

SYBEX Sample Chapter

Active Directory[®] Best Practices 24seven[™] : Migrating, Designing, and Troubleshooting Brad Price

Chapter 3: Domain Name System Design

Copyright © 2004 SYBEX Inc., 1151 Marina Village Parkway, Alameda, CA 94501. World rights reserved. No part of this publication may be stored in a retrieval system, transmitted, or reproduced in any way, including but not limited to photocopy, photograph, magnetic or other record, without the prior agreement and written permission of the publisher.

ISBN: 0-7821-4305-9

SYBEX and the SYBEX logo are either registered trademarks or trademarks of SYBEX Inc. in the USA and other countries.

TRADEMARKS: Sybex has attempted throughout this book to distinguish proprietary trademarks from descriptive terms by following the capitalization style used by the manufacturer. Copyrights and trademarks of all products and services listed or described herein are property of their respective owners and companies. All rules and laws pertaining to said copyrights and trademarks are inferred.

This document may contain images, text, trademarks, logos, and/or other material owned by third parties. All rights reserved. Such material may not be copied, distributed, transmitted, or stored without the express, prior, written consent of the owner.

The author and publisher have made their best efforts to prepare this book, and the content is based upon final release software whenever possible. Portions of the manuscript may be based upon pre-release versions supplied by software manufacturers. The author and the publisher make no representation or warranties of any kind with regard to the completeness or accuracy of the contents herein and accept no liability of any kind including but not limited to performance, merchantability, fitness for any particular purpose, or any losses or damages of any kind caused or alleged to be caused directly or indirectly from this book.

Sybex Inc.
1151 Marina Village Parkway
Alameda, CA 94501
U.S.A.
Phone: 510-523-8233
www.sybex.com



chapter 3

Domain Name System Design

YOU CANNOT HAVE Active Directory without having the Domain Name System (DNS) in place. I know that is a blunt way to open the chapter, but it is the fundamental truth with this chapter. DNS is required when you implement Active Directory. Although you do not have to run Microsoft's version of DNS, there are many reasons why you would want to do so.

As with all services that are used within a network, you have many options as to how you will implement the service. However, you should follow some general guidelines if you want to make sure you are taking advantage of the best way to use DNS. Throughout this chapter, we are going to look at why DNS is required and how you can implement an efficient and secure DNS infrastructure. Later, in Chapter 14, "Maintaining DNS," I will cover troubleshooting DNS.

Tied Together

If you are looking to implement Active Directory within your environment, DNS is required. Active Directory cannot exist without DNS. If you haven't immersed yourself in the finer details of DNS, now is the time. If you think you understand how DNS works, you should still go back and review all of the new options that have been added to Windows Server 2003 DNS. Where Windows 2000 added some fancy new features into the Microsoft DNS world, such as support for dynamic updates and SRV records, Windows Server 2003 upped the ante even more with support for stub zones and the ability to use directory application partitions for Active Directory–integrated zones.

As I mentioned in the introduction to this chapter, you are not required to use Microsoft's implementation of DNS; UNIX BIND DNS will work just fine as long as it meets certain criteria. We will look at using BIND within your infrastructure later in the chapter.

Looking at the correlation between your Active Directory and DNS, you will find the two will share the same zone naming conventions. If your Active Directory domain name is going to be `zygort.1c1`, the DNS namespace will also be `zygort.1c1`. This is due to the fact that Active Directory needs to register records within the DNS zone in order for the Active Directory clients to locate domain controllers. The records in question are service locator records, more commonly referred to as *SRV records*.

As a domain controller comes online, part of its startup routine is to attempt registration of the SRV records that identify the services that are running on the domain controller. If the SRV records are not

listed within the zone, the client will not be able to locate the domain controller. If the SRV records are listed, the host name of the server that is providing the service is returned to the client. The client will then query the DNS server for the A record of the domain controller in order to resolve the IP address.

How to Resolve

Windows 2000 DNS servers introduced *forwarders* to the Microsoft DNS world. Using forwarders, you can specify another DNS server that will attempt to resolve queries when the local DNS server cannot. By default, a DNS server will use the DNS servers that are configured within the Root Hints tab of the DNS server properties. However, there may be instances when your DNS server cannot reach the root servers defined there or when you want to control the servers that perform the iterative queries from your organization.

There is one interesting configuration setting that you will find with a Windows 2000 DNS server: if you allow the DNS server to be created when you promote your first domain controller, it will become the root of your DNS infrastructure. All of the queries from clients will result in internal resolution only; you will not be able to resolve external zone information without additional configuration. If you do not want to isolate yourself, you will need to manually configure the DNS server with a zone that will be used to host Active Directory. In doing so, the DNS server will automatically configure itself to use the root hints.

NOTE For more information on root hints and forwarders, see the TechNet article 229840 at <http://support.microsoft.com/default.aspx?scid=kb;EN-US;229840>.

Another item to note, if a DNS server is configured as the root server for the organization, you cannot configure it to forward requests to another DNS server. If by accident this has happened to you, you can simply delete the root zone from the DNS server, which is specified by the dot (.), as seen in Figure 3.1. In the case of a Windows 2003 Server, the root zone is designated by `.(root)`, as seen in Figure 3.2. Once the root zone is deleted, you can enter external root servers into the root hints, as well as configure forwarders.

This behavior does not occur within a Windows Server 2003 DNS server when you promote the first domain controller. This doesn't mean that you should let Dcpromo install the DNS service; instead, you should configure the DNS zone first, and then promote the domain controller. Doing so will allow you to configure the zone the way you want and then allow the domain controller to register. Make sure that you configure the zone for dynamic updates, however. Otherwise, you will receive an error message stating the domain is not configured.

Windows Server 2003 introduced another method of forwarding, *conditional forwarding*. Using conditional forwarding, you can specify a DNS server that will be used to resolve queries based on the domain name in question. For example, if a user needs to resolve an address for `zygort.com` and if a conditional forwarder is created for the `zygort.com` domain, the DNS server will send a recursive query to the server specified within the forwarder setting. Figure 3.3 shows conditional forwarders configured for the zones `zygort.com` and `blomco.org`. Notice the All Other DNS Domains entry. Configuring DNS addresses within that setting specifies which servers will be used as standard forwarders if no conditional forwarders meet the query needs.

FIGURE 3.1
Root zone in Windows 2000

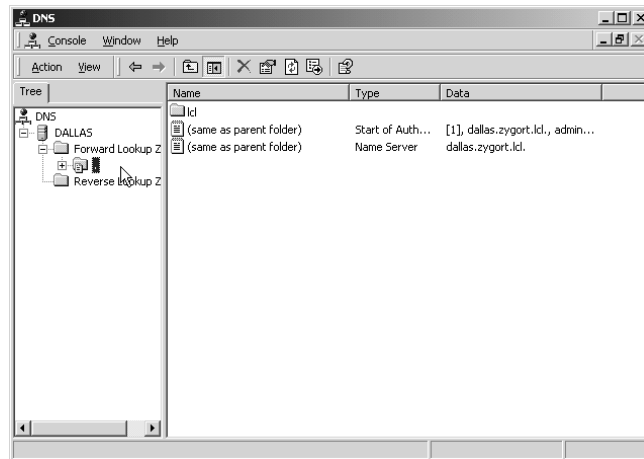
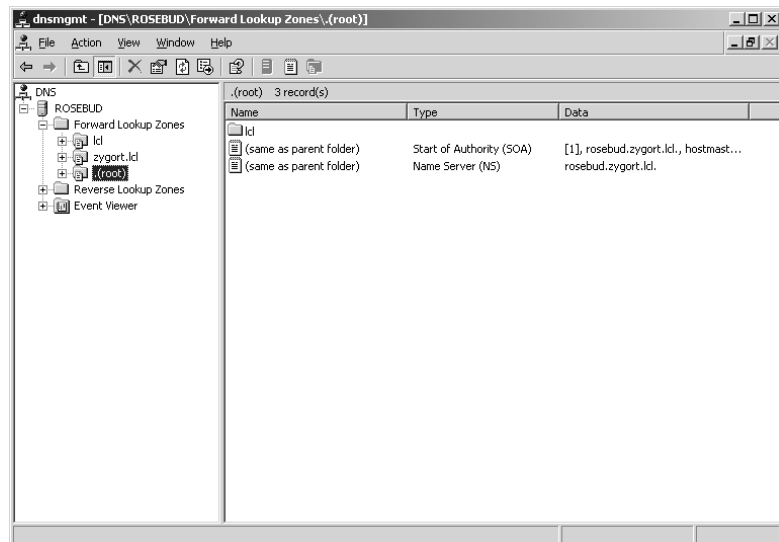
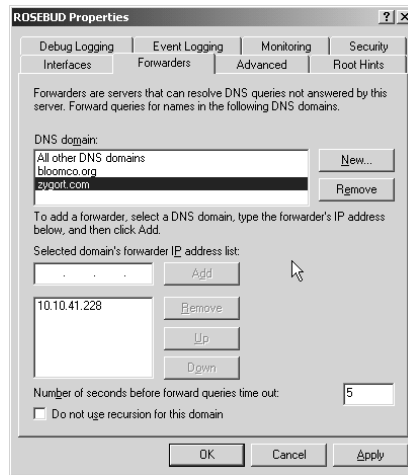


FIGURE 3.2
Root zone in Windows Server 2003



NOTE For more information on conditional forwarding, see the TechNet article 304991 at <http://support.microsoft.com/default.aspx?kbid=304491&product=winsvr2003>.

FIGURE 3.3
Conditional
forwarders



So Many Zone Types

For every zone that you use, you will need to determine how you will configure the DNS servers to use them. There are basically three zone types: primary, secondary, and stub. Of the three zone types, you have the choice of making standard and stub zones Active Directory–integrated, but secondary zones cannot be Active Directory–integrated. Each of them has their place within your infrastructure, but knowing when to choose one over the other can be confusing.

PRIMARY ZONES

Primary zones have traditionally been held on a single system and are known in the Microsoft world as *standard primary zones*. The limitation to these zones is their inherent single point of failure. Although the zone data can be transferred to another server that acts as the secondary zone, if the server holding the primary zone fails, you no longer have an update point. In order to make updates to the zone while the server holding the original primary zone is unavailable, you have to change a secondary zone to a primary.

Another limitation to standard primary zones also stems from the single update point. When using clients that support dynamic DNS updates, the only server in the zone that can receive the updates is the server holding the primary zone. Whenever a dynamic DNS client comes online, it queries its preferred DNS server for the Start of Authority (SOA) record for the zone in which it is preparing to register. The SOA record informs the client of the server that is authoritative for the zone. The client then sends the dynamic DNS registration information to the server holding the primary zone. This is not a problem unless the server with which the client is registering is across a slow or over-consumed WAN link. The additional DNS registration traffic may become too cumbersome. In addition, the same data then has to travel back across the WAN link if a server holding the secondary zone requires a zone transfer.

This has led administrators to create subdomains within the DNS hierarchy to support the remote locations. In this manner, the remote locations have their own DNS servers to hold their primary

zones, with the parent domain holding delegation records to the subdomain. Clients within the zone register locally, and the only data that needs to be sent across the WAN link are the queries for zone information and zone transfers if a secondary zone is configured on another server.

However, this scenario has two problems: you may not have an administrative staff in place at the remote locations, and query traffic could consume more bandwidth on the WAN link than the registration traffic would. So what is an administrator to do?

Besides evaluating the traffic that would be generated from either of the two scenarios to determine which will be the lesser of two evils, you could break away from the archaic DNS methodologies and start using the new and improved Microsoft DNS technologies. Using Active Directory–integrated zones greatly enhances your DNS infrastructure. Gone are the days of having one update point and tedious zone transfers. (Do I sound like a salesman yet?)

Active Directory–integrated zones boast the benefit of being able to share the responsibility of updating the zone, whether it is from dynamic DNS clients or manually entered records. The single point of administration and single point of failure disappear. There are limitations to using Active Directory–integrated zones, however.

First, you can create an Active Directory–integrated zone only on a DNS server that is also a domain controller. If you have a location where you do not want to place a domain controller, you will not be able to take advantage of Active Directory–integrated zones on a DNS server at that location. Of course, if you do not have enough clients to warrant placing a domain controller at that location, you probably will not have dynamic update client issues either.

The second limitation is only a limitation of Windows 2000 domain controllers and not typically a problem with Windows Server 2003 domain controllers. The zone data is replicated to every domain controller within the domain. As you will see in the section “Propagating the Changes,” Active Directory replication is far more efficient than zone transfers, but there are still some problems with replicating changes. Windows 2000 domain controllers will only replicate changes to other domain controllers within the same domain, and that replication goes out to all domain controllers, not just those that are DNS servers.

Windows Server 2003 made Active Directory–integrated zones more efficient, but at the same time, using the zones gave the administrators a little more to think about. Active Directory–integrated zones within a Windows Server 2003 environment do not hold the zone data within the domain partition; instead, a separate partition, a directory application partition, is used. An application directory partition does not rely on any specific domain within Active Directory, it can be replicated to any domain controller in the forest. If you look at the partitions within a Windows Server 2003 domain controller, you will find the typical partitions (Schema, Configuration, and Domain), and you will also find directory application partitions for the forest and domain.

When you create an Active Directory–integrated zone on a Windows Server 2003 domain controller, you have the option of determining the scope of replication for the zone. Four options are available:

- ◆ Replicating to all DNS servers in the forest
- ◆ Replicating to all DNS servers within a domain
- ◆ Replicating to all domain controllers within the domain
- ◆ Replicating to all domain controllers defined in the replication scope of a DNS application directory partition

If you choose the first option—replicating to all DNS servers within the forest—every DNS server within the forest will receive the DNS zone information. The zone information will be held within an application partition. This will cause the most replication because every DNS server within the forest will hold the records for the zone, but the only domain controllers that will receive the data will be those that host the DNS service. You cannot have any Windows 2000–based domain controllers in this scenario.

The second option—replicating to all DNS servers within the domain—will reduce the amount of replication traffic because only the domain controllers that are DNS servers for the domain in question will hold a copy of the domain records. As with the previous option, the zone is stored in an application partition. Again, as with the previous option, you cannot have any Windows 2000–based domain controllers within the domain.

The third option—replicating to all domain controllers within the domain—essentially makes all of the domain controllers within the domain behave as if they are Windows 2000 domain controllers. Every domain controller, whether or not it is a DNS server, will hold the data for the zone. If you still have some Windows 2000–based domain controllers that are DNS servers, this is the option you will choose.

The final option—replicating to all domain controllers defined in the replication scope of a DNS application directory partition—is also an option that is only available to Windows Server 2003 domain controllers. Using an application directory partition, you can choose which of the domain controllers will host a copy of the partition. In this manner, you can control exactly which domain controllers, that are also DNS servers, will host the zone data. If you do not want to replicate the zone to a server that is across a WAN, you will not have to replicate it.

NOTE For more information about controlling the replication scope by creating application partitions, also known as Active Directory Application Mode (ADAM), see Chapter 14, “Maintaining DNS.”

SECONDARY ZONES

Secondary zones still have their place within an organization. If you have a remote location where you do not want to support a domain controller but want to provide local resolution to the clients, you can create a secondary zone on a server within that location. This will reduce the amount of query traffic that has to pass across the WAN link, but you will be required to send zone transfers from a master server across the WAN link to the secondary. Typically, there will be more queries sent by clients than there will be dynamic updates from clients. Even so, you should monitor the traffic that is passing across the WAN link to determine if you are using the link appropriately.

STUB ZONES

New to Windows Server 2003 is the stub zone. Although the name may sound a little strange, it does perfectly describe this zone type. Stub zones do not contain all of the resource records from the zone, as the primary and secondary zones types do. Instead, only a subset of records populates the zone, just enough to provide the client with the information necessary to locate a DNS server that can respond to a query for records from the zone.

When you create the stub zone, it is populated with the SOA record along with the NS records and the A records that correspond to the DNS servers identified on the SOA record. All this is done

automatically. The administrator of the zone is not required to create the SOA, NS, or A records. Instead, as the zone is created, the DNS server will contact a server that is authoritative for the zone and request a transfer of those records. Once populated, the DNS server holding the stub zone will contact the authoritative server periodically to determine if there are any changes to the SOA, NS, and A records. You can control how often the DNS server requests updates by configuring the Refresh Interval on the SOA record for the zone.

As a client queries the DNS server to resolve the IP address for host, the DNS server is going to attempt to locate the A record for the hostname. If the DNS server is configured with a stub zone for the domain name contained within the query, the DNS server will send an iterative query directly to an authoritative DNS server for the zone. In Figure 3.4, you will find the common query path that is taken when a client is trying to resolve an address. In this case, the client is trying to locate `server1.dallas.bloomco.com`. When the client in `chicago.zygort.com` sends the recursive query to its DNS server, the DNS server will “walk the tree” by sending iterative queries to DNS servers along the path to eventually get to a DNS server that is authoritative for `dallas.bloomco.com`.

In Figure 3.5, we have configured the same server with a stub zone for `dallas.bloomco.com`. When the client sends the recursive query to its DNS server, the DNS server has a zone listed within its database that lets it know which servers to contact when trying to locate `dallas.bloomco.com`. The DNS server can then send a single iterative query to the authoritative server and then send the result back to the client, thereby making the resolution process far more efficient.

FIGURE 3.4
Standard name
resolution

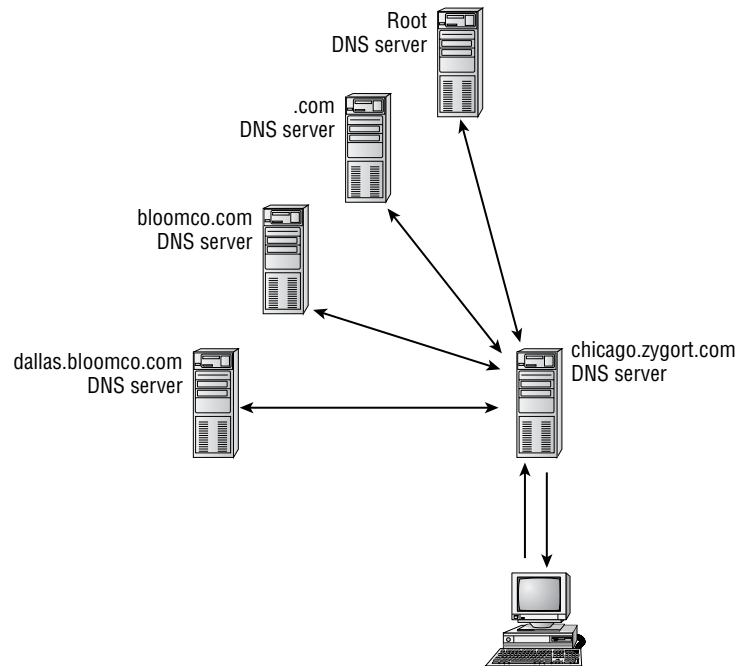
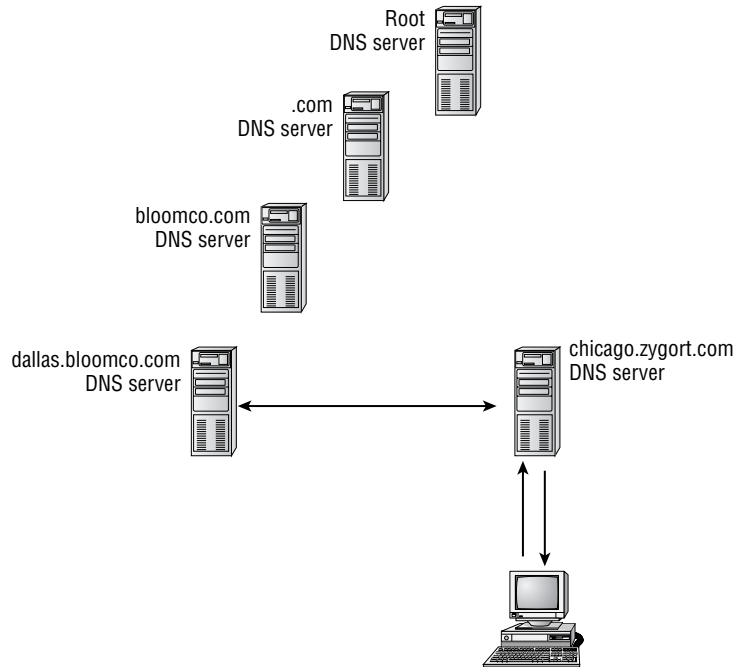


FIGURE 3.5
Name resolution
using a stub zone



So the question on your mind is “Why not use a conditional forwarder instead of the stub zone?” There are two reasons why you would want to use a stub zone over a conditional forwarder. First, the stub zone has automatic updating features. When the refresh interval on the SOA record is reached, the server holding the stub zone will contact an authoritative server for the zone and update the list of name servers and their associated addresses. Conditional forwarders rely on administrative staff to keep them updated. Secondly, conditional forwarders will require more processing power to perform the logic of evaluating the conditions to determine which one matches. Stub zone information is held within the DNS database and can be parsed far more quickly.

NOTE For more information on conditional forwarding and stub zones, see the Microsoft webcast at <http://support.microsoft.com/default.aspx?kbid=811118&product=winsvr2003>.

How to Name a Zone

Trying to determine what you are going to name your zone can be one of the more difficult things you will do. The name should be descriptive enough so that it can be easily remembered while at the same time short enough so that it is not too difficult to type. If you are using any other DNS servers within your environment besides Windows DNS server, you should follow the DNS naming guidelines, as set forth in RFC 952. This document spells out the characters that can be used within a DNS implementation. Any character within the ANSI character set is legal to use.

However, if your network uses only Windows-based DNS servers you can use extended characters from the UTF-8 character set. This includes the underscore (`_`) character that is so popular amongst Windows NT administrators. During migration from Windows NT to Windows 2000 or Windows Server 2003, you will not have to rename computers that use an underscore in their name in order for them to be added to the DNS zone. Be wary, however. If you have any DNS servers that cannot handle the extended character set, you will receive errors during zone transfers.

***TIP** You must register your Internet presence with an Internet registration authority so that you are ensured of owning your domain name. If you do not register your domain name, another company could register it and use it. Even if you do not plan to use the name on the Internet, register it so that it is reserved in case you ever do need an Internet presence.*

Internal and External Name Options

Basically, you have two options when you are choosing internal and external namespaces for your organization: using different namespaces or using the same namespace. Each of the options presents its own trials and tribulations for administrators, so you should take the time to plan which method you will implement.

Keeping Them Separate

If you have determined that your organization will need an Internet presence, you need to determine the name with which you will be identified. Your name should identify your company. Users who are accessing your external resources should find your name easy to understand. One guideline is to make your name short, yet understandable. The easier it is to remember and type, the easier it will be for users to return to your site. The name `zygort.com` is much easier to remember and type than `zygort-manufacturing-inc.com`. Plus, if you are using a subdomain as your internal name, the longer the external DNS name is, the longer the internal namespace will be as you append the subdomain. Users will not appreciate having to enter `accountspayable.accounting.corp.zygort.1c1`.

To keep your internal resources hidden from external users, you should keep the internal namespace different than the Internet namespace. If you want to keep the two namespaces separate, you have the option of making the internal domain name a child domain from the Internet namespace or having two completely different namespaces.

Even if you never add any delegation records to the Internet domain so that the internal domain name is available from the Internet domain, you will still be using a domain name structure that will make sense to your users. If you decide to make the internal domain accessible, you can add a delegation record to the DNS servers that are used for your Internet presence or create a subdomain to allow specific servers to be accessed.

Identical Confusion

Using the same name for your internal infrastructure that you are using to identify your organization on the Internet can be very time consuming and confusing. While users will not have any problem remembering just a single namespace, the administrative staff will have the burden of allowing users the ability to access both internal and external resources.

One of the basic rules for protecting your resources is not allowing external entities to discover your internal resources. If you want to use the same namespace internally as well as externally, you will have to use two completely different zones with the same namespace, in order to guarantee that they will not share any zone information. Otherwise, zone transfers or Active Directory replication will populate the DNS servers that the external clients use with information about your internal network. Letting anyone outside of your organization access this information is not a good thing.

Therein lies the problem. How do you allow your internal clients the ability to access resources outside of your internal infrastructure? For each of the web servers, SMTP servers, and any other server that is part of your Internet presence, you will have to manually enter the records into your internal DNS zones. If anything changes, you must make sure that you update the records accordingly. Missing any updates or forgetting to enter records for resources that the users need to access will cause plenty of phone calls to come your way!

Understanding the Current DNS Infrastructure

DNS has been around for many years, and chances are you will already have DNS within your infrastructure. Whether or not your current DNS implementation will support your needs will have to be determined. After all, what works for the Unix or Novell side of your network may not work the best for Active Directory. Case in point, DNS is normally a single master database. This means that updates and entries into the database can only be made on one server—the server holding the primary zone. Every other DNS server that holds a copy of the zone will use secondary zone types that contain read-only copies of the zone database. In order for clients that support dynamic update to enter their resource records within the database, they have to be able to contact the DNS server that hosts the primary zone. As mentioned earlier in the chapter, this is an inefficient method of utilizing DNS.

Unix and Novell DNS solutions that are already in place may not support the Active Directory requirements. At the very least, your DNS has to support SRV records as recorded in RFCs 2052 and 2782. If it doesn't, Active Directory domain controllers will not be found by Active Directory-aware clients. The best environment would be to have a DNS server that not only supports SRV records, but also support dynamic updates as recorded in RFC2136. If the DNS server does not support dynamic updates, you will have to manually enter the correct information for the domain controllers, which will include all of the SRV records that are found within the `NETLOGON.DNS` file that is created when the domain controller is promoted. Doing so could be a time-consuming, boring task.

After determining if the current DNS servers will support Active Directory, take a look at where the DNS servers are located and the client population that they serve. You probably retain DNS functionality at those locations. You should also determine whether or not you want to place DNS servers in locations that are not supported by local DNS servers. Ask yourself if the clients would be better served to have a local DNS server. The answer will be based on the difference between the queries made by the clients and the zone replication between DNS servers. In a large zone, you may have a large amount of zone transfer data, so you need to weigh that against the number of queries the clients are making. Use a network monitoring tool, such as Microsoft's Network Monitor or McAfee's Sniffer, to analyze the data that is traveling through your network links to determine where the majority of the data is coming from.

That Other DNS Server

What are you supposed to do if another division is responsible for the DNS infrastructure? Some companies have a complete division of responsibilities, and the DNS servers may not be under your control. People are very possessive of the things they manage, and you may find yourself fighting a battle to get the support you need in order to implement Active Directory.

The Windows-based DNS service was designed to interoperate with the latest DNS standards. It was also designed to support additional features that are available only to a Microsoft DNS implementation. These additional features are beneficial to administrators who want to have easier administration and additional security options.

Due to Windows Server 2003's compliance with DNS standards, it will interoperate with Berkley Internet Name Domain (BIND) DNS servers running versions 9.1.0, 8.2, 8.1.2, and 4.9.7. Windows Server 2003 DNS is also fully compliant with Microsoft Windows NT 4's DNS service.

As you will note in Table 3.1, Windows Server 2003's DNS service and BIND 9.1.0 support some important DNS features. Other versions of DNS do not support all of the options.

TABLE 3.1: DNS FEATURES SUPPORTED ON MULTIPLE PLATFORMS

	SRV RECORDS	DYNAMIC UPDATES	INCREMENTAL ZONE TRANSFER	STUB ZONES	CONDITIONAL FORWARDING
Windows Server 2003	X	x	x	x	x
Windows 2000	X	x	x		
Windows NT 4			x		
BIND 9.1.0	X	x	x	x	x
BIND 8.2	X	x	x		
BIND 8.1.2	X	x			
BIND 4.9.7	X				

Additional features are present in a Windows Server 2003 DNS environment that are not supported by other DNS servers. A list of additional features is presented in Table 3.2.

When attempting to integrate Windows Server 2003 DNS into an existing environment, take the previously mentioned interoperability into account. If the existing infrastructure does not support some of the features, you may be forced to upgrade the current infrastructure to Windows Server 2003 DNS so that all of the features that you need for your design are met.

In many companies, a DNS infrastructure is already controlled by a DNS group. If this group is unwilling to relinquish control of DNS or will not allow you to implement your own Windows Server 2003 DNS server, you may be forced to use the existing DNS services. Some organizations do have separate divisions that are responsible for specific portions of the network infrastructure. If your organization is one of them and you are not allowed to implement DNS due to departmental standards and regulations, you will be forced to use what the DNS administrative staff dictates. Be aware of the requirements for Active Directory, however. You may need to force them to upgrade their existing servers to handle the service locator (SRV) records and dynamic updates that Active Directory uses.

TABLE 3.2: DNS FEATURES NOT SUPPORTED ON NON-WINDOWS PLATFORMS

	SECURE DYNAMIC UPDATES	WINS INTEGRATION	UTF-8 CHARACTER ENCODING	ACTIVE DIRECTORY INTEGRATED ZONES	APPLICATION DIRECTORY SUPPORT	OBSOLETE RECORD SCAVENGING
Windows Server 2003	X	x	x	x	x	x
Windows 2000	X	x	x	x		x
Windows NT 4		x				
BIND 9.1.0						
BIND 8.2						
BIND 8.1.2						
BIND 4.9.7						

Propagating the Changes

In order to have an effective DNS solution, you will want to make sure that the clients have access to a local DNS server. In order to have DNS servers close to the clients, you will probably need to propagate the zone data to DNS servers in several locations.

Zone transfers come in two flavors: *authoritative zone transfers (AXFRs)* and *incremental zone transfers (IXFRs)*. An AXFR, sometimes referred to as a complete zone transfer, transfers the entire zone database when the zone transfer is initiated. An IXFR, as defined in RFC1995, only transfers the changes in the zone since the last zone transfer. As you can probably guess, the amount of data that is transferred during an IXFR transfer could be substantially less than that of an AXFR.

The choice to use zone transfers is usually made because the DNS servers in your environment are not Windows 2000– or Windows Server 2003–based. Third-party DNS servers do not participate in Active Directory replication, nor can they read the Active Directory database to determine the resource records that are used. To keep the network usage as low as possible, you should make sure the DNS servers all support IXFR. Otherwise, every time a zone transfer is initiated, the entire zone records will be passed to all of the appropriate DNS servers.

Active Directory–integrated zones can take advantage of Active Directory replication to propagate the changes made to resource records. When you use Active Directory replication, not only do you have the additional benefit of only having one replication topology, but a smaller amount of data is usually passed across the network. For instance, take a record that changes a couple of times before the replication or zone transfer occurs. In the case of a zone transfer, if the record changes twice before the transfer is initiated, both changes have to be sent, even if some of the data is no longer valid. In the case of Active Directory replication, only the effective changes are replicated. All of the erroneous information is discarded.

You also gain the advantage of the built-in functionality of replication. In an environment where you have multiple sites, many of which could be connected through WAN links, if there is considerable amounts of zone information to be transferred, the replication traffic that is sent between domain controllers in different sites is compressed to reduce network overhead.

DNS Design Best Practices

DNS is often met with a grimace, but it doesn't have to be that way. If you understand how DNS functions, you will be able to manage it easier. The following list is a quick set of best practices to follow when working with DNS in an Active Directory environment.

- ◆ In a Windows 2000 infrastructure, create the zone prior to promoting your first domain controller.
- ◆ Use conditional forwarding to control sending queries to DNS servers that will be responsible for performing the iterative queries for specific zones.
- ◆ Use stub zones to ease the administrative burden of updating the DNS servers that are responsible for the zone.
- ◆ Place DNS servers on the same side of a WAN link to the users and close to the clients that need resolution in order to reduce the bandwidth-hungry query traffic and zone transfers passing across WAN links.

Next Up

Now that we have covered the design options for a successful DNS implementation that will support Active Directory, we need to look at some of the other requirements for our Active Directory infrastructure. We are going to look at a plethora of topics in the next chapter. First, we will look at site design, and then we'll move on to Global Catalog servers and the master operations, also referred to as flexible single master operations. A well-organized infrastructure will allow you to use all of these services efficiently. A good understanding of these topics will aid you in designing your infrastructure so that you will not have to make many changes in the future.