be enabled on all Windows 2003 domain controllers. This will enable the domain controllers to gain increased processing capabilities from the new generation Xeon™ processors. Below is a summary of Intel's® Hyper-Threading technology.

Hyper-Threading Technology allows multi-threaded server software applications to execute threads in parallel within each processor in a server platform. The Intel® Xeon™ processor family uses Hyper-

Threading technology, along with the Intel® NetBurst™ microarchitecture, to increase compute power and throughput for today's Internet, e-Business, and enterprise server applications.

Hyper-Threading technology enables this thread-level parallelism (TLP) by duplicating the architectural state on each processor, while sharing one set of processor execution resources. When scheduling threads, the operating system treats the two distinct architectural states as separate "logical" processors. This allows multi-processor capable software to run unmodified on twice as many logical processors. While Hyper-Threading technology will not provide the level of performance scaling achieved by adding a second processor, benchmark tests show some server applications can experience 30 percent gain in performance. Benefits of Hyper-Threading technology include:

**High processor utilization rates**: One processor with two architectural states enable the processor to more efficiently utilize execution resources. Because the two threads share one set of execution resources, the second thread can use resources that would be otherwise idle if only one thread was executing. The result is an increased utilization of the execution resources within each physical processor package.

**Higher performance for properly optimized software**: Greater throughput is achieved when software is multithreaded in a way that allows different threads to tap different processor resources in parallel. For example, Integer operations are scheduled on one logical processor while floating point computations occur on the other.

**Full backward compatibility**: Virtually all multiprocessor-aware operating systems and multithreaded applications benefit from Hyper-Threading technology. Software that lacks multiprocessor capability is unaffected by Hyper-Threading technology.

Hyper-Threading technology delivers significant performance gains across a wide range of business productivity and enterprise applications. Hyper-Threading Technology is not restricted to servers. Intel has also introduced Hyper-Threading Technology into workstation platform, and it will be soon be available on both business and consumer desktops.

# Resource Publishing in Active Directory

Users are typically interested in two types of resources: printers and file shares (including Web sites). While other resources, for example SQL Server databases or other service access points, can be published in Active Directory, they are usually referenced via application code hidden from the user.

Common file shares, Dfs shares, and Web sites can be published in Active Directory, allowing the user to discover resources independent of using a server name or location. Dfs adds the attribute of location affinity to replicated copies of data providing a mechanism to seek out data and retrieve the copy that is closest to the user.

Locating resources in Active Directory is best executed with a query for the type of data one is after. Active Directory is structured internally for the management of resources and may not be a useful structure for users to browse. From the **Start** menu, some tools exist under **Search** that reference Active Directory, like the Printers search. The print dialog box is simply a saved search for the data type "Printers". To access a general search for any data type in Active Directory, the following access methods can be presented to the user.

## *Searching Active Directory*

### *Shortcut to Active Directory*

Perform the following steps when building a link to Active Directory for any desktop.

1) Browse to the Entire Network.

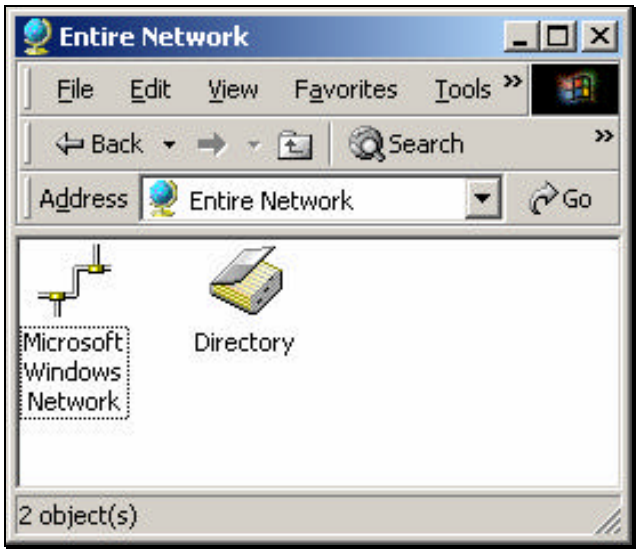



**Figure 31. Browsing the Network**

2) Select the directory (in this case Active Directory is called "Contoso," and:

   A. Create a Shortcut, which can be placed on user desktops.

   B. Right-click to get the "Find" dialog box.

   C. Save custom queries that are used often.

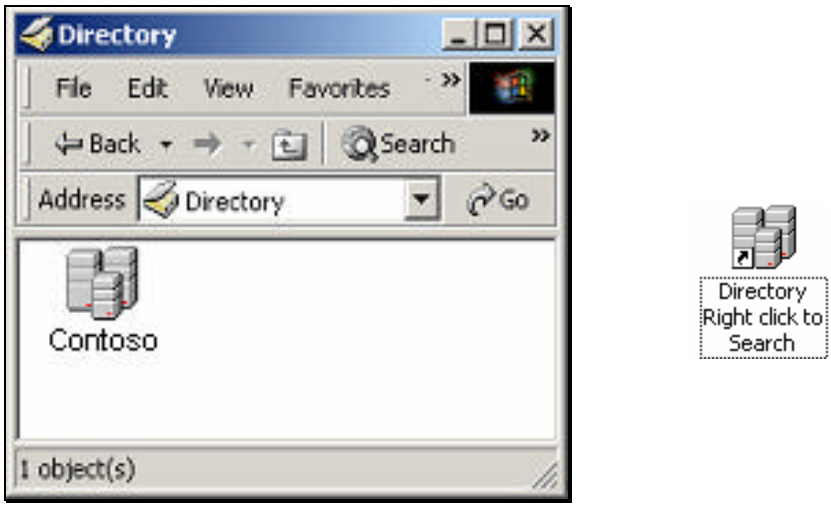



**Figure 32. Find the Directory "Contoso"**

The shortcut created (shown to the right) provides the user with a link to browse through the directory, or with a right click to "Find…" the user is taken to a query dialog box, which can be subsequently saved as a custom query.

**Note** The **Keywords:** field on the **Find Shared Folders** dialog box in the next picture can be used when publishing resources.

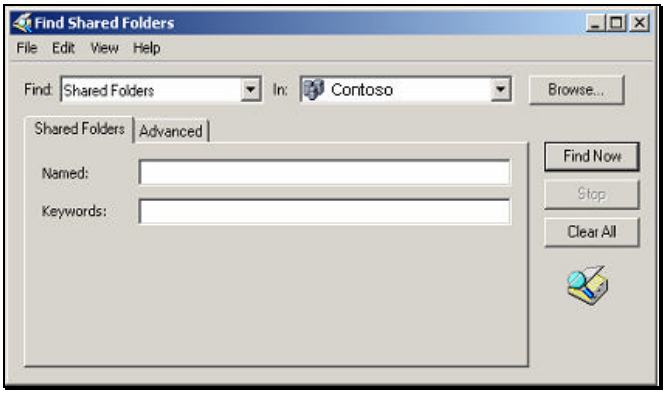



**Figure 33. Find Shared Folders**

The shortcut mentioned above can also be added to the **Start** menu as shown in the following graphic.

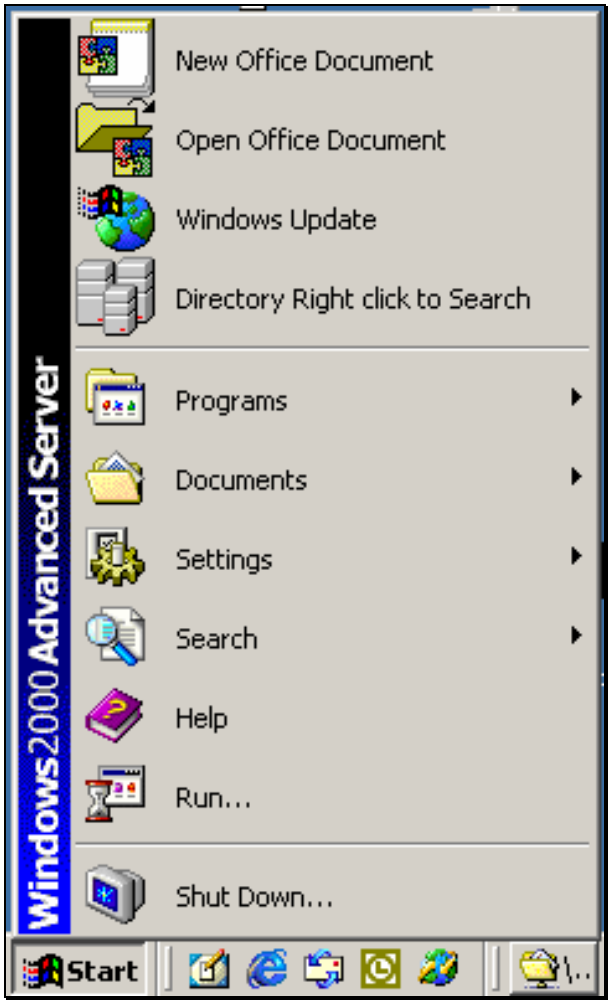



**Figure 34. Start Menu Showing Shortcut**

### *Publishing Shares*

In any organizational unit controlled by the resource manager, a newly-shared folder can be created such as the “Proposals” share below. The “Proposals” share can point to a conventional server share or it can point to some location in a Dfstree.

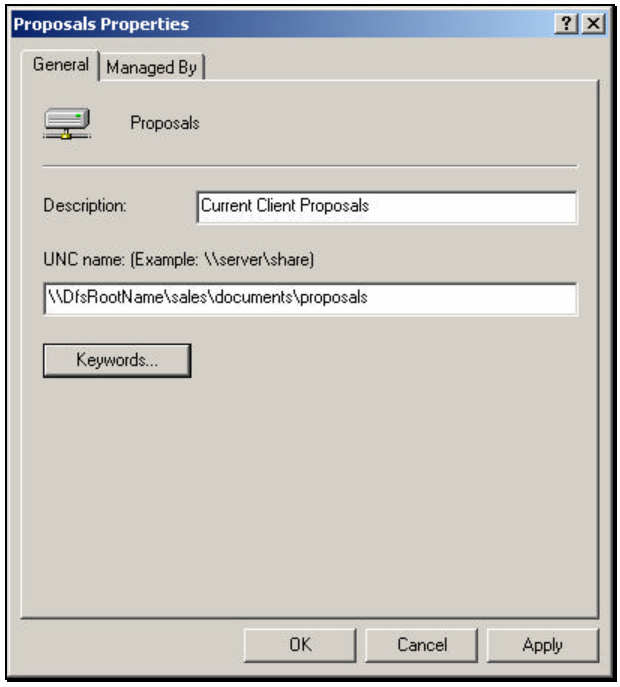



**Figure 35. Search using Keywords**

Clicking **Keywords…** allows for the addition of searchable keywords that the user can enter into a keyword search of the directory.

### *Publishing Printers*

Printer publishing is very robust in Windows 2003. While this section contains more detail than a typical architecture plan, it is appropriate for several reasons. This information is assembled from a few disparate resources and helps clarify overall usage and location affinity to physical print devices.

Windows 2003 Servers, by default, publish printers in Active Directory when they are shared. The printers published in this manner do not show up in the usual Users and Computers snap-in. Printer location and organization takes advantage of the **location** property in Active Directory. Entering this property on the printer enables the printer to be associated with a specific location.

The core structure to the location hierarchy is created by the **location** property entered into the sub-nets in **Sites and Services**. Since printers have a natural affinity with location, due to the fact they are physical devices that produce physical media, they map very well to physical network identification. Other resources like file, Web, or mail servers don't have this same physical bond since there is no need to physically visit a Web server, for instance. For this reason, printers have a customized infrastructure in Active Directory to facilitate their use. In addition, printers may be added or removed from servers, which Active Directory handles by an automatic registering and pruning process. The resulting system provides mechanisms for:

- Automatic publishing in Active Directory.
- Automatic pruning and registration.
- Inherent location affinity.

These features allow the user to quickly and efficiently find printers based on location and/or printer property (like color versus black and white), while at the same time minimizing the administrative overhead involved with maintaining the printers in the enterprise directory.

*Printers*

1) Set the GPO to "Pre-Populate Printer Search Location Text."

   This GPO enables the physical location tracking support feature of Windows 2003 printers.

   Location tracking allows the user to design a location scheme for an enterprise and assign computers and printers to locations in the scheme. Location tracking overrides the standard method of locating and associating users and printers, which uses the IP address and subnet mask of a computer to estimate its physical location and proximity to other computers.

   If you enable Location Tracking, a Browse button appears beside the Location field in the Find Printers dialog box. (To get to the **Browse** button, click **Start**, click **Search**, and click **For Printers**.) The **Browse** button also appears on the **General** tab of the **Properties** dialog box for a printer. It enables users to browse for printers by location without having to know the precise location (or location naming scheme). In addition, if you enable the "Computer Location" policy, the default location you type appears in the location field.

   If you disable this policy or do not configure it, Location Tracking is disabled. Printer proximity is estimated based on IP address and subnet mask.

2) Add a location to the Sub-Nets in Sites and Services in the form "location/sub-loc/…"

3) When creating a new printer, select the appropriate location (built from a list of locations from step 2 above).

   Now users can perform a "Find Printers" against the directory and select from the valid locations to find the printer near them. A default location can be pre-filled for the user by setting the GPO "Computer Location" by organizational unit as appropriate.

*Non-Windows 2003 Printers*

Non-Windows printers can be added to Active Directory through the graphical interface or via a script included on the installation CD, loaded to the winnt\system32 directory.

```
CScript C:\Winnt\System32\PubPrn.vbs \\NT4server\PrintShare
"LDAP://OU=OUName,DC=corp,DC=company,DC=com"
```

The location field can be used to give location affinity to Windows NT 4 printers.

### Situation and Requirements

Active Directory should deliver a resource navigation scheme that can provide the user with efficient discovery of a resource, while at the same time providing a location aware mechanism to provide the nearest replica or unique instance of that information.

### Design Decisions

At this time, some of the aspects of Active Directory publishing such as DFS is considered out of scope. Other aspects of Active Directory publishing such as Printer and Share publishing will be enabled through delegation for department administrators to publish relevant object information into Active Directory.

## Migration Considerations

While the migration process is a considerable effort, it must be recognized as a one-time cost and should not affect the end-state vision.

Key points to bear in mind:

- Destination Domains must be native sidHistory is required for Clone and Move processes).
- ClonePrinciple is destructive to password (extra-forest user movement).

### *In-place Upgrade*

The in-place upgrade is the simplest and most rapid route to Windows 2003. The Windows NT domain controllers can be upgraded in place at any desired pace, independent of the client migration. Backward compatibility is maintained and the client systems can continue to access the same NetBIOS names for the domains. In-place upgrades work well for clean directory environments that reflect the desires of the future end-state.

### *In-place Upgrade – Restructure*

In-place upgrade and restructure takes advantage of the simplicity of the in-place upgrade where most of the structure is suitable for end-state but some refinement is required. The restructure option, particularly in the consolidation of account domains, has the benefit of preserving user passwords, allowing for mass restructuring without having to deal with the administrative overhead of having users reset their passwords all at once.

### *In-place Upgrade – Clone Into*

The Clone-Into option of the in-place upgrade uses the in-place step to achieve large degrees of migration, followed by a clone-into process that selects only the pieces that are destined for the end-state system.

### *Pristine – Clone Into*

The Pristine option creates the desired end-state structure ahead of time without touching the current operating environment. This approach allows for the final production environment to be built and tested before the first "real" user is added. To populate the pristine environment, users are typically cloned from outside the forest into the new "clean" environment. Extra-forest cloning requires a reset of the user password, so this approach requires close coordination with the user community and does lend itself to a site-by-site focused upgrade approach. This approach requires that additional hardware be cycled into a site, while the users and resources are moved from the legacy configuration into the end-state system. As an alternative to the extra-forest approach, the users and other objects can be copied from other domains that have already been brought into the forest. This means that domains can be upgraded into the same forest as the pristine system, and moved from the legacy domain into the new domain.

### *Pristine plus Preload*

The Exchange directory or another corporate users database can be used to create the accounts in the pristine environment. Exchange can use Active Directory Connector (ADC) to provide a low cost/complexity method for doing this. As users are moved or cloned into the new environment, they need to be linked with the pre-created ID from the ADC or other load process. A tool or process such as ADCLEAN or a DSAddSidHist process needs to be used to add the old security identifier to the newly created id.

### Migration Tools

Included with Windows 2003 are scriptable COM objects that provide complete programmable control of the upgrade. The Active Directory Migration Tool provides a graphical interface to the common migration inter and intra forest situations. ISVs (independent software vendors) provide a suite of tools that allow for enhanced functions, modeling, and planning for complex environments and complex restructuring activities.

### Situation and Requirements

Outline and evaluate various migration options against the goal of obtaining the highest quality data, in University-wide standard format, while mitigating risk and having minimal impact on users and operations. Timing for user migration, machine rollouts, and application upgrades should be as flexible as schools or departments require.

### Design Decisions

# Operations

Core Windows 2003 operations are included as a minimum set of suggestions and are intended as a highlight of some of the main operational considerations. The Microsoft Operations Framework (MOF) team has created information that includes a suite of integrated Windows 2003 operations guides (a set of 20) along with white papers and other documentation to help customers plan and implement operational best practices throughout their environment.

The following table gives a short summary of each guide's content. Customers should work with their MCS and Microsoft Support team to complete an operations assessment and develop an entire operations plan.

| Guide Title | Description |
|---|---|
| Change Management | Tracking changes to an enterprise including but not limited to: hardware, software, communications equipment, system software, application software, processes, procedures, roles, responsibilities, and documentation that are relevant to managing, supporting, and maintaining the systems. |
| Release Management | Facilitating the introduction of software and hardware releases into managed IT environments. Typically this includes the live production environment and the managed pre-production environments. Release management is the coordination point between the release development/project team and the operations groups responsible for running the release in production. |
| Directory Services Administration | Administrating Active Directory, including monitoring the directory size, replication traffic, and backup and restore functions. Directory services administration may also include interfacing with other directory services within a company |
| Storage Management | Optimizing data storage, such as database management and backup/restore activities, as well as media management, such as storage array networks, optical jukeboxes, and RAID configurations. |
| Security Administration | Performing security administration functions within Windows 2003, including identification, authentication, access control, confidentiality, integrity, non-repudiation, and auditing. |
| Service Monitoring and Control | Monitoring of hardware, software, and services in an enterprise, taking appropriate action when faults occur, and generally measuring whether service level agreements are being met. |
| Configuration Management | Tracking the relationships of objects in an enterprise, including but not limited to: hardware, software, communications equipment, system software, application software, processes, procedures, roles, responsibilities, and documentation that are relevant to the running, supporting, and maintenance of systems. |
| Financial Management | Overseeing the methods and processes to financially manage an information technology (IT) environment. |
| Print/Output Management | Managing all data that is printed or compiled into reports that are distributed to various members of the organization. The print and output management team must ensure that any sensitive printed material is properly secured. |
| Incident Management | Managing which faults and disruptions in the use or implementation of IT services as reported by customers or IT partners are managed and controlled. |

| | |
|---|---|
| Service Level Management | Managing the quality of IT services. The aim of SLM is to negotiate, monitor, and maintain service level agreements between the IT service provider and its customers. Service level management is a key service delivery discipline that requires considerable interface with the other service management disciplines. |
| Systems Administration | Administering the whole distributed processing environment; responsible for keeping the enterprise systems running. |
| Service Desk | Managing processes and support procedures necessary to successfully operate a functioning service desk. |
| Service Continuity Management | Planning to cope with, and recover from, an IT disaster. This subject also provides guidance on safeguarding the existing systems by the development and introduction of proactive and reactive countermeasures. |
| Capacity Management | Ensuring that appropriate IT resources are available to meet business requirements. The capacity management process is a key element in providing quality IT services to meet evolving business needs on time and at minimum cost. |
| Network Administration | Managing the physical network inside an enterprise and addressing the areas of responsibility in the daily administration of a network operations center. This document focuses on network administration processes and methodology, including people management, fault management, configuration management, performance management, service level management, security management, and contingency planning. |
| Availability Management | Ensuring that services are available and that service disruptions are minimized. |
| Job Scheduling | Managing the continuous organization of jobs and processes into the most efficient sequence, maximizing system throughput and utilization to meet SLA requirements. Job scheduling is closely tied to service monitoring and control, and represents day-to-day capacity management. |
| Problem Management | Investigating and resolving of the root causes of IT service incidents, faults, and disruptions. |
| Workforce Management | Providing best practices to continuously assess key aspects of the IT workforce management; covers topics such as recruiting, skills development, knowledge transfer, competency levels, team building, process improvements, and resource deployment. |

The following sections describe in detail some Windows 2003 operational practices that should be implemented in conjunction with the entire MOF Windows 2003 operations plan.

# Time Synchronization

The Windows Time service (W32Time) is required by the Kerberos authentication protocol. The purpose of the time service is to ensure that all Windows 2003-based computers within an enterprise use a common time. The Windows Time service uses a hierarchical relationship that controls authority and does not permit loops to ensure appropriate common time usage.

Time should not be manually configured on downstream domain controllers and servers. All systems should receive their time through the synchronization hierarchy. A time skew of more than five minutes can cause Kerberos tickets to become invalid, causing authentication problems and communication failures with other computers.

### *Configure Time Servers in Windows 2003*

Windows 2003 computers use the following hierarchy by default:

- All client desktops nominate as their in-bound time partner the authenticating domain controller.
- All member servers follow the same process as client desktops.
- All domain controllers in a domain nominate the primary domain controller (PDC) flexible single master operation (FSMO) as their in-bound time partner.
- All PDC FSMOs follow the hierarchy of domains in the selection of their in-bound time partner.

Following this hierarchy, the PDC FSMO at the root of the forest becomes authoritative for the enterprise, and should be configured to gather the time from an external source. This fact is logged in the System log on the computer itself as Event ID 62. Administrators can configure the Windows Time service on the PDC FSMO at the root of the forest to recognize an external Simple Network Time Protocol (SNTP) time server as authoritative, using the following NET TIME command:

```
net time /setsntp:server list
```

There are several SNTP time servers run by the U.S. Naval Observatory that are satisfactory for this function. For example:

- ntp2.usno.navy.mil at 192.5.41.209
- tick.usno.navy.mil at 192.4.41.40
- tock.usno.navy.mil at 192.5.41.41

  **Note** SNTP defaults to using UDP port 123. If this port is not open to the Internet, you cannot synchronize your server to Internet SNTP servers.

### *Situation and Requirements*

The SNTP time service must be implemented at the Purdue.LCL domain to support time services within the entire Active Directory.

### *Design Decisions*

The Domain Controller in the Purdue.LCL domain which is hosting the Primary Domain Controller Emulator (PDC Emulator) Flexible Single Master of Operation (FSMO) role will be configured to synchronize to at least two U.S. Naval Observatory Time Servers. Doing this ensures that the Forest Root PDC Emulator is synchronized with "Real-Time", which following the hierarchy of SNTP services in Active Directory ensures that all Domain Controllers and Windows 2003 or greater clients will automatically configure their clocks correctly.

Note: It is important to remember that if the PDC Emulator roll is ever transferred or seized from the Root domain FDC Emulator that the SNTP service on the new PDC Emulator must be manually configured to ensure that the time stays in synch with "Real-Time".

# Backup and Restore

The Windows Backup utility included with Windows 2003 can perform backup and restore of data to a system. There are other third-party products that also do this. A full disaster backup and recovery strategy should be developed and tested.

## *Backup*

System State in Backup is a collection of system-specific data that can be backed up and restored. For all Windows 2003 operating systems, the System State data includes the registry, the COM+ Class Registration database, and the system boot files. For Windows 2003 Server, the System State data also includes the Certificate Services database (if the server is operating as a certificate server). If the server is a domain controller, the System State data also includes Active Directory services database and the SYSVOL directory.

These components cannot be backed up or restored separately due to interdependencies. System State should ALWAYS be backed up.

## *Checking the Health of the Active Directory Database*

The backup process provides an excellent opportunity to check the condition of the Active Directory database. An offline backup of the database can be performed following a rigorous database check. This gives assurance that the database that was backed up is certified as a "clean" copy. A batch file containing the following lines performs a check on the condition of the database:

```
ntdsutil "popups off" quit
ntdsutil files recover quit quit
ntdsutil files integrity quit quit
ntdsutil "semantic database analysis" go quit quit
ntdsutil "popups on" quit quit
pause
```

This process needs to be performed on a domain controller that is booted to Active Directory Restore mode, which takes Active Directory offline.

The first step is a soft recovery and ensures that all committed transactions are made to the database file. Soft recovery is performed automatically when the domain controller starts if the previous shutdown was not clean.

The second step uses the integrity command, which can detect low-level (binary level) database corruption. The integrity command reads every byte of the data file. Therefore, depending upon the size of your database, the process might take a considerable amount of time. The integrity command also makes sure that the correct headers exist in the database itself and that all of the tables are functioning and are consistent.

The third step is the semantic check, which detects data consistency in the Active Directory database. The semantic test checks several aspects of the data including a check to ensure that each object has a GUID, a distinguished name, and a non-zero reference count.

This database health check should be performed regularly as part of the scheduled backup process.

### Restoring a Domain Controller

If a domain has more than one domain controller, you can restore it in two ways. You can either restore Active Directory using replication with another domain controller, or restore Active Directory from backup media. If there are no domain controllers in the domain, then you can only restore Active Directory from backup media.

In order to restore the System State data on a domain controller, you must first start your computer in a special safe mode called directory services restore mode. This allows you to restore the SYSVOL directory and Active Directory database.

### Restoring Active Directory with a Replica

Use the Active Directory Installation Wizard to reinstall Active Directory, promoting the server to a domain controller. Active Directory and SYSVOL will be brought up-to-date through replication from a domain controller. Before you run the Active Directory Installation Wizard, delete any references to the old domain controller using the Sites and Services snap-in.

### Restoring Active Directory from Backup Media

Use Backup to restore the System State, which will recover Active Directory, File Replication Service (including SYSVOL) and Certificate Services (if installed). If the domain controller computer has been replaced because of malfunction or the network adapters have been replaced, you might need to reconfigure the network settings manually.

If you are restoring the System State data to a domain controller, you must choose whether you want to perform an authoritative restore or a non-authoritative restore. The default method of restoring the System State data to a domain controller is non-authoritative.

> **Note** Since the machine account password will be reset to the date/time of the restore, the domain controller computer account secure channel will possibly need to be reset while restoring a domain controller system state from backup media. This involves both authoritative and non-authoritative restores.

### Active Directory Leaf and Branch Restoration

*Non-Authoritative Restore*

In this mode, any component of the system state that is replicated with another domain controller, such as the directory service or the File Replication service (including the SYSVOL directory), will have its original update sequence number. The Active Directory replication system uses this number to detect and propagate Active Directory changes among the servers in your organization. Thus, after you restore the data, it will be brought up to date by replication. For example, if the last backup was performed a week ago, and the system state is restored using the default restore method (non-authoritative), any changes made subsequent to the backup operation will be replicated from the domain controllers.

*Authoritative Restore*

In some cases, you may not want to replicate the changes that have been made subsequent to the last backup operation. In other words, there may be instances where you want all replicas to have the same state as the backed-up data. To achieve this state, you must perform an authoritative restore.

For example, you have to perform an authoritative restore if you inadvertently delete users, groups, or organizational units from the directory service, and you want to restore the system so that the deleted objects are recovered and replicated. To do this, you need to run the Ntdsutil utility after you have restored the data but before you restart the domain controller. This utility lets you mark objects as authoritative. When an object is marked for authoritative restore, its update sequence number is changed so that it is higher than any other update sequence number in the Active Directory replication system. This ensures that any replicated or distributed data you restore is properly replicated or distributed throughout your organization.

Authoritatively restore only the Engineering container:

```
Ntdsutil
Authoritative restore
Restore subtree OU=ENGINEERING,DC=CONTOSO,DC=COM
```

For additional information on Backup and Restore, see the "Disaster Recovery" chapter of the *Branch Office Deployment Guide* at:
http://www.microsoft.com/windows2003/techinfo/planning/activedirectory/branchoffice/dply10.asp

For additional information on Active Directory Disaster Recovery, see:
http://www.microsoft.com/windows2003/techinfo/administration/activedirectory/addrstep.asp

For additional information on Windows 2003 Server Disaster Recovery Guidelines, see:
http://www.microsoft.com/windows2003/techinfo/administration/fileandprint/recovery.asp

## Situation and Requirements

All domain controllers will be backed up on a persistent scheduled backup. Because the Active Directory changes daily it is extremely important to have a current backup of any domain controller in case their is a catastrophic event or failure that might cause the need for an Active Directory Restore.

## Design Decisions

### *Backup Schedule*

Each domain controller will be configured to back up its System State Data on a daily schedule using NTBACKUP.

### *Backup Location*

Each domain controller will be configured to back its System State Data to a central file server location. Once all of the domain controller backups are completed all of the backup files will then be backed up to tape. This gives the Active Directory Administrators the ability to quickly restore the Active Directory from a file if an Active Directory restore ever becomes necessary.

### *Restore*

There are several different Active Directory restore solutions that are best used in certain circumstances:

1. Re-Install and Replicate Restore

The Re-Install and Replicate Restore is generally used more than other types of restores. A Re-install and Replicate Restore is used when there is a complete server loss or hardware failure on a domain controller that cannot be initially recovered from. In this type of restore, when the server is failed, it is just deleted from the Active Directory, and the server is re-installed or replaced. The server is brought on-line just as though it where a new domain controller being introduced into the domain. All of the Active Directory information is then replicated to the server bringing it up to date with the rest of the domain controllers.

Because domain controllers hold a complete Multi-Master copy of the Active directory for its domain you can just replicate the existing data instead of restoring the directory from backup and replicating changes.

2. Non-Authoritative Restore

   A Non-Authoritative Restore is used in situations where the Active Directory may be too large or take to much time to replicate from a different Active Directory Site in order to restore it. In this situation the failures would be similar to that of a failure that would require a Re-Install and Replicate Restore but the full replication would take too much time or utilize more network bandwidth than can be allowed such as a T-1 link to one of the outlying campus location during the day when that link is heavily utilized already.

   The Non-Authoritative Restore will restore the Active Directory on that domain controller to the state in which is was when it was backed up, then immediately request an update replication of all missing Update Sequence Number (USN) changes to the Active Directory. This means that only what has changed since that backup was taken will be replicated across the WAN link.

3. Authoritative Restore

   The Authoritative Restore is not generally used for server or domain controller failures, but is used more for administrative errors. If objects are deleted from the Active Directory, then these changes will replicate to all of the other domain controllers.

   If objects are accidentally deleted and only those objects need to be restored then a domain controller will need to be restored with a backup that contains these objects using an Authoritative Restore. The Authoritative Restore will place new time stamps and USNs on these objects so that they are created similarly to new object on the other domain controllers, but will contain all of the original object attribute information.

# Server Recovery

The primary reason Windows NT 4 servers were configured with a file allocation table (FAT) partition was to provide the ability to boot to an MS-DOS® environment and be able to edit and replace system files as needed in an attempt to recover a down server.

Key features have been added to Windows 2003 to provide for this important functionality while having the benefit of configuring your file system with NTFS for best performance and reliability.

## *Safe Mode Startup Options*

Windows 2003 can start a safe mode menu if F8 is pressed at startup. This menu gives a number of start-up options. In the event a server does not start up properly, booting to the Startup menu is the first process in a recovery attempt. The following options are available:

- **Safe Mode**. Starts Windows 2003 using only basic files and drivers (mouse, except serial mice; monitor; keyboard; mass storage; base video; default system services; and no network connections). If your computer does not start successfully using safe mode, you may need to use the Emergency Repair Disk (ERD) feature to repair your system.
- **Safe Mode with Networking**. Starts Windows 2003 using only basic files and drivers, plus network connections.
- **Safe Mode with Command Prompt**. Starts Windows 2003 using only basic files and drivers. After logging on, the command prompt is displayed instead of the Windows desktop, Start menu, and Taskbar.
- **Enable Boot Logging**. Starts Windows 2003 while logging all the drivers and services that were loaded (or not loaded) by the system to a file. This file is called ntbtlog.txt and it is located in the %windir% directory. Safe Mode, Safe Mode with Networking, and Safe Mode with Command Prompt add to the boot log a list of all the drivers and services that are loaded. The boot log is useful in determining the exact cause of system startup problems.
- **Enable VGA Mode**. Starts Windows 2003 using the basic video graphics adapter (VGA) driver. This mode is useful when you have installed a new driver for your video card that is causing Windows 2003 not to start properly. The basic video driver is always used when you start Windows 2003 in Safe Mode (either Safe Mode, Safe Mode with Networking, or Safe Mode with Command Prompt).
- **Last Known Good Configuration**. Starts Windows 2003 using the registry information that Windows saved at the last shutdown. Use only in cases of incorrect configuration. Last Known Good Configuration does not solve problems caused by corrupted or missing drivers or files. Also, any changes made since the last successful startup will be lost.
- **Directory Service Restore Mode**. Not applicable for Windows 2003 Professional. This is for the Windows 2003 Server operating system and is only used in restoring the SYSVOL directory and Active Directory on a domain controller.
- **Debugging Mode**. Starts Windows 2003 while sending debug information through a serial cable to another computer.

### Recovery Console

This console can be used in a recovery attempt of a server. The Recovery Console can be invoked by using the three Windows NT startup floppy disks or the Windows NT CD-ROM. The Console can also be installed by using the **Winnt32 /cmdcons** switch either during or after Windows 2003 is installed. When the console is installed on the system it can be invoked by selecting Recovery Console from the Startup menu.

Once in the console, there are a limited number of commands that can be executed such as file restore, Registry restoration, disabling problematic services and more.

#### *Restrictions and Limitations*

- The console has the following broad security restrictions and limitations.
- Individual commands may possess certain limitations. These commands should be reviewed and understood.
- You cannot copy a file from the local hard disk to a floppy disk. You can copy from the floppy disk to any hard disk, and from hard disk to hard disk, however.
- You can only view the %WINDIR% directory and subdirectories of the Windows 2003 installation you are currently logged in to. If you try to view another %WINDIR% directory, you will get an "Access Denied" error.
- See the Microsoft Knowledge Base article, "Description of the SET Command in Recovery Console" at: http://support.microsoft.com/support/kb/articles/Q235/3/64.ASP for information on how to use the Set command with the "allowallpaths" option to be able to access the entire hard drive.
- If the SAM hive is corrupt or missing on a computer with a single installation of NT, you will not be able to use the console since you cannot log on to the system to verify your identity.

### Emergency Repair Disk (ERD)

An emergency repair disk can be generated from the Windows Backup Utility. The Windows 2003 emergency repair feature can be used to fix problems that may be preventing you from starting your computer.

### Usage Scenarios

#### *Safe Mode*

In the event of a server being down and unable to start, use F8 upon starting to get the Safe Mode Boot menu. Starting in Safe Mode starts the server with limited drivers and, once up, logs can be inspected and repairs performed as required.

#### *Last Known Good*

The Last Known Good Configuration is useful when a configuration change was just made and as a result the server will not boot.

*Recovery Console*

The following information is useful when working with recovery consoles:

- Create a batch file for recovery commands to automate common recovery tasks. For example, automate the disabling of several services (such as Telnet, Server, Telephony, or Messenger) simultaneously, or configure Recovery Console for Auto-Logon and automate a restore of the registry. Use the command:

  `batch<nameofbatchfile> <file to contain output of batchfile>.`

  - **Scenario 1**. A user attempts to start the computer and receives an error message that "Advapi32.dll is corrupt or missing" and the computer displays a Stop error. The user can start the cmd console, and log on to the Windows 2003 installation. Copy Advapi32.dll from the CD to the %windir%\system32 directory of the system. The user then exits the cmd console, and restarts the machine.
  - **Scenario 2**. If the system hive (registry) becomes corrupt and needs to be replaced from a backup, the administrator can copy a backup copy of the system hive to a floppy disk. The administrator starts the cmd console on the computer and logs on to the Windows NT installation. Copy the system hive from the floppy to the %windir%\system32\config directory. The administrator exits the cmd console, and restarts the machine.
  - **Scenario 3**. A virus has infected the master boot record (MBR), preventing Windows 2003 from starting (an error message, 0x04, appears in a blue screen). The administrator starts the cmd console, and runs the fixmbr command. In some cases, the fixboot command may be used as the boot sector may have been corrupted also.

*Emergency Repair Disk*

When the system cannot start up under safe mode and you cannot log into the Recovery Console, then recover using the ERD is an option.

## Design Decisions

The Recovery Consol will be installed on every Domain Controller, and should also be installed on every Windows 2003 server. An Emergency Repair Disk (ERD) will be made for each Domain Controller and kept in a safe location so that they can be used in the even that a critical failure makes the recovery console un-usable. The ERDs should be periodically updated. The best time for this is after major Service Packs are installed.

The Recovery Consol will be used to perform all advanced Active Directory Server restoration using the Authoritative and Non-Authoritative restore procedures.

# Monitoring

After completing your deployment of Active Directory, it is very important to continue to perform quality assurance checks on your domain controllers. Doing so allows detection of any potential problems before they have a chance to cause a significant impact on the environment and the user's ability to access network resources.

Quality assurance checks should be performed on domain controllers on a regular basis. These checks should be performed at least once a day, after a replication interval. This allows verification that the replication was successful for the day or period, detection of any issues that may have occurred, and correction of any issues before the next replication interval. If the quality assurance check is performed before a replication interval, you may not become aware of any problems until a second replication cycle occurs, which could allow the problem to have a larger impact on the environment.

Three main areas should be examined as part of the ongoing quality assurance for domain controllers:

- General domain controller monitoring
- Active Directory replication monitoring
- FRS replication monitoring

This Active Directory branch office guide includes a set of quality assurance scripts that can be used to perform a daily quality assurance check on the branch office environment. These scripts should be scheduled to run daily on all domain controllers.

For more information, see Chapter 9, "Ensuring Active Directory Health in Branch Office Environments" of the *Active Directory Branch Office Planning Guide* at: http://www.microsoft.com/windows2003/techinfo/planning/activedirectory/branchoffice/dply09.asp

For more information about monitoring scripts, see *Windows 2003 Scripts for Setting up Active Directory Branch Office Environments* at: http://www.microsoft.com/downloads/release.asp?ReleaseID=26816.

It is important to perform some general monitoring of the domain controllers. Two areas to monitor are processor utilization and available disk space.

The performance counters discussed in this section should always be monitored on the bridgehead domain controllers, and on field DCs if a problem is suspected.

## *Processor Utilization*

Monitoring the processor utilization on your domain controllers will allow you to determine if your domain controllers are being overloaded by logon or, on bridgehead servers, by replication. This will also allow you to verify that you are meeting your service level agreements.

To monitor a domain controller's processor utilization you can use System Monitor or Performance Logs and Alerts to monitor the Processor\% Processor Time counter.

### Available Disk Space

Monitoring available disk space is important as problems can arise with your domain controllers if the partition storing any of the following runs out of available disk space:

- Active Directory database files
- Active Directory log files
- SYSVOL folder

By default, these are stored in either C:\WINNT\NTDS or C:\WINNT\SYSVOL.

To monitor the free disk space on the partition containing your Active Directory database and log files and the SYSVOL folder, use System Monitor or Performance Logs and Alerts to monitor the LogicalDisk\Free Megabytes counter.

### Monitoring Domain Controller Performance

In addition to monitoring the processor utilization and free disk space, you can also monitor domain controller performance by tracking performance counters in Performance Logs and Alerts and then viewing the results in System Monitor. For example, if you want to monitor whether a server is regularly receiving and applying directory replication updates, you can select one or more counters from the NTDS performance object, and then view the current activity in System Monitor.

Use the counters of the following two performance objects to monitor domain controller performance:

- NTDS object counters
- Database object counters

  **Note** Before you can use the Database performance object, you must install it manually. Instructions are provided later in this module.

The NTDS and Database counters should all show some activity when monitored over a period of time. However, the amount of activity will greatly depend on the environment. Factors that affect the activity include the number of domain controllers, number of sites, number of clients, how often replication is scheduled, the number of directory changes that occur, and so on.

A complete list of performance counters and installation instructions can be found in Chapter 9, "Ensuring Active Directory Health in Branch Office Environments" of the *Active Directory Branch Office Planning Guide* at:
http://www.microsoft.com/windows2003/techinfo/planning/activedirectory/branchoffice/dply09.asp

### Active Directory Monitoring

An important aspect of any Active Directory deployment that is often overlooked is ongoing monitoring of the environment. Ongoing monitoring will allow you to detect any issues that may arise in your environment and correct them, hopefully before they impact your environment or users.

If ongoing monitoring is not performed on a regular basis, a problem could arise with a domain controller that would not be identifiable until it started to impact users. By the time user problems are reported and the cause of the issue is identified, the problem could be having a larger impact on the environment than if it had been detected with the ongoing monitoring process.

### What to Monitor

When monitoring Active Directory, the key areas to monitor include:

| Area to Monitor | Utilities for Monitoring |
|---|---|
| DNS and Network configuration | Netdiag.exe |
| Connection objects | Repadmin.exe and Replmon.exe |
| Replication | Dcdiag.exe, Repadmin.exe, and Replmon.exe |

If problems occur in any of the above areas, there will be an impact on Active Directory. Chapter 9, "Ensuring Active Directory Health in Branch Office Environments" of the *Active Directory Branch Office Planning Guide* describes the utilities that can be used to monitor these areas.

### Monitoring FRS Replication

FRS is a multithreaded replication engine used to replicate files between different computers simultaneously. When you add, remove, or modify the contents of the SYSVOL folder on a domain controller, FRS replicates those changes to the SYSVOL folders on all other domain controllers in the domain.

FRS uses the same connection objects as Active Directory when replicating SYSVOL content. Therefore, it uses the same schedule as Active Directory for inter-site replication.

It is very important to monitor FRS replication and ensure that it is functioning in your environment. If FRS is having problems replicating to domain controllers and you are using Group Policy, Group Policy changes will not replicate to the domain controllers that are experiencing replication problems.

Unfortunately there are not many tools for monitoring FRS replication. Three methods can be used to monitor FRS replication:

1) A pragmatic approach is to copy a "tag file" to the SYSVOL share. After the next replication interval for a domain controller's replication partners, you can check the SYSVOL share of the replication partners to see if the "tag file" replicated successfully.

2) Examine the FRS log files for errors. The log files generated by FRS are a comprehensive way to follow the actions performed and any problems encountered by FRS.

3) Use Ntfrsutl.exe from the Microsoft Windows 2003 Resource Kit to view FRS information.

The QA_Check.cmd script detailed in the Branch Office Deployment Guide uses methods 2 and 3 to monitor the FRS service for problems. For more information about monitoring scripts, see Windows 2003 Scripts for Setting up Active Directory Branch Office Environments at: http://www.microsoft.com/downloads/release.asp?ReleaseID=26816.

### Event Log

Event logs should be examined regularly for any errors or warnings about the operation of the system in general, the security of the system and the condition of Active Directory. There are several tools available for filtering and capturing Event Log events. Some of the tools include dumpel.exe, eventlog.pl, uptime.exe, elogdmp.exe, eventquery.pl and others. Using a combination of tools and scripts, a comprehensive view of all systems can be achieved. Monitoring more than several servers can become a significant task requiring a great deal of scripting and filtering. Several monitoring tools available today can provide this type of information in pre-processed fashion with reports and proactive alerting mechanisms.

### DNS Consistency

DNS availability and consistency with Active Directory should be monitored. There are several DNS records that are registered for domain controllers. These records are essential for proper replication, authentication, and service discovery. The Netdiag tool mentioned above can provide a basic check for DNS record consistency.

### Operations Management Tools

Several management and monitoring tools have been developed to deliver a comprehensive management solution ranging from proactive alerting to capacity planning. While the health of the Windows 2003 enterprise can be determined though scripts and the tools provided with the system and in the Windows 2003 Resource Kit, it can be a significant task to build in the appropriate level of automation. Well-managed enterprises of significant size would easily achieve a return on investment with a properly deployed management tool.

### Situation And Requirements

All domain controllers must be actively monitored and maintained to ensure maximum up time.

### Design Decisions

All of the domain controllers will at a minimum be monitored using the integrated monitoring tools. Currently Purdue is using HP Openview to monitor many aspects of the network and systems. There are plug-ins for HP Openview that will monitor all of the service specific functions of Active Directory such as the domain controller database itself, replication etc. using the HP Openview SPI (Smart Splug-in) for Microsoft Active Directory