

Planning a Windows 2000 Network Migration

This chapter focuses on identifying the type of Windows NT-to-Windows 2000 (Win2K) migration you need to undertake, planning the project, and preparing your network for the migration. Any migration, or unification of two infrastructures, can significantly disrupt your business and require a significant amount of resources and staff. Thus, you need to carefully consider the implications and timing of the migration to ensure that it's worthwhile.

As I move through this chapter, I'll raise potential issues and discuss solutions and tools that you can use to help plan and deliver Win2K migration.

Identifying Your Migration Goals

Migrating to Win2K has undeniable advantages for your business. It also satisfies certain technical requirements. Thus, identifying your migration goals need to focus on your business goals and your technical goals.

Assembling a Migration Team

With Windows NT, Microsoft began to exert an influence on other parts of your network—for example, Domain Name System (DNS) and Web services. With Win2K, the impact on your network will be even greater. This creates many potential ownership issues. You need to assemble a team to ensure that the project proceeds according to plan and to resolve any ownership issues to the satisfaction of your business requirements.

In addition, you need to work with many departments and groups, even in a small business, to ensure a successful migration. As I outlined in Chapter 1, the following groups are the most important members of your migration team. Of course, the size of your business will determine the size of the team.

- Project management (program management)
- Networking
- DNS support
- Security
- Messaging (for example, Exchange)
- Directory services
- Technology
- Help Desk and support
- Communications
- Facilities

☞ The team that will have the most experience with Active Directory (AD) technology might very well be your Exchange team. While AD may be a more advanced service than the Exchange directory, there are many lessons to be learned from structuring the Exchange directory and replication. In fact, most Exchange teams will likely already be looking at Exchange 2000 (E2K), a product that requires Win2K to be in place.

In addition, some companies may already have an existing directory-services team, and despite a number of differences between AD and more general directories, be sure to leverage such expertise as much as possible.

These team members will be focused on the technical and logistical aspects of your migration. In addition, you need to identify all of the stakeholders in the project with a business interest, such as:

- Departmental heads or representatives
- End user representatives
- Training

Examples of such key players in many organizations are Human Resources, Customer Management, Sales, Marketing, and so on. These departments will play a key role in the success of your migration, so make sure you involve them. For example, I recommend using end user representatives to direct any issues, concerns, and even potential praise to the core migration team.

One of the most important aspects of your migration planning is understanding what you want to gain. There is a cost associated with migrating to any new system, even a single desktop, so any business will need to be assured that this isn't just a technology craze. Thus, it's absolutely critical to ensure that your goals and objectives are clearly defined. Identifying your key stakeholders is an important part of this process. Once you've assembled the members of your migration team, you can bring them together to define your migration goals.

Determining a Planning Methodology

One of the most important aspects of any network migration project is the methodology you employ to actually plan and implement the migration. For my own migration projects, I often use what's known as the "4DS" planning methodology. A diagram illustrating this concept is shown in Figure 2.1.

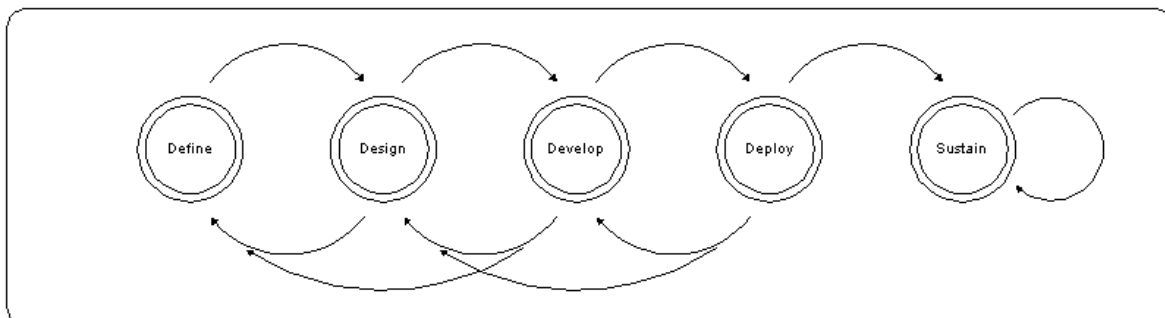


Figure 2.1: *The 4DS project planning methodology life cycle.*


This book isn't designed to describe this particular methodology in great detail. However, the 4DS process is fairly basic in its design, so let's take a few moments now to discuss the basic steps and terminology associated with this concept so that we may refer back to it later in the book.

- Define—Determine the scope and deliverables you want to provide and in what environment.
- Design—Set down the details of what you plan to deliver, then document and create a higher-level project plan.
- Develop—Create the code and identify the tools you need and deliver a detailed project plan. At this stage, you should also be ready to pilot or run your project through a quality assurance (QA) or testing phase.
- Deploy—Deliver the solution according to your plan.
- Sustain—As with any product, you need a plan to sustain or maintain it through its various life cycles, including new deployments or updates.

This is a standard progression and loop cycle, one used by most project management methodologies. As with many projects, you may notice a loop effect as you go through the definition and design cycles. This is because as you learn more about what you plan to deploy, you may change your definition of what you plan to deploy. Thus, most of this chapter focuses on defining and designing your migration project. Keep in mind that you can always step back should you realize that the project isn't going in the right direction or if new information comes to light that changes its perspective. Let's start to work on your reasons to migrate and how you can begin planning.

Return on Investment and Other Business Goals

Chapter 1 discussed a large number of the new features in Win2K. These will go a long way to helping you understand the technical efficiencies that Win2K can offer your organization. However, as you evaluate your goals for migrating to Win2K and E2K, the first step is actually looking at your business requirements. Once you've done that, you can apply the necessary technical solutions.

 Many Win2K deployments flounder or fail because there is no compelling business need, not enough staff that are experienced with the product, and too much complexity to make it successful. Before you can successfully deploy Win2K, you must identify and mitigate these issues.

The goals of the business generally focus on maintaining the status quo unless there are guaranteed, and sometimes significant, returns of at least some of the following:

- Minimizing the cost of the infrastructure
- Minimizing the cost of supporting the infrastructure
- Improving employee productivity
- Increasing the business functionality of existing systems

- Creating competitive advantage
- Supporting a new initiative or partner program.

Technical Goals

Secondary to the business goals of migrating to Win2K are the technical goals and requirements. Where the business and technical aims intersect are your best bets for gaining support for this new deployment. Thus, your technical goals should align with your business goals. The following list provides these key technical goals and requirements:

- **Security**—This is always an area that many organizations believe they provide, yet often fail to do so. Win2K provides many mechanisms to help ensure that security settings are maintained across the entire network. Now is a good time to involve your security team and discuss whether the existing services are secure enough. Also, consider using the migration project as the reason to perform a security audit on your network and resources to determine whether you need to change your existing security policy or, as some companies might find, more than one set of policies.
- **Manageability**—Any organization will want to minimize the number of administrators required to perform a specific task and the associated costs. Maximizing resource usage and minimizing resource costs are always important to any organization.
- **Availability**—Most organizations will want to ensure that their network services are available whenever the business requires them. This should also be a goal of your migration project so that the current environment is disrupted as little as possible. This includes maintaining the required access controls and other security settings.
- **Scalability**—Many organizations are growing, stretching the limits of their existing installations. (For example, NT limits account domains to somewhere between 30,000 and 40,000 accounts.) They tend to contrive solutions that support other business requirements, such as using contiguous namespaces. Large organizations may also reduce the number of servers on their networks, and this has a negative effect on manageability.
- **Integration**—New applications may require introducing new infrastructure.

The goals of migrating to Win2K generally focus on the concepts shown in Table 2.1, as derived from Microsoft Consulting Services (MCS) common practices:

This Goal	Has These Implications for the Migration Process
Security	The project must have a minimal impact on security policy. Perform a risk assessment to identify any potential threats and take the appropriate countermeasures.
Minimum disruption to the production environment	If possible, maintain users' familiar environment during and after. Most important, maintain users' access to data, resources, and applications during and after.
No degradation of system performance	Maintain or improve expected performance.

Minimum administrative overhead	User accounts should be seamlessly migrated. If possible, users should be able to retain their passwords. Administrators should only visit client computers a minimum number of times. New permissions for resources should require minimal setup.
Maximize “quick wins”	The enterprise should obtain access to key features of the new platform as soon as possible.

Table 2.1: Goals for and implications of migrating to Win2K.

Defining Migration Tasks

In this section, I’ll discuss another key aspect of planning a migration to Win2K—defining migration tasks:

- Training users and administrators
- Using TCP/IP
- Migrating servers versus workstations first
- Auditing network infrastructure
- Auditing your domain structure
- Auditing applications.

Training Users and Administrators

While reading a book such as this is one part of your work, I recommend that you take some of the Microsoft-specific courses in Win2K technology. They’ll help you get hands-on experience in using the product and its solutions. Furthermore, although many of you may be familiar with directory technology from encounters with other directory-enabled products, Microsoft introduces its own flavor to the mix, so consider at least spending some time learning about namespace design in AD.

In fact, AD is potentially one of the first things that bring Win2K deployments to a standstill. Because it’s new to most people and has many potential pitfalls if it isn’t implemented properly, you really need some advanced training to understand how it works and how it will benefit your organization. So be sure to receive this training before or soon after your migration project starts.

I consider Win2K training essential to prepare you for a successful deployment of Win2K. Win2K is also a completely new environment and toolset for your system administrators, so be sure to arrange Win2K training for them as well as for end users, support staff, and networking staff. Taking the Microsoft certification program is one way to achieve technical training (although it should only supplement your Win2K training). You can find information on Win2K training, including the Microsoft Certified Systems Engineer (MCSE) track, at <http://www.microsoft.com/trainingandservices/default.asp?PageID=mcp>.

Training end users is something that your end user representatives should help with. Receiving training will make end users aware of when the migration will take place and the impact it'll have on them. It'll also ensure that they have the correct level of experience with the product, be it from seminars or one-on-one training.

Inexperienced administrators and implementers of any system are an issue faced by any migration, and in the case of Win2K, the shortage of well-trained IT personnel simply exacerbates the problem. Although the situation is improving as more companies try to migrate, inexperience is a major factor in any unsuccessful Win2K deployment.

To mitigate this problem, consider reinforcing your migration team with experienced implementers from systems integrator organizations, whether from large organizations such as Microsoft, Compaq, or IBM Consulting, or smaller boutique firms. It's essential that you not only gain from their experience, but before attaining their services, validate their experience with other clients to ensure that they're not simply learning from you. Participating in the migration with experienced implementers also gives your staff a chance to achieve training by doing.

Using TCP/IP

If you're not already using Transmission Control Protocol/Internet Protocol (TCP/IP) as your network protocol, now is the time to switch. To implement Win2K, your network must be running TCP/IP, as opposed to NT, because it allows you to operate using NetWare LINK (NWLink), NetBIOS Extended User Interface (NetBEUI), or TCP/IP. The primary difference between NetBEUI and TCP/IP is that TCP/IP is routable, so it's much more efficient in large networks, as demonstrated by the Internet.

This is another opportunity to minimize the costs of managing your network by consolidating it onto a single supported protocol. This won't be possible in all organizations, such as those that still run large mainframe services requiring Data Link Control (DLC). However, all UNIX installations support TCP/IP, as does Apple and Novell.

Migrating Servers versus Workstations First

Deciding whether to migrate your servers or workstations first will depend on the following:

- The impact on your user population
- Application compatibility issues
- User desires (for example, Win2K performs much more efficiently on laptops than NT)

To determine the correct choice, you'll need to complete the audits described in "Auditing Your Domain Structure" later in this chapter. However you decide to proceed, I recommend that you perform clean installations on both server and workstations.

Table 2.2 presents Microsoft's estimates of the time it takes to install Win2K on workstations, depending on which method you choose.

	Traditional (Manual)	Disk Duplication	In-Place Upgrade	Unattended Installation	Remote Installation
1. Prepare Target Computer					
Backup User Data	30	30	30	30	30
Format Hard Disk	15	0	0	15	0
2. Install OS	45	10	45	45	60
3. Install Device Drivers	30		30		
4. Install Applications	45		45		
5. Restore Data	45	45	45	45	45
7. Login and Test Functionality	30	15	30	15	15
Hours per computer	4	1.83	3.75	2.5	3.5
Time Saved vs. Manual Method		54%	6%	38%	13%

Table 2.2: Target times for installing Win2K.
 (Source: Microsoft's "Deployment Cost Savings with MS Windows 2000 Professional" white paper)

For more details on how to implement the deployment options described above, see Chapter 3 of *The Definitive Guide to Windows 2000 Administration* by Sean Daily and Darren Mar-Elia (Realtimepublishers.com), a free eBook available at <http://www.realtimepublishers.com>.


Considering Security Identifier Duplication Issues

If you plan to use any disk-image-duplication methods when you deploy Win2K—for example, Ghost (Symantec) or Drive Image (PowerQuest)—you need to consider potential security identifier (SID) duplication issues. Each computer on a Windows network is given a machine-specific SID, which is also used to create unique SIDs for local accounts. Imaging solutions harvest a duplicate image of a computer's disk after a SID has been assigned, and this creates a potential security conflict. By using the duplicate image to create a new machine, this target machine, and any others, will be imprinted with the same SID as the original reference machine. Unfortunately, when an account is created on a machine, the SID it's given is based on the machine SID. As a result, if two machines were to have the same SID, any specific account names created on them would have the same SID.

Microsoft offers its own take on duplication issues and cloning utilities at: <http://www.microsoft.com/TechNet/winnt/ntwrkstn/technote/cloning.asp>.

Microsoft also offers a solution for these issues, called Windows 2000 System Preparation Tool (SysPrep). SysPrep is a tool that works with these cloning tools and allows an administrator to write the image, install all required applications, then force the clone system to revert to a virgin state as if it had never had a SID assigned at all. SysPrep strips an installed system of its SID, computer name, and other unique configuration data. When the system is restarted after SysPrep

has been run, the operating system (OS) forces itself to re-initiate the SID generation, creating a new and unique SID. Thus, if you run SysPrep, then clone the system, it'll automatically generate a new SID when it starts up.

 The source and target computers being used for SysPrep-based disk-image duplication must have identical Hardware Abstraction Layers (HALs), Advanced Configuration and Power Interface (ACPI) support, and mass-storage-controller devices. Essentially, these need to be extremely similar hardware and computers.

Microsoft prefers to support the SysPrep method, as opposed to the clone solutions alone. If you call Microsoft for support on a cloned system that wasn't created using SysPrep, it's unlikely that you'll receive assistance.

 SysPrep 1.1 can be freely downloaded from Microsoft's Web site at <http://www.microsoft.com/windows2000/downloads/deployment/sysprep>, and it's also included with the Windows 2000 Resource Kit.

Auditing Your Network Infrastructure

Your network environment is critical to the success of your Win2K deployment. This is because Win2K relies on specific network services. In addition, you need to understand the performance characteristics of the network to ensure that you configure Win2K to perform optimally in your environment.

I'll deal with the audit and documentation of your NT environment later in this chapter (see "Auditing Your Domain Structure"), but first let's look at the actual layout of your low-level networking.

Taking Inventory of the Network

The first step is to determine your network topology, or how it's laid out. You may be lucky enough to have a networking department that already manages this type of activity, and it can provide you with the appropriate physical and logical topological maps. Ensure that you can determine how your various locations are connected (if at all) and how each location is wired, from central resources to end user workstations. The critical factors are the performance characteristics of the network connections—they'll determine where you'll locate your Win2K domain controllers and what services they provide (for example, security, lookup, and networking).

Figure 2.2 is a sample network diagram showing what information is important for a successful migration to Win2K. Important aspects are:

- Performance and type of connectivity
- Hosts
- Subnets
- DNS locations and types

- Client counts per subnet.

Having a network map will also help with your overall audits by identifying the locations and network infrastructure you'll deploy on. Plan to expand the network map with the information that you'll gather later on in the host and domain structure audits.

If possible, identify the current traffic on your networks; this will help identify any potential saturation points caused by adding Win2K. Consider that by adding Win2K to a network that already has Exchange 5.5 installed, you'll increase the traffic for directory replication. However, migrating to E2K will alleviate some of that traffic.

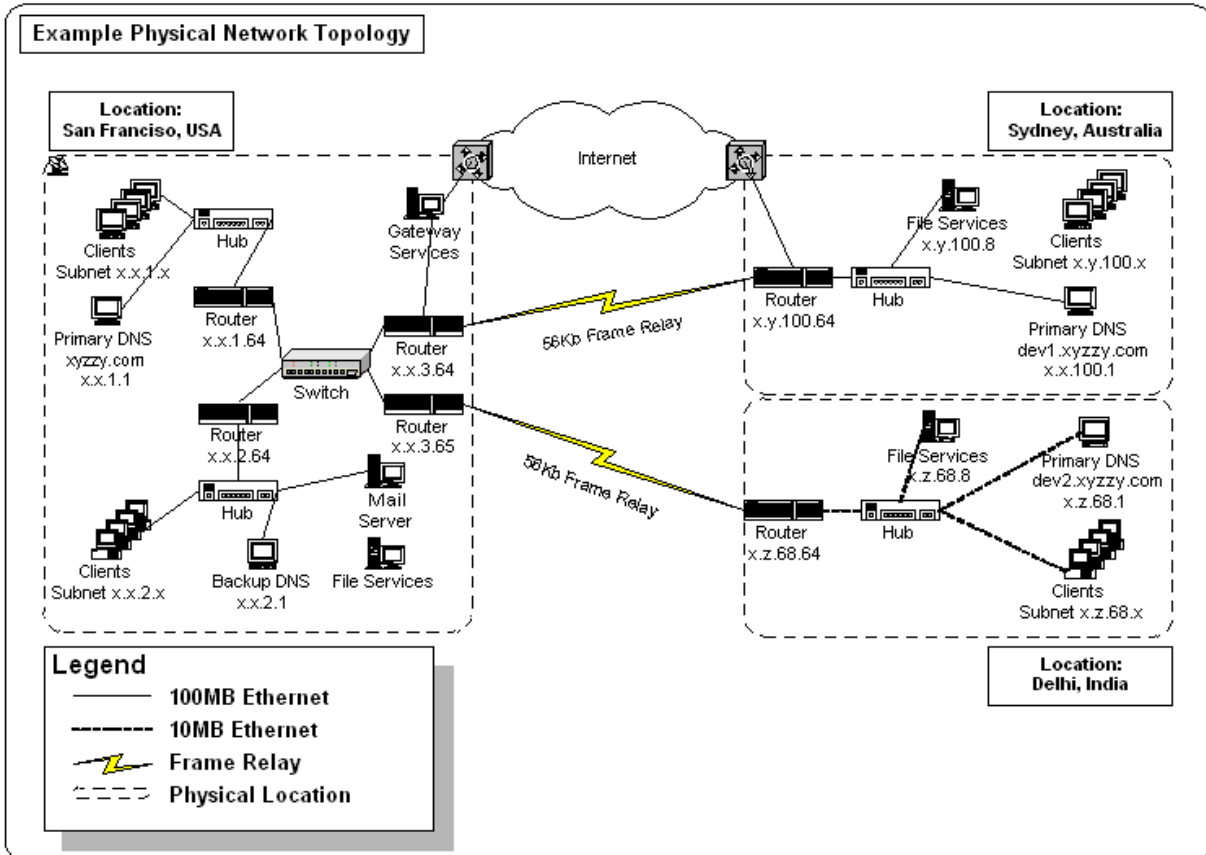


Figure 2.2: A sample network diagram.

Managing the Network

Most Microsoft deployments don't manage a network and its systems very well. If you haven't already implemented a network-management solution, you need to supplement your network management by adding some third-party tools. While Microsoft's own Systems Management Server (SMS) provides basic audit capabilities, including software management and remote control, it's a far cry from what you need to ensure that you're managing your network efficiently.


Network analysis tools like those offered by Tivoli, Hewlett-Packard (HP), and Computer Associates (CA) offer many levels of network administration and management, but you need to

consider that Win2K applications are driven from all layers of the stack. As a result, review your needs for tools that understand layers other than just the network layer—Operations Manager (NetIQ Corporation), DM/Suite (FastLane Technologies), and others from BindView Corporation and BMC Software.

Not only will these applications help you migrate to Win2K, but they'll also provide valuable diagnostic information should any problems arise in the application layers down to the low-level network protocols. For example, say replication isn't occurring as expected among domain controllers. If a router was malfunctioning or improperly configured, network-management software would help you isolate the problem to that router by allowing you to check where information was flowing and where it was failing.

Reviewing Naming Conventions

As you go through the audit process, review your naming conventions for domains and hosts. Win2K now uses DNS names for these, and according to Request for Comments (RFC) 952, "DOD Internet Host Table Specification," and RFC 1123, "Requirements for Internet Hosts—Application and Support," you can use names that are drawn from the alphabet (A-Z), digits (0-9), and the minus sign (-). No distinction is made between uppercase and lowercase characters. Win2K also supports names from characters as defined in RFC 2044, "UTF-8, a transformation format of Unicode and ISO 10646."

 Although Win2K DNS supports RFC 2044, other DNS services may not, so I don't recommend making extensive use of this standard unless you think that your environment fully supports it. The only exception I'd note is the underscore character (_) character, which Microsoft uses extensively in naming its Win2K services in DNS, allowing them to be found over TCP/IP networks.

Keep in mind that DNS names can be longer than the standard 15 characters allowable in Network Basic Input/Output System (NetBIOS). DNS allows host names to be up to 24 characters long, as distinct from fully qualified domain names (FQDNs), which include the host name as well as the full domain. In any case, it's still good practice to try and keep names short because they're simpler to understand and manage.

As you audit, review whether you need to change the names you're using for host names.

Reviewing Tools and Utilities


Many tools exist to help you migrate to Win2K, and I'll discuss numerous tools throughout this book. For now, though, let's take a look at some specific examples and where to locate them.

Creating Network Diagrams

As you assess your network, you may find that your networking diagrams aren't all they're meant to be. I've seen cases where organizations had no real idea what networking equipment, or even applications, they had installed. It may help to obtain some form of automated network scanning service, although it can be quite expensive.

Applications that I've used successfully are Microsoft's Visio 2000 Enterprise Edition and NetViz 5.0 Professional. Both offer auto-discovery of networks—Layer 3 (IP network), Layer 2

(data link), and frame relay information—allowing you to identify network components and connectivity. NetViz also supports interrogating your NT environment, allowing you to audit server and workstation configuration as well. Other similar solutions, offering differing levels of functionality, include LANDesk (Intel Corporation), Network Inventory Management (NetCracker Technology Corp.), WhatsUp Gold (Ipswitch Inc.), and SNMPc (Castle Rock Computing).

 Many networks have unmanaged devices and potentially badly managed bridge tables, which can affect the accuracy of the results these tools provide. Again, use any networking team you can to ensure that you use accurate information in your planning.

Some of these applications also support network simulation and traffic generation, which can help you design and test your new environment.


Microsoft Migration Tools

Microsoft offers the following migration tools as part of its NT 4.0 and Win2K resource kits: Netdom, ClonePrincipal, SIDWalker, MoveTree, Xcacl, and SubinACL. For example, Xcacl.exe is a program that allows you to set NT file system (NTFS) permissions on both files and folders. It also gives you the option to view permissions on files and folders for auditing purposes. SubinACL.exe allows you to obtain security information on files, Registry keys, and services and transfer this information from user to user, from group to group, and from domain to domain.

These are just two of the tools available. You may want to review the Win2K Resource Kit for others.

Microsoft Migration Tools

For a list of more migration-related tools, see the Microsoft article “Free Windows 2000 Resource Kit Tools for Administrative Tasks” at <http://support.microsoft.com/support/kb/articles/Q274/3/05.ASP>, as well as Microsoft’s hosted Web site for tool downloads at <http://www.microsoft.com/windows2000/library/resources/reskit/tools/default.asp>.

 Although resource kit tools are useful in some situations, they aren’t perfect. Specifically, they’re stand-alone, non-integrated tools, Microsoft doesn’t officially support them, and they’re rarely localized. While these tools can save you in many situations, and I use them extensively, they can sometimes be complicated to use. They come with only basic documentation, and you often need to use several components (executables) to perform a single migration-related task.

One of the most widely known migration utilities is the Active Directory Migration Tool (ADMT), which Microsoft licensed from development out of Mission Critical software, one of the early rapid-development partners in Win2K. (NetIQ acquired Mission Critical in 2000 and offers an advanced version of the solution.) ADMT assists in migrating from NT domains to pure Win2K domains. You can download it from <http://www.microsoft.com/windows2000/downloads/deployment/admt/default.asp> and install it, as shown in Figure 2.3.

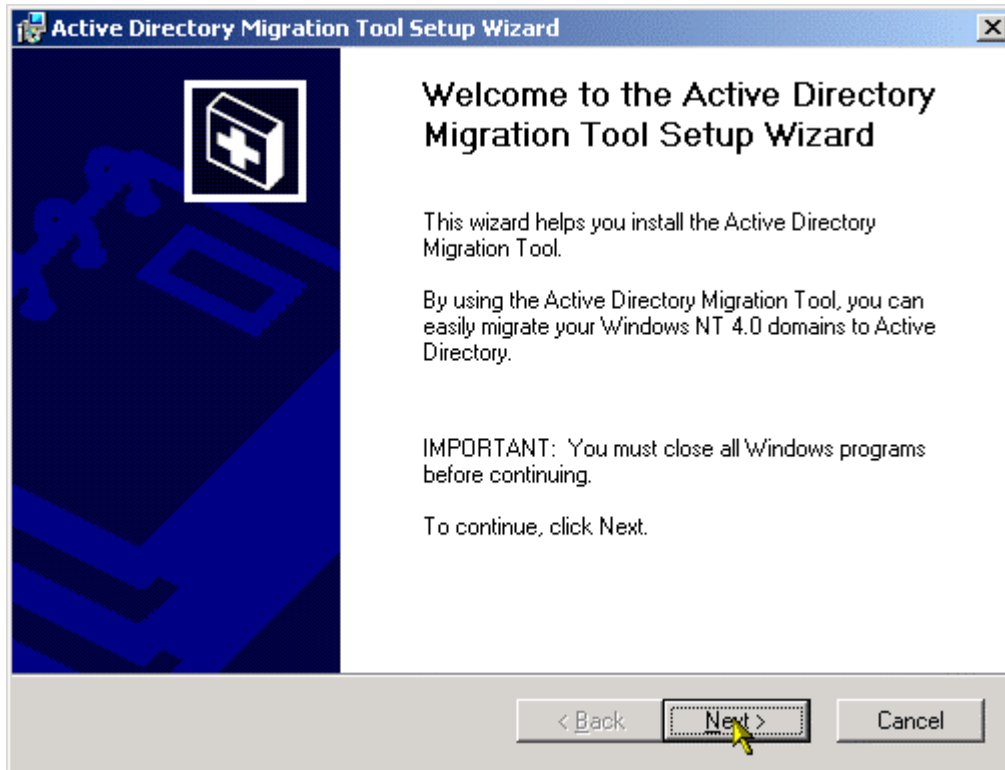


Figure 2.3: The ADMT installation screen.

ADMT allows you to migrate user and machine accounts, security principals, and groups from a source NT or Win2K domain to a target Win2K native domain. The tool offers wizards, including User Migration, Computer Migration, Group Migration, Service Account Migration, Trust Migration, and Reporting, to simplify various parts of the migration process. Interestingly, the tool offers a trial mode, which logs all activity that will take place, but not actually perform these tasks. This allows you to test your migration plan and see the results but actually perform the complete migration later. A review of the log files and reports generated by the wizards help you identify and troubleshoot any potential problems before performing the actual migration.


ADMT is a rather comprehensive tool, and it ensures that you have a swift and clean migration to Win2K. However despite being offered by Microsoft, it's task-specific, and it won't help you in the long run if you plan to manage multiple NT domains for some time after you migrate to Win2K.

Table 2.3 lists Microsoft's migration tools, including ADMT.

Tool	Description
Active Directory Migration Tool (ADMT)	A Microsoft Management Console (MMC) snap-in that provides wizards to automate migration tasks such as moving users, groups, and computers (between or within forests), migrating trusts, and performing security translation.
ClonePrincipal	Used to clone user and group accounts from an NT 4.0 or Win2K source domain in a separate forest to a native-mode Win2K target domain.

Netdom	Often used in the same scenarios as ClonePrincipal to move computer accounts from an NT 4.0 or Win2K source domain to a Win2K target domain. Netdom is also used to re-create trusts, typically in migration scenarios between the target domains and any domains trusted by or trusting the source domain.
MoveTree	Used to move users, groups, and Organizational Units (OUs) among Win2K domains in the same forest.

Table 2.3: Microsoft Win2K migration tools.

 There are several major differences between ClonePrincipal and MoveTree (we'll lump ADMT and ClonePrincipal together here, since they have the same characteristics and limitations):

ClonePrincipal and ADMT are nondestructive, meaning they leave the original account intact, allowing users to roll back in the event of a problem. MoveTree is a one-way function with no contingency options.

ClonePrincipal and ADMT only operate among domains in separate forests (interforest), and MoveTree can be used only among domains in the same forest (intraforest).

MoveTree migrates the user's password. ClonePrincipal and ADMT require the administrator to pre-set a new initial password.

Third-Party Migration Tools

To mitigate the issues with Microsoft migration tools, you'd be wise to review third-party tools, which take away a lot of the complexity and risk while typically improving speed and efficiency. Migration utilities can minimize the effort it takes to manage creating users accounts in a new domain and to update access rights on each server to include these new accounts. Whether you're trying to consolidate domains or migrate to a new Win2K infrastructure, having a tool that makes your life easier can be vital.

Examples of such tools are described in Table 2.4.

Company	Product Set	Description
NetIQ Corporation	Domain Migration Administrator (DMA) Server Consolidator	DMA provides extensive automation features to enable you to migrate user accounts, passwords, groups, member servers, workstations, user rights, and many other components between NT and Win2K domains—while preserving access to existing resources and without disrupting end users.
FastLane Technologies	DM/Consolidator	Facilitates complete data-movement projects (one-time or ongoing), supporting data migration within partitions, across partitions, across machines, across domains, and across forests.
Aelita Software Corporation	Domain Migration Wizard Server Consolidation Wizard	Domain Migration Wizard allows you to restructure domains and migrate to Win2K and E2K. Key words are fast, complete, WAN-optimized, team workflow-enabled, zero system downtime, and zero impact on users. Server Consolidation Wizard is a server-consolidation and data-migration solution for enterprise NT, Win2K, and Novell NetWare/NDS networks.

BindView Corporation	bv-Admin for Windows 2000	Allows organizations to reduce the cost of system administration by delivering a coordinated approach to system administration and directory management. bv-Admin simplifies the administration of user information across multiple directories as well as Windows-based servers, such as Microsoft Exchange. In addition, bv-Admin enables smooth migration to Win2K from NT and Novell systems.
BMC Software	PATROL	Although not specific to Win2K migration, provides automated monitoring and management to increase the availability of Windows NT Server and Windows 2000 Server. In fact, BMC offers this option for all BackOffice servers, including Exchange.

Table 2.4: Win2K migration tools from third-party application vendors.

Most of these vendors also offer tools to help you manage Exchange and migrate to E2K. These will be discussed in detail in Part 2 of this book.

If you decide to use any of these tools, keep in mind their level of integration with existing Microsoft tools, like Active Directory Connector (ADC), used to synchronize AD with Exchange 5.5 installations. Any conflicts in this area could cause disruptive and unanticipated results in your migration. Chapter 4 will discuss migration best practices and deployment tools and techniques in much greater depth.

Auditing Your Domain Structure

As you plan your migration to Win2K, you need to consider many aspects of your existing network. Essentially, you need to audit your network environment to identify key information about the following components:

- Master account domains
- Resource account domains
- Applications
- DNS services
- Trust relationships
- Current trust setups.

Many of these components may be well documented already. If not, the people on your core migration team should be able to obtain the necessary documentation.

The following sections contain sample audit-information tables that show you the type of information you need.

Master Account Domains

Table 2.5 is an example of the information you need about master account domains. This information will also be helpful later when you evaluate the impact of the migration on users and services.

Domain Name	PDC	Service Area	Number of Accounts	Number of Users	Number of Workstations	Number of Member Servers
MASTER1	MASTERPDC	San Francisco	5000		4000	
...

Table 2.5: Sample audit table of master domains.

Workstations also have “accounts” in these domains, and identifying them is part of the audit. This information will be important if you decide to integrate your domains before performing your migration.

Furthermore, one of the most complicated aspects of any migration is reassigning share, file, and print access control lists (ACLs). It’s essential to identify important share, file, and print points in your domains so that you can plan to migrate them and deal with permission issues, especially in cases where they’re essential to business productivity. Earlier in this chapter, I discussed several tools and toolkits that can help you in this endeavor (see “Reviewing Tools and Utilities”).

In each domain, you identify each service host and its role; you also need to ensure that it has enough capacity to deal with the migration. (For example, AD requires much more space than the simpler Security Account Manager (SAM) database. I’ve already discussed tools that help you map out your network (see “Managing the Network”). Some of these tools also help you audit each key domain controller and application server for the information listed in Table 2.6.

Data Point	Sample Data	Notes
Server name	Host1	
DNS name	Appserver1.company.com	
Role(s)	Application Server DHCP Print Server	Shouldn’t be responsible for DHCP.
Location	Building 2, Floor 20, Cube 15, Under Steve’s desk ☺	
OS	NT 4.0 SP6a	
Processor (count)	2 x 500 PIII	
Memory (RAM)	128MB	
Physical drive configuration	Drive 1: 10GB Drive 2: 40GB	

Logical drive configuration	C: System 4MB D: Applications 6MB E: Data 40GB	No fault tolerance for system or user data.
Supported network protocols	TCP/IP NetBEUI	

Table 2.6: Sample audit table of NT servers.

Resource Account Domains

Resource domains usually serve many purposes: they can house e-mail services and provide access to files, printing, applications, and development-specific applications. Table 2.7 provides a sample audit table of resource account domains.

Domain Name	Area Serving	Number of Users (If Any)	Number of Accounts	Number of Workstations	Number of Servers
RESOURCE1	All areas	Service: 25 Admin: 5	5000	800	28
DEVDOM	Development	75	100	25	30
...

Table 2.7: Sample audit table of resource domains.

Auditing Applications

Identifying domains is only the first part of auditing your domain structure. You also need to identify the applications that need the domains, whether for security (for example, authentication/authorization) or simply housing the server(s) on which they run. Auditing your applications allows you to identify the impact that the migration will have on them as well as which applications will need to be certified to run in a Win2K environment. You may find, for example, that applications run across platforms. (A Human Resources application may have Web services running on NT, while the actual RDBMS is running on a UNIX box.)

Table 2.8 shows a sample audit of these applications, along with important notes and observations. Your audit should include any warning flags, essential applications and services that are in use by users, and the amount of time they can be taken out of service. Consider carefully those applications that support the core business. For example, an organization might perform a lot of work at the end of a quarter to report to the financial markets. Alternatively, other companies may be service-oriented, allowing maintenance windows only during specific times of day or at the end of the week.

Application Name	Host Domain	Number of Users (If Any)	Number of Servers	Win2K Compliant	Notes
SQL 7.0	MASTER1	30	4	Yes	
SQL 6.5	RESOURCE1	10	2	No	SP4


Exchange 5.5	RESOURCE1	5,000	3	Yes	SP3
Exchange 5.0	RESOURCE1	50	1	?	Need to upgrade to 5.5 SP3 immediately to support Win2K migration.
Fax Connector	RESOURCE1	50	1	?	Check compatibility with Win2K. Essential usage each weeknight 5 p.m. –10 p.m. PST to support delivery of daily customer news.
PeopleSoft 7.0 HRMS	HRDOM	35	3	No	Only PeopleSoft 8 is fully Win2K-certified; we'll need to upgrade.
Enterprise Directory	NA	All	2	No	Suggest metadirectory review to ensure data is current with AD.
Sales Force Automation (SFA) tool	SALES	1500	3	No	Application needs to be available 6 a.m.–10 p.m. weekdays; reports are run 6 a.m.–9 a.m. on Saturdays.
...

Table 2.8: Sample audit table of current applications.

Application-Compatibility Issues

Once you've identified essential applications, you need to consider application compatibility. Applications obviously need to work together, and upgrading them, as well as any necessary hardware, will increase project costs.

Because application compatibility can make or break an upgrade, or derail it completely, you need to understand the implications for even a single desktop component. Microsoft sponsors a Certified for Windows program; you can find information on it and a list of officially certified applications at <http://msdn.microsoft.com/certification/default.asp>.

 Microsoft also certifies hardware to work with Win2K in its Logo Program for Hardware. You can find information on this program and a list of certified systems at the Windows Hardware Quality Lab (WHQL) at <http://www.microsoft.com/hwdev/winlogo/>. This program is also an important part of the Hardware Compatibility List (HCL) provided in Chapter 1.

You need to plan the migration carefully for each application. While you'll decide to upgrade certain application, you may choose to replace or retire others. Things that may impact your migration decisions are primarily platform compatibility and platform reliance.

Platform Compatibility

You may remember an issue that arose when you first tried to implement earlier versions of Windows NT. Despite the work Microsoft put into the product, a shift to 32-bit architecture

occurred that caused a number of applications to stop working, especially those that relied on direct access to hardware. This seems to be less of a problem with Win2K because most vendors are either using the access models and APIs supported by Microsoft or working much closer with Microsoft to certify their software and device drivers.

In fact, Microsoft has decided to completely rewrite the printer driver subsystem in Win2K and manage creating drivers itself. This creates a more stable system, although some argue that new drivers are slower to market. I haven't seen this at all, and personally, I support a more stable system. Despite these efforts, you should test all business critical applications extensively for platform compatibility, as well as your company's ability to support these applications on the new platform.

You may find that certain applications won't work at all with Win2K; others may require specific levels of update in order to work in a Win2K environment. For example, Microsoft's own Systems Management Server (SMS) can operate with Win2K only when you run v2.0 with SP2. Even then, although SMS runs with Windows 2000 Server and can recognize Win2K Professional clients and share data with AD, it can't command changes based on AD.

Platform Reliance

An example of platform reliance is Windows Internet Name Service (WINS). Although you can run Exchange 5.5 in a Win2K environment, it requires WINS in order to function. Thus, WINS needs to be operational until Exchange v5.5 is no longer installed on the network (that is, you've upgraded to E2K) and NT servers have been decommissioned. I'll discuss Exchange specifically, and in much more detail, in Part 2 of this book.

Table 2.9 provides a list of Microsoft applications and their compatibility with Win2K.

Application	Compatibility with Windows 2000
BackOffice 4.5	Use with the BackOffice Readiness Kit for Win2K.
Exchange 5.5	Requires SP2.
SQL Server 7.0	Fully compatible.
SQL Server 6.5	Requires SP5a.
Proxy 2.0	Requires Update Wizard for Service Pack 1.
SMS 2.0	Requires SP2.
Site Server 3.0	Requires SP3.
SNA Server 4.0	Requires SP3.
IIS	IIS 5.0 is a core part of the OS install and will take over from any previous installed IIS versions. This is done without any changes to settings of existing installations.


Table 2.9: Compatibility of Microsoft BackOffice applications with Win2K.

Application compatibility also affects client-side applications. You must evaluate the client side of your application services (fax, SFA, HRM, and so on) just as you do the server side. Table 2.10 shows the compatibility of some Microsoft client applications with Win2K.

Application	Compatibility with Windows 2000
Internet Explorer	Versions 3.x and later will run on Win2K. However, unless mission-critical solutions are written for a specific version, you probably don't want to run anything earlier than IE 5. The major reason is security. You should regularly check and download the security and application patches available or visit the Microsoft Windows Update site at http://windowsupdate.microsoft.com to ensure that you have what Microsoft considers the most up-to-date system patches and application software.
Office 2000	Yes.
Office 97	Yes.
Proxy Client	Yes.

Table 2.10: Compatibility of client applications with Win2K.

As I noted in Chapter 1, Microsoft has released BackOffice 2000. It includes SQL Server 2000, Internet Security and Acceleration Server (ISA, which replaces Proxy Server), Host Integration Server (HIS, which replaces SNA Server), and Systems Management Server 2.0.

 Although Table 2.10 mentions the Windows Update site, some press reports have questioned how up-to-date this site actually is. It seems that security updates aren't immediately posted to this site, and to avoid being caught out, review the bulletins page at the Microsoft security site at <http://www.microsoft.com/security/> or sign up for automatic e-mail notification of security issues and patches at <http://www.microsoft.com/technet/security/notify.asp>.

Current Name Service Architecture

Microsoft has based the trees and forests of the Win2K infrastructure on the DNS namespace. As a result, DNS is at the core of any Win2K deployment, and you need to consider it a key part of any deployment plan. DNS also plays an important part in all AD activity and is therefore essential to the management aspects of Win2K.

A critical decision is whether you want Win2K to be responsible for the highest level of DNS in your organization. When reviewing your DNS infrastructure, you need to determine who owns the highest level of the DNS tree and where inside that infrastructure the Win2K DNS servers will fit. For example, if your organization already deploys DNS, it's likely to be managed from a UNIX derivative, most commonly Berkeley Internet Name Domain (BIND), but there are other commercial products out there as well.

Finally, you need to determine certain information about your current Dynamic Host Configuration Protocol (DHCP) and WINS servers. This is essential in planning the migration of these services and minimizing the amount of time the migration takes and disruption for users. Table 2.11 shows the type of information you need.

Service	Host	Role	OS	Notes
DNS	dns1.domain.com	DNS master	Solaris 2.8	Serves root domain: company.com
WINS	wins1.domain.com	WINS master	NT 4.0 SP6a	

DHCP	dhcp.domain.com	DHCP for SF	NT 4.0 SP5	Running on BDC—need to migrate off.
WINS	wins2.domain.com	WINS replica	NT 4.0 SP6a	Running on BDC—need to migrate off.
...

Table 2.11: Sample audit table of name services.

Later in this chapter, I'll discuss DNS, its relationships with DDNS, DHCP, and WINS, and their impact on your migration efforts, in more detail (see "Considering Name Services: DHCP, WINS, and DNS").

Trust Relationships

Domain configurations are different for almost every organization, but they're usually based on one of the Microsoft domain models: single domain, master domain, multimaster, and complete trust. Understanding trust relationships will help you determine which domain model you have in place and whether to restructure your domains before you migrate. It will also help you ensure that the existing network security is maintained when you begin your migration.

Win2K supports two-way Kerberos trusts. It also allows you to flatten and consolidate domains. (I discussed this option in "Technical Goals" earlier in this chapter.) If your environment has a complex set of trust relationships, you may want to consolidate domains: fewer master and resource domains mean fewer trusts. This isn't a problem for most mid-size and smaller organizations, however.

Designing a Migration Strategy

Now that you've completed your audits, it's time to consider your migration strategy.

Upgrading versus Restructuring

You have two choices for designing your migration to Win2K. You can either upgrade existing servers and workstations in place, or you can create a pristine, new infrastructure and migrate your client and server applications as required. If you choose the latter option, you can restructure existing domains before upgrading them, or you can upgrade first, then restructure. These options require different timelines and different toolsets.

Whichever method you choose, you'll likely want to minimize the amount of new hardware you need to purchase. Understanding your network and domain infrastructure is the key to determining the most appropriate and cost-effective migration strategy for you. This includes the physical layout of the network, its performance characteristics, the physical location of your users, and political and local realities.

Upgrading

The upgrade method is also known as *in-place migration*. This is because there is no specific migration from one domain to another, and no new user or machine accounts are created. The key characteristics of an upgrade are shown in Table 2.12.


Feature	Description
Cost	Is less expensive than restructuring. There may be no need to undertake major capital purchases to support new environments.
Complexity	Provides the fastest and easiest migration path from the current structure to the desired state. It's non-destructive and coexists with the down-level domains.
Impact on existing structure and status quo	Maintains the current infrastructure and has no impact on the current workings and systems. It provides the same level of functionality among all the other systems.

Table 2.12: Key characteristics of the upgrade method.


The upgrade approach essentially requires a simple network and domain environment. You upgrade the primary domain controller (PDC) of an NT domain first, then upgrade any remaining backup domain controllers (BDCs) until you can switch the domain to native mode. You retain the existing domain and network structure and add the Win2K services you want—AD, Kerberos, Group Policy, and so on—to it.


The upgrade option carries inherent risks, but your budget may require you to use it nonetheless.

Because the upgrade method requires you to upgrade the PDC, and it's the key component of your NT infrastructure, there are few ways to back out. The solution is to install a new BDC into the domain, ensure that it's synchronized with the PDC, take it offline, then upgrade the PDC. The intent is to ensure that you have a reliable replica of the domain should anything fail when you upgrade. This is similar to the Exchange recovery solution, which allows organizations to bring any Exchange server back to life to salvage the data in users' mailboxes.

 Be aware that an NT backup domain controller (BDC) taken off of the network as a fallback measure during a migration has a limited shelf life. Once you begin your migration, you'll modify the SAM database on your remaining BDCs, and your PDC will be running AD. In the past, Microsoft has suggested that this situation could last for seven days, but I suggest no more than a day or two.

This is also a good time to make sure that you've created an Emergency Repair Disk (ERD) for your PDC and BDCs. This disk will help restore your configuration to a known good state. Use the RDISK utility to create and update your computer's ERD.

 There may be times when an ERD fails to be created, potentially because your domain and server information takes up too much space on the disk. In this case, you can either create a compressed version of the ERD or manually back up the Registry and related files. For more information on this and on the potential issues raised by these options, check out <http://www.microsoft.com/technet/tips/0500bob.asp>.

 The ERD can save you if your servers become corrupted, but it can also turn into a security liability if it falls into the wrong hands. Keep any ERD secure—it contains critical security and Registry information and other system files to help recover or repair a corrupted Windows installation.

These suggestions don't define a true disaster recovery plan, and should you not already have one in place, you really must invest the time to do so. At a minimum, review what would happen if your PDC failed and your domain became corrupt.

Table 2.13 examines at the advantages and disadvantages of the upgrade method.

Advantages	Disadvantages
Server-side migration—has little impact on enterprise client nodes.	One domain at a time; an all-or-nothing approach.
No need to update resource ACLs or restamp SIDs—saves time and energy.	No way to automatically generate the AD hierarchy until after the upgrade is complete.
Existing domain structure remains intact—makes it possible to roll back.	No way to synchronize Exchange and NT.
No need to move computers and re-connect to new hardware.	No audit log.
	No reconfiguration—need to clean up existing domains; must accept the namespace that is created.
	Lack of test-migration support.
	Lack of collision handling—cloned domains aren't supported.

Table 2.13: The advantages and disadvantages of the upgrade method.

Restructuring

Restructuring creates a clean and new environment to which you can migrate users and machines in a phased manner. The key characteristics of restructuring are shown in Table 2.14

Feature	Description
Design	The design is clean and uses an ideal forest model.
Consolidation	Domains can be consolidated or collapsed into one or more larger domains.
Flexibility	Users and computers can be moved among domains to be in the "right" places.

Table 2.14: Key characteristics of the restructure method.

Restructuring, while it's more complex and more costly, is safer than upgrading. It requires more hardware, but it's less risky and makes it easier to back out. Furthermore, creating a clean Win2K environment ensures that no legacy issues affect the use of the domain, and users, applications, and computers can move to the new environment as they're ready.

Hardware Requirements

As you plan your migration, you'll need to determine whether the hardware you have is suitable for deployment. This issue has two aspects.

- Compatibility—this is addressed largely by the HCL in Chapter 1
- Size, given the use of AD—To determine this, use Microsoft's ADSizer tool. Adsizer.exe, or Active Directory Sizer, estimates the hardware required for deploying AD in an organization. Figure 2.4 is an example of the output from ADSizer based on the metrics shown. Keep in mind that these estimates generally allow for an optimal situation; use them only as a guide.

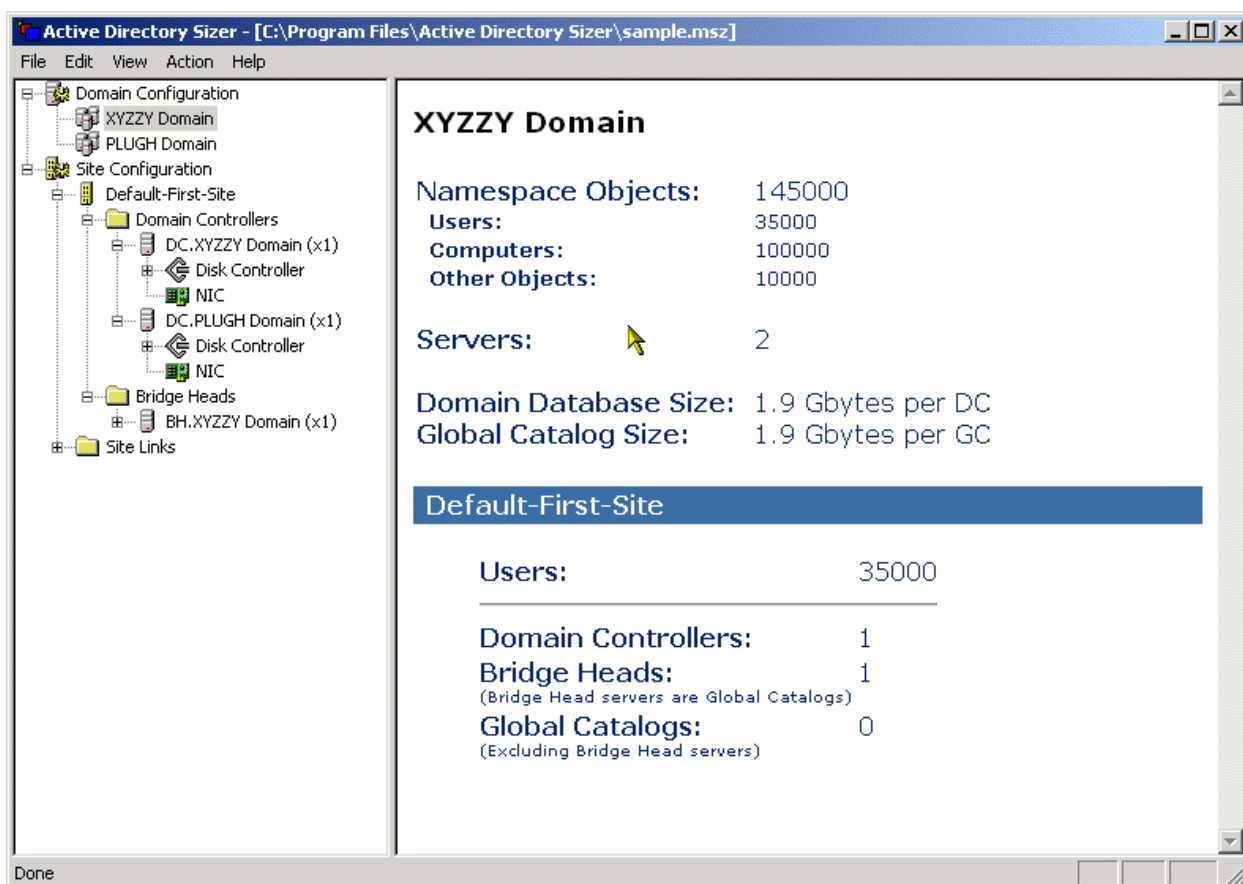


Figure 2.4: The ADSizer result screen.

In this section, you learned the advantages and disadvantages of each migration option. In some cases, upgrading may meet the stated goals, but this is usually in smaller environments. In larger environments, the idea of a complete in-place upgrade without extensive back-out planning causes a lot of trepidation. Both options can be very costly, but upgrading is more prone to error.

The recommended way to play safe is to perform the restructuring option if you can afford it. In Chapter 3 of this book, I'll discuss in detail how you use both migration methods.

Switching from Mixed Mode to Native Mode

When you begin to implement a Win2K infrastructure, one of two scenarios can take place. In fact, given the number of NT legacy domains out there, you'll likely need to deal with both regardless of whether you upgrade or restructure.

If you maintain an existing NT domain, then introduce Win2K servers into the environment, those Win2K servers must operate in what is called *mixed mode*. This allows them to interact with the NT domain by loading and supporting additional services for Windows NT LAN Manager (NTLM) authentication, trust management, and network interoperability. If you plan to upgrade in place, mixed mode is very important because it allows Win2K servers to interoperate with NT domains and share resources and security models.

If you create a "pure" Win2K domain infrastructure, this is called running in *native mode*. The advantages of native mode are that you can ensure that you're only managing services relevant to Win2K, your domain isn't susceptible to the existing security issues that apply to an NT domain, and finally, you can make full use of the new Win2K functionality, such as Universal Security groups and nesting. (I'll discuss these features in later chapters.)

When all your domain controllers are running Win2K, you can move to native mode, which enables pure Win2K functionality using Win2K domain and security services. The conversion to native mode poses a number of challenges and potential pitfalls depending on when you decide to do it. To decide when you'll make the move, you need to understand a little more about the differences between these two options.

About Mixed Mode

The premise of mixed mode is to allow a Win2K domain to assign a Win2K domain controller to perform the role of PDC for the NT BDCs in the network environment. Only a single Win2K domain controller can emulate a PDC, and it's by default the first domain controller installed into a Win2K domain. A PDC emulator does the following:

- Manages replication of account information to BDCs
- Supports account management such as updates, especially password changes
- Acts as master browser for NT clients
- Provides the NTLM authentication service.

In addition, in mixed mode:

- The upgraded PDC still acts like an NT PDC and uses NETLOGON replication to replicate directory information to down-level (NT) BDCs as a flat SAM accounts database.
- As a result of this, new NT BDCs can still be added to the domain, and if no Win2K domain controllers are online, an NT BDC can be promoted to PDC.
- A limited subset of AD functionality is available. That is, the domain SAM is copied to AD when you upgrade the PDC, and the Win2K domain controllers replicate amongst themselves using AD replication.

- Because you have AD, you can create OUs, but down-level BDCs know nothing of them, and they can only be administered from Win2K machines using Win2K tools.
- Because you now have AD, you can exceed the recommendations for the number of users imposed by SAM (around 40,000 objects, which includes machine accounts). However, this isn't a particularly wise course of action because if you have a problem during migration, you might want to add NT BDCs, and they don't cope very well with a larger user population.
- You also have access to Kerberos authentication. If the domain is part of a Win2K forest, Win2K workstations and servers can use the Kerberos protocol in the domain and to other domains
- If you use Kerberos authentication, you can set up transitive trusts. For example, if you add your upgraded Domain A to a Win2K forest as a child of a native-mode Domain B, and Domain B trusts Domain C, you've created a trust path to Domain C.

Keep in mind too that NTLM clients cannot change their passwords if a PDC emulator (that is, an operations master) isn't available.

Win2K clients always attempt to locate Win2K domain controllers using DNS first, and if they're not successful, they use NTLM. This means that if your Win2K domain controller isn't available, a client can still authenticate. However, because NT doesn't support Group Policy, new access and group information, such as updated Win2K logon scripts, won't be available.


This is an example of an operations master, of which there are many. I discuss operations masters below (see "Operations Masters").

About Native Mode

Native mode is pure Win2K and doesn't allow for any NT domain controllers. This means that you can have NT clients or "member" servers in the domain, but you can no longer use NT security services.

This is the ultimate goal of a Win2K migration, whether you decide to move from mixed mode or go directly for native mode. Native mode allows you to expand the number of users beyond the limits imposed by NT SAM. Using Win2K AD allows for a much richer set of management functionality, including Universal Security groups, nested groups, and transitive trust relationships.

If an NT domain remains outside a Win2K native-mode domain, you'll need to establish an explicit trust between them to allow access to resources in the NT domain.

 While Win2K allows you to use NT-style security policies even in native mode to support NT clients in a native-mode domain, this creates potential conflicts. There is no synchronization between NT security policies and Group Policy Objects (GPOs), so you run the risk of creating policy conflicts and seriously compromising your security policy.

When you install a new Win2K domain controller, it's in mixed mode by default, and you can then switch it to native mode. Switching to native mode doesn't preclude introducing NT BDCs

later (you may need them, for example, to provide application compatibility). However, this is a little awkward because you need to first add another Win2K domain running mixed mode, install your BDCs into that new domain, then set up all of the trusts as they existed in NT.

Only when the domain controller is switched to native mode is it able to leverage the full capabilities of Win2K. Switching to native mode has the following implications:

- NETLOGON replication ceases, so no NT BDCs can operate in the domain or be added to it.
- Because you don't have to worry about down-level replication anymore, you can take full advantage of AD scalability.
- New types of Win2K security groups become available, such as universal and domain local groups.
- You can take advantage of a new Win2K feature, the ability to nest security groups.

Operations Masters

Another important detail to introduce at this stage is the concept of *operations masters*. We've already discussed one in the form of the PDC emulator. (See "About Mixed Mode" earlier in this chapter.)

Operations masters are essential services running on a Win2K server that create a role for that server in a Win2K environment. To ensure that critical services and operations are unimpeded, Win2K defines operations masters to handle specific tasks relating to changes in AD. These are also known as *Flexible Single-Master Operation (FSMO) roles*.

Each Win2K domain has five FSMO roles, each of which operates within specific boundaries. These roles are described in Table 2.15.

Role	Boundary	Role Description
Schema master	Per forest	This is the only domain controller on which you can update the schema.
Domain naming master	Per forest	Responsible for managing adding and removing domains and cross-references to domains in external directories (for example, Lightweight Directory Access Protocol, or LDAP, directories) from a forest. If this role isn't available, you can't perform these functions.
PDC emulator	One per domain	This is the only role that provides support for legacy NT systems. The PDC emulator is the preferred domain controller for processing password changes, replicating SAM updates to legacy NT BDCs, and acting as the domain master browser. In addition, the PDC emulator is the authoritative time source for all systems in a domain and the default server for editing Group Policy and processing changes to the DFS configuration. This role still exists when you move to native mode, however; it continues to function as the central reference for password updates in Win2K.

Infrastructure master	Per domain	Ensures that domain controllers update cross-domain group-to-user references in a timely manner. Win2K can perform this function without the infrastructure master, but without it, the process takes longer.
Relative identifier (RID) master	Per domain	A unique SID represents each security principal (that is, user, group, and computer) in AD. A security principal's SID consists of a RID and the domain's unique SID. The RID master allocates to each domain controller in a domain a pool of RIDs from which to create SIDs. When the number of available RIDs falls below a predetermined number (100 by default), the domain controller requests additional RIDs from the domain's RID master. If the RID master is unavailable and a domain controller exhausts its store of RIDs, the domain controller can't create additional security principals.


Table 2.15: Roles and responsibilities of the FSMO roles (operations masters).

Any Win2K domain controller can host one or more FSMO roles. An important aspect of roles is that they can be either transferred manually or seized. If any of the FSMO servers fails, another domain controller can seize a role to ensure that the Win2K environment remains functional. Alternately, FSMO roles can be managed by an administrator, who can manually transfer roles from one domain controller to another to optimize network operations. To transfer roles, you can use the command-prompt utility Ntdsutil or the AD Users and Computers MMC snap-in. (In the snap-in, right-click the domain you want to view, then select Operations Masters from the context menu).

You don't normally need to worry about operations masters. However, if an FSMO server fails, you need to know how your Win2K environment is deployed, which servers are performing FSMO roles, and which servers are acting as backups for which FSMOs. This is very important because FSMO roles can affect upgrading and migrating to Win2K. For example, FSMO roles manage updating the schema across Win2K when you install E2K.

Considering Name Services: DHCP, WINS, and DNS

By now, you should have assembled a support team, mapped out your DNS infrastructure, and defined your strategy. Other important aspects of planning the migration to Win2K are your WINS and DHCP services. I'll describe each at a high level, then define how your Win2K deployment needs to use them to be successful.

 For more in-depth coverage of DNS, see Chapter 4 of *The Definitive Guide to Windows 2000 Administration* by Sean Daily and Darren Mar-Elia (Realtimerepublishers.com), a free eBook that you'll find at <http://www.realtimerepublishers.com>.

Windows Internet Name Service (WINS) is an NT (or Win2K) network service that provides dynamic NetBIOS name-to-IP-address resolution services to network clients and NetBIOS over TCP/IP name registration. WINS is similar to DNS, but it's specific to Windows environments that support NetBIOS.

Dynamic Host Configuration Protocol (DHCP) is a protocol for assigning dynamic IP addresses on a TCP/IP network. DHCP also allows you to assign network settings such as default gateways, WINS servers, and DNS servers. While DHCP servers maintain a database of host names and IP addresses, DHCP doesn't respond to DNS queries. More unfortunate still, while older or previous DHCP can assign WINS and DNS servers to each request, DHCP servers don't update WINS or DNS servers either. That is up to the clients. Win2K DHCP, however, can update address (A) and pointer (PTR) resource records in DNS either by client request or dynamically, depending on your needs. This addresses the WINS model in DNS, allowing DHCP clients to automatically update DNS when they obtain a new address.

All of these services can pose challenges for your migration project, especially if you're running these services atop NT domain controllers. Although these services don't depend on NT domain services or security, they all rely heavily on TCP/IP addresses.

In Win2K, any DNS service controlling the domain must support RFC 2052, "A DNS RR for specifying the location of services (DNS SRV)." This is essential because Win2K clients perform network lookups or service location resource records (SRV RRs) on a DNS server to locate an AD server and related services such as Kerberos servers, domain controllers, and Global Catalog servers. These are fundamental services in a Win2K environment, providing security and locating resources across the network.

The Win2K DNS service support, Dynamic DNS (DDNS) allows hosts on the network to automatically register their host names and IP addresses. It does this using a standards solution like that provided by WINS, thereby essentially replacing WINS. Although DDNS isn't absolutely required to implement Win2K, the advantage of having it on the network and servicing the Win2K DNS namespace is that it minimizes the administrative burden of manually updating DNS with the same information. Thus, consider who will own the DNS namespace in your organization.

Many organizations that have used DNS for some time are running it on UNIX variants. They likely haven't upgraded to the most recent version of the Internet Software Consortium's (ISC's) BIND, which is 8.2.2, or to a new DNS version from their UNIX and DNS vendors that supports both SRV RRs and dynamic-update standards. You may face this challenge in your organization too.

Thus, your DNS team will need to choose among the following:

- Migrate to Win2K DNS services and make Win2K responsible for your organization's root domain and all Win2K domains
- Pursue a co-existence strategy—for example, making Win2K DNS responsible for all subdomains that contain Win2K clients and servers
- Remain on your existing DNS implementation and manually implement changes.

Let's take a look at these options and their implications.

Using a Contiguous or Disjointed DNS Namespace

In planning, or reviewing, your DNS infrastructure, you need to determine if you'll use a contiguous or disjointed namespace and, as discussed above, which DNS service will be responsible for which part of the namespace.

In a *contiguous namespace*, a root domain (such as company.com) and all child domains always contain the name of the parent domain in their names. For example, intranet.company.com is a contiguous child domain of company.com.

In a *disjointed namespace*, a child domain doesn't contain the name of a parent domain as part of its domain name. For example, directoryservice.com could be a child of company.com, but it seems distinct and disjointed from the company.com namespace.

The choice you make affects how LDAP searches take place. In a contiguous namespace, referrals are created automatically to support LDAP searches across the domain namespace. In a disjointed namespace, referrals aren't created, so searches are limited to the known namespace. However, you can set up referrals manually, and while this is a more complex operation, it's an option if you need it.

Using Win2K DNS Services

There are a lot of reasons to use Win2K DNS. The functionality of the service is expansive, and its close integration with AD makes it an easy choice in organizations that haven't already deployed another DNS solution.

Unlike NT environments, Win2K domains are named using the DNS namespace. As discussed in "Reviewing Naming Conventions" earlier in this chapter, you must identify a root domain suffix to use, and this may already be registered with a DNS registrar. If not, now is the time to decide what your root domain will be.

As a result, one of the most important concepts to understand before you implement your DNS design is your AD design; you can then support that namespace with an appropriate DNS namespace. Specifically, you should know what domains and subdomains will be used in the deployment and whether those domains are already managed by another DNS. You can then decide on which domain controllers you need to install the Win2K DNS service.

Microsoft makes installing the Win2K DNS service simple by offering the Active Directory Installation wizard. The wizard, along with installing AD, can also install and configure the DNS server and add the Win2K- and AD-specific locator records to the DNS service. You can also manually configure the DNS service using the Microsoft Management Console (MMC) DNS Console.

Win2K also supplies a command-prompt utility for managing the service, known as dnscmd.exe. It allows you to script actions against the service.

When you start the Active Directory Installation wizard and choose to create a new domain, the wizard finds the DNS server that is authoritative for the name of the new AD domain, then checks whether that server will accept dynamic updates. If the test is positive, the wizard doesn't install or configure a local DNS server. If the test fails, the wizard offers you the option to install DNS locally.

Using Win2K DNS servers allows you to do away with configuring clients to use WINS because Win2K DNS can perform a WINS query on behalf of client name-resolution requests. Another major advantage of implementing Win2K DNS is that if all computers run Win2K, there is no need to use WINS anymore. The caveat is that a number of older applications rely on WINS, and you need to address this in your audit of applications in use.

Microsoft offers an excellent DNS installation section on its Web site at <http://www.microsoft.com/technet/win2000/win2ksrv/reskit/tcpch06.asp>. However, I'll discuss the details of AD and DNS installations in Chapters 3 and 4.

Using Third-Party DNS Services

If you plan to implement or retain a third-party DNS service, you need to understand how Win2K uses the DNS namespace to structure its internal services, especially *Lightweight Directory Access Protocol (LDAP)*.

In LDAP, you can implement a domain name using a *distinguished name*. A distinguished name allows you to locate a specific object, such as a domain definition, in the directory tree. As an example, consider intranet.company.com. In LDAP, it can be represented as:

```
dc=intranet, dc=company, dc=com
```

In this example, *dc* represents individual domain components of the FQDN. As discussed in “Assembling a Migration Team” earlier in this chapter, you must have the right team on board (whether existing, hired, or recently trained). The example above shows the strong relationship between your DNS design and directory design; this should impact your review of team members as you progress through this stage of your migration.

In “Name Services: DHCP, WINS, and DNS” earlier in this chapter, I mentioned that for third-party DNS servers to work in your new Win2K environment, they must support dynamic updates and SRV RRs. This process is supported by BIND 8.1.2 or later, but you need to use BIND 8.2.1 or later to support incremental zone transfers.

Zone transfers copy zone information from a primary server to a secondary server, and this is normally a manual process. It isn't essential and in some cases is even discouraged for security reasons. For example, some nefarious person could set up a DNS service, request a zone transfer from a DNS master, then quite simply obtain a comprehensive list of all hosts in a domain for his or her own purposes.

I recommend that you use the *dcdiag_setup* tool available from Microsoft to check your environment and make sure that your third-party service supports all the required RFC and functional aspects. You can download *dcdiag_setup* from http://download.microsoft.com/download/win2000platform/Update/5.0.2195.2103/NT5/EN-US/dcdiag_setup.exe. There is a huge amount of information on this site, so check around with your vendors for compatibility information as well.

Considering the Impact on Exchange

It's unfortunately still common for organizations to use more than one messaging application. In some cases, they may use different applications, such as Lotus Notes, Microsoft Exchange, MS

Mail, or others. In other cases, they may use the same application but have it installed in different parts of the organization and with different namespaces. If you're faced with one of these situations, it's very similar to the problem of merging accounts from disparate NT domains. If you plan to migrate to E2K, you need to consider the impact of your Win2K namespace decisions on your E2K deployment.

The two specific issues you need to consider are:

- **Conflicting namespaces**—Where multiple installations have used the same “naming standards,” and should you try to combine the installations, you'll find collisions. For example, having two accounts registered as jsmith@company.com will cause a conflict if you try to combine these “namespaces.”

In this case, you need to have a procedure to resolve conflicts. Alternately, although this isn't usually well received, you can ask that all users accept new account names.

- **Non-aligned namespaces**—Where the naming conventions of one domain or application are different from the other.

This may be easier to deal with in Win2K migration. In this case, you can likely consolidate all of the names and allow users who want them to have new names.

You may also want to allow users to log in using their e-mail addresses. This seems a simple approach, and if your account namespace doesn't match your messaging namespace, you can use a Win2K feature of AD known as the user principal name (UPN). The UPN is a login name for a Win2K user based on the naming standard defined in RFC 822. Unlike a distinguished name, a UPN is shorter, easier to remember, and perhaps most importantly, is not tied to a specific hierarchy in the directory tree. As such, the domain structure can be changed around the object while the object retains the same UPN. Once a UPN is assigned to an object in AD, it is not affected by changes to that object (for example, other attributes values, moving, or renaming the object). Thus, you can administratively change the UPN at any time required.

The UPN is the preferred logon name for Win2K users and is commonly mapped to the user's e-mail name. Microsoft's intention is to consolidate the e-mail and logon namespaces so that the user need only remember a single name. I'll discuss the UPN in more detail throughout this part of the book, and re-examine it as part of the Exchange migration given its relationship between e-mail address and login name.

Planning for Windows 2000 Replication Traffic

As I discussed earlier in “Auditing Your Domain Structure,” auditing your current network environment is essential to planning for Win2K replication traffic. You need to use your network audit to identify areas of high-speed connectivity that can sustain AD sites. (Sites are discussed in Chapter 1.) For the purposes of this discussion, remember that an AD site is a collection of one or more IP subnets.

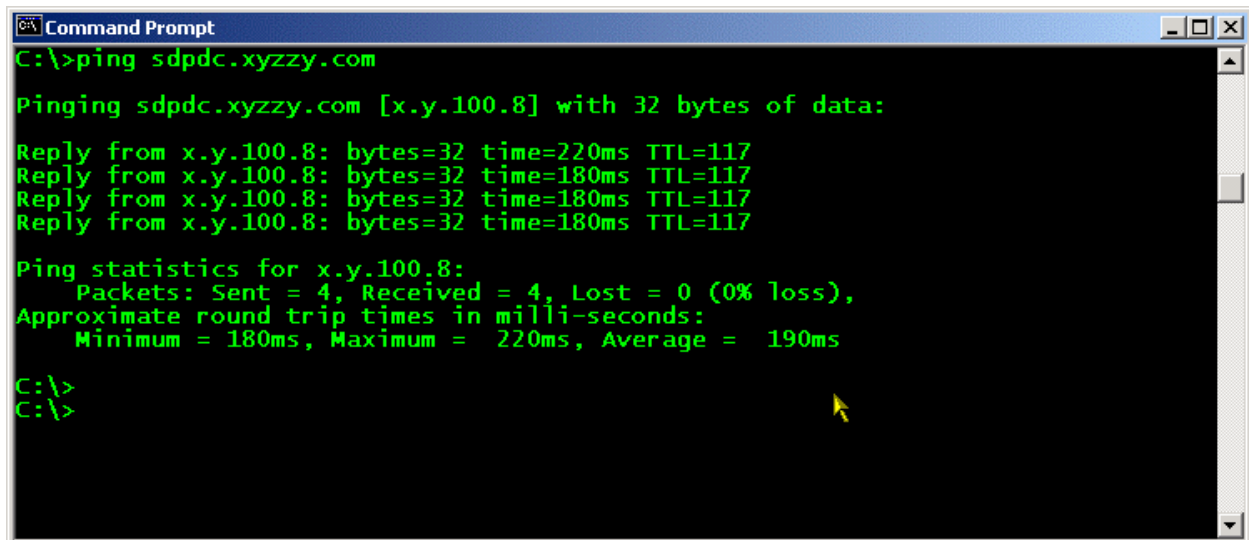
Identifying Potential AD Sites

Now, what exactly is a high-speed link? It depends on whom you ask. Seriously, though, in the context of Win2K, it's considered to be any area of connectivity that can allow you to ping a

device and receive a response in less than 200 milliseconds. Microsoft also recommends a boundary condition, whereby the available network connection should be at least 512 kilobits per second.

Although you may measure network connectivity at a specific instance in time, you must take into account the average usage of the network. If you find your network is saturated by traffic from other applications, you will still need to maintain at least the average availability of 512 kilobits per second in order to support intra-site replication, otherwise make it a site boundary. To measure network connectivity you can use network monitoring equipment or Network Monitor software that Microsoft provides as part of SMS or Win2K. Remember to monitor from the remote side of routers to accurately identify traffic patterns.

Figure 2.5 shows a sample ping over a WAN network with response times.



```
Command Prompt
C:\>ping sdpdc.xyzy.com

Pinging sdpdc.xyzy.com [x.y.100.8] with 32 bytes of data:

Reply from x.y.100.8: bytes=32 time=220ms TTL=117
Reply from x.y.100.8: bytes=32 time=180ms TTL=117
Reply from x.y.100.8: bytes=32 time=180ms TTL=117
Reply from x.y.100.8: bytes=32 time=180ms TTL=117

Ping statistics for x.y.100.8:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 180ms, Maximum = 220ms, Average = 190ms

C:\>
C:\>
```

Figure 2.5: A sample PING over WAN links.

When a response time goes below the 200-millisecond threshold, that link is a potential site boundary. This rough metric, and your audit information, allows you to determine which servers are suitable to include at which sites and where you might locate other sites to facilitate clients and minimize network traffic.

At any site, replication occurs constantly among domain controllers. Sites allow you to control how AD replicates with domain controllers at other sites, allowing you to optimize network traffic based on your network metrics as well as your business needs. Workstations also become attached to sites and always try to locate a domain controller at their own sites. This means that workstations add network traffic, so you may want to restrict them to areas of high-speed connectivity.

Reviewing Sample Site Designs

Let's look at some sample site designs, review the parameters, then discuss the implications of site boundaries and what they mean.

Figure 2.6 shows two high-speed LANs, one in San Francisco and one in San Diego, connected by a 56-kilobyte frame relay. The latency over the link is high with a similar ping time as shown in Figure 2.5. In this scenario there are two hundred clients and a few servers in each location. The two networks have connectivity to support single-site definition, and the frame relay marks the site boundaries.

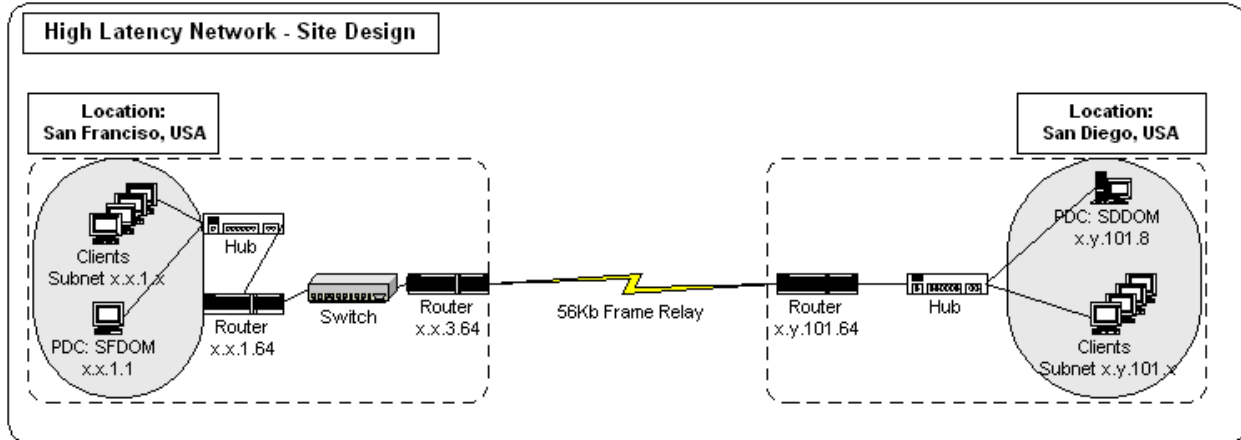


Figure 2.6: A site design for a high-latency network.

Although similar to Figure 2.6, Figure 2.7 illustrates a more complex scenario with thousands of clients and tens of servers on each side of the high-speed network. In this situation it makes sense to break the network into two separate sites due to the number of clients and servers in each physical location. The issue is not bandwidth, but a network management decision to control traffic and consign replication traffic to specific servers.

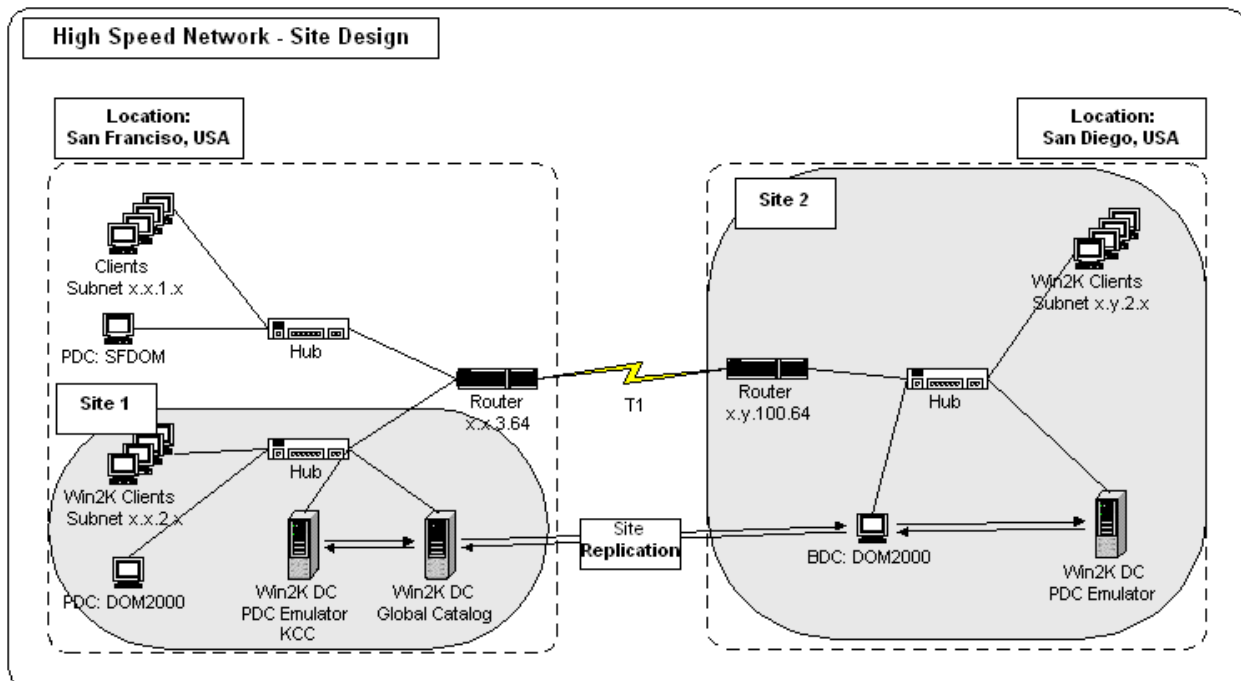


Figure 2.7: A site design for a complex high-speed network.

A general rule of thumb is to create site boundaries at:

- High latency network connections (that is, greater than 200ms)
- WAN boundaries with more than 256 computers on each side
- Available network capacity boundaries (that is, a network connection should offer at least 512 kilobits per second.)

The site design in Figure 2.7 shows the location of domain controllers, Global Catalog servers, and Knowledge Consistency Checkers, or KCCs. (Because of the complexity of this design, we can assume that each site has a KCC.) The environment at each site generates a high level of connectivity. Nevertheless, this design ensures reliability and minimizes costly replication traffic over the communications link between the physical locations.

KCCs ensure that replication is being performed optimally both at a site, and among sites, by continuously checking the network and prior replication traffic. KCCs also assign the role of bridgehead server for site-to-site replication and handles failures on the bridgehead server. Although I don't advise it, you can manually override all this using the AD Sites and Services MMC snap-in.

Global Catalog servers create high levels of replication traffic because they hold a subset of information on every object in the organization, across all managed domains. The impact of this is considerable, particularly in large-scale deployments. Thus, try to minimize the number of GCs you deploy on your network. This also applies to domain controllers in general.

In both sample site designs, separate sites are created for different reasons despite the difference in connection speeds. In the first scenario, multiple sites are created due to the latency issue. In the second scenario, multiple sites are created to ensure control over data flow.

Summary

In this chapter, I discussed many of the challenges you'll face in planning your Win2K migration and suggested approaches to deal with them. I then reviewed the information you'll need to gather that will be essential to your migration, then some of the tools and techniques you can use to gather that information. Finally, I reviewed some of the core technologies you'll need to understand to successfully migrate to Win2K.

In Chapter 3, I'll describe in detail the steps you need to take to prepare your network for Win2K, even before you perform the full migration.

Copyright statement

This site contains materials created, developed, or commissioned by Realtimepublishers.com, Inc. and is protected by international copyright and trademark laws. No material (including but not limited to the text, images, audio, and/or video) may be copied, reproduced, republished, uploaded, posted, transmitted, or distributed in any way, except that one copy may be downloaded for your personal, non-commercial use on a single computer. In connection with such use, you may not modify or obscure any copyright or other proprietary notice.

If you have any questions about these terms, or if you would like information about licensing materials from Realtimepublishers.com, please contact us via e-mail at info@realtimepublishers.com”