Chapter 7

Implementing Domain Name System (DNS)

About This Chapter

In this chapter, you will learn how Domain Name System (DNS) is used to resolve host names on your local area network (LAN) and across the public Internet. Microsoft Windows 2000 includes an enhanced version of DNS. For more information about how Windows 2000 uses DNS, see the next chapter, "Using Windows 2000 Domain Name Service." This chapter is designed to provide you with an overview of DNS and how to implement the service on Windows 2000. By the end of this chapter, you will be able to identify the primary components of DNS, install and configure DNS, and troubleshoot the Domain Name Service on Windows 2000.

Before You Begin

To complete this chapter, you must have


Lesson 1: Introducing DNS

DNS is similar to a telephone book. Each computer on the Internet has both a host name and an Internet Protocol (IP) address. Typically, when you want to connect to another computer, you must enter a host name. Your computer then contacts a DNS server that cross-references the host name you provided to the actual IP address. This IP address is then used to connect to the remote computer. This lesson describes the architecture and structure of DNS.

After this lesson, you will be able to

- Describe the structure, architecture, and components of DNS
- Explain how DNS is used to resolve names and IP addresses

Estimated lesson time: 25 minutes

DNS Origins

Before the implementation of DNS, the creation of user-friendly computer names was done using HOSTS files that contained a list of names and associated IP addresses. On
the Internet, this file was centrally administered and each location would periodically
download a new copy. As the number of computers on the Internet grew, this became
an unmanageable solution. As a result, DNS was designed to replace the singularly
administered HOSTS file with a distributed database that would allow for a
hierarchical name space, distribution of administration, extensible data types, virtually
unlimited database size, and better performance. DNS is the name service for Internet
addresses that translates friendly domain names to numeric IP addresses. For example,
www.microsoft.com translates to 207.46.130.149. DNS is analogous to a telephone
book. The user looks up the name of the person or organization that he or she wants to
contact and cross-references the name to a telephone number. Similarly, a host
computer queries the name of a computer and a domain name server cross-references
the name to an IP address.

The Microsoft implementation of DNS Server became a part of the operating system in
Windows NT Server 4.0 and has continued to be included in Windows 2000.

DNS and Windows 2000

In addition to providing traditional Internet name resolution, DNS is the primary name
service of Windows 2000. It is, by design, a highly reliable, hierarchical, distributed,
and scalable database. Windows 2000 clients use DNS for name resolution and service
location, including locating domain controllers for logon. DNS in Windows 2000
provides a unique DNS Server implementation that is fully interoperable with other
standards-based implementations of DNS Server. For more information about the
version of DNS included in Windows 2000, please see the next chapter, "Using
Windows 2000 Domain Name Service."

How DNS Works

The purpose of the DNS database is to translate computer names into IP addresses, as
illustrated in Figure 7.1. In the DNS, the clients are called resolvers and the servers are
called name servers. DNS works using three main components: resolvers, name
servers, and the domain name space. With basic DNS communication, a resolver sends
queries to a name server. The name server returns the requested information, a pointer
to another name server, or a failure message, if the request cannot be satisfied.

DNS maps to the application layer and uses User Datagram Protocol (UDP) and
Transmission Control Protocol (TCP) as the underlying protocols. For performance
reasons, resolvers send UDP queries to servers first, then resort to TCP if truncation of
the returned data occurs.
Figure 7.1 Resolvers and name servers

Resolvers

A resolver provides clients with address information about other computers on the network. The function of the resolvers is to pass name requests between applications and name servers. The name request contains a query, such as the IP address of a Web site. The resolver is often built into the application or is running on the host computer as a library routine. Resolvers first send UDP queries to servers for increased performance and resort to TCP only if truncation of the returned data occurs.

Name Servers

A name server contains address information about other computers on the network. This information can be given to client computers that make a request to the name server. If the name server is not able to resolve the request, it can forward the request to a different name server. The name servers are grouped into different levels that are called domains. A domain is a logical group of computers in a large network. Access to each computer in a given group is controlled by the same server.

The Structure of DNS

The domain name space is a hierarchical grouping of names, as illustrated in Figure 7.2.
Root-Level Domains

Domains define different levels of authority in a hierarchical structure. The top of the hierarchy is called the root domain. References to the root domain are expressed by a period (.)

Top-Level Domains

The following are the present top-level domains:

- **com**: Commercial organizations
- **edu**: Educational institutions and universities
- **org**: Not-for-profit organizations
- **net**: Networks (the backbone of the Internet)
- **gov**: Nonmilitary government organizations
- **mil**: Military government organizations
- **num**: Phone numbers
- **arpa**: Reverse DNS
- **xx**: Two-letter country code

Top-level domains can contain second-level domains and hosts.

**NOTE**

An Internet Society committee is planning several additional top-level domains such as firm and web.

Second-Level Domains

Second-level domains can contain both hosts and other domains, called subdomains. For example, the Microsoft domain, microsoft.com, can contain computers such as ftp.microsoft.com and subdomains such as dev.microsoft.com. The subdomain dev.microsoft.com can contain hosts such as ntservers.microsoft.com.

Host Names

The domain name is used with the host name to create a fully qualified domain name (FQDN) for the computer. The FQDN is the host name followed by a period (.), followed by the domain name. For example, this could be fileserver1.microsoft.com, where fileserver1 is the host name and microsoft.com is the domain name.

Zones
The administrative unit for DNS is the zone. A zone is a subtree of the DNS database that is administered as a single, separate entity. It can consist of a single domain or a domain with subdomains. The lower-level subdomains of a zone can also be split into separate zones.

Zones of Authority

A zone of authority is the portion of the domain name space for which a particular name server is responsible. The name server stores all address mappings for the domain name space within the zone and answers client queries for those names. The name server's zone of authority encompasses at least one domain. This domain is referred to as the zone's root domain. You can also have a secondary DNS server that can copy domain information across the network from the primary DNS server that has authority over at least one zone. This is called a zone transfer.

As illustrated in Figure 7.3, microsoft.com is a domain, but the entire domain is not controlled by one zone file. Part of the domain is located in a separate zone file for DEV.MICROSOFT.COM. Breaking up domains across multiple zone files may be necessary for distributing management of the domain to different groups or for data replication efficiency.

![Domains across multiple zones](image)

**Figure 7.3** Domains across multiple zones

Name Server Roles

DNS name servers can be configured in different roles, affecting how they store and maintain their database of names. A Microsoft DNS server can be either a primary or secondary DNS server to another Microsoft DNS server, or to a DNS server running under another operating system such as UNIX. The minimum number of DNS servers you need in order to serve each zone is two—a primary and a secondary. Both a primary and a secondary server are required to provide database redundancy and a degree of fault tolerance.

Primary Name Servers

A primary name server is a DNS server that gets the data for its zones from the local DNS database files. When a change is made to the zone data, such as delegating a portion of the zone to another DNS server or adding hosts in the zone, the change must be made on the primary DNS server so that the new information is entered in the local zone file.

Secondary Name Servers
A secondary name server gets its zone data file from the primary DNS server that is authoritative for that zone. The primary DNS server sends a copy of the zone file to the secondary DNS server in a process referred to as a zone transfer.

There are three reasons to have secondary name servers:

- **Redundancy.** You need at least one primary and one secondary name server for each zone. The computers should be as independent as possible. Generally, plan to install the primary and secondary servers on different subnets to provide continual support for DNS name queries if one subnet should go down.

- **Faster access for remote locations.** If you have a number of clients in remote locations, having secondary name servers (or other primary name servers for subdomains) prevents these clients from communicating across slow links for name resolution.

- **Reduction of load.** Secondary name servers reduce the load on the primary server.

Because information for each zone is stored in separate files, this primary or secondary designation is defined at a zone level. This means that a particular name server may be a primary name server for certain zones and a secondary name server for other zones.

**Master Name Servers**

When you define a zone on a name server as a secondary zone, you must designate another name server from which to obtain the zone information. The source of zone information for a secondary name server in a DNS hierarchy is referred to as a master name server. A master name server can be either a primary or secondary name server for the requested zone. When a secondary name server starts up, it contacts its master name server and initiates a zone transfer with that server.

**Caching-Only Servers**

Although all DNS name servers cache queries that they have resolved, caching-only servers are DNS name servers that only perform queries, cache the answers, and return the results. In other words, they are not authoritative for any domains (no zone data is kept locally) and they only contain information that they have cached while resolving queries.

When trying to determine when to use such a server, keep in mind that when the server is initially started, it has no cached information and must build this information up over time as it services requests. Less traffic is generated between servers because the server is not doing a zone transfer. This is important if you have a slow connection between sites.

**Lesson Summary**

As an improvement to the original method of resolving host names to an IP address on the Internet, DNS was created. In DNS, a client (called a resolver) sends queries to a name server. Name servers then take name requests and resolve computer names to IP addresses. The domain name space is a hierarchical grouping of root-level domains, top-level domains, second-level domains, and host names. Specific servers are
Lesson 2: Name Resolution and DNS

Files

There are three types of queries that a client (resolver) can make to a DNS server: recursive, iterative, and inverse. These servers store their DNS information in four possible files: database, reverse lookup, cache, and boot files.

After this lesson, you will be able to

- Explain how recursive, iterative, and inverse queries work
- Explain how queries are placed in a cache for future requests

Estimated lesson time: 10 minutes

Recursive Queries

In a recursive query, the queried name server is petitioned to respond with the requested data, or with an error stating that data of the requested type does not exist or that the domain name specified does not exist. The name server cannot refer the request to a different name server.

Iterative Queries

In an iterative query, the queried name server gives the best answer it currently has back to the requester. This answer may be the resolved name or a referral to another name server that may be able to answer the client's original request.

Figure 7.4 shows an example of both recursive and iterative queries. In this example, a client within a corporation is querying its DNS server for the IP address for www.microsoft.com.

1. The resolver sends a recursive DNS query to its local DNS server asking for the IP address of www.microsoft.com. The local name server is responsible for resolving the name and cannot refer the resolver to another name server.

2. The local name server checks its zones and finds no zones corresponding to the requested domain name. It then sends an iterative query for www.microsoft.com to a root name server.

3. The root name server has authority for the root domain and will reply with the IP address of a name server for the com top-level domain.

4. The local name server sends an iterative query for www.microsoft.com to the com name server.
5. The com name server replies with the IP address of the name server servicing the microsoft.com domain.

6. The local name server sends an iterative query for www.microsoft.com to the microsoft.com name server.

7. The microsoft.com name server replies with the IP address corresponding to www.microsoft.com.

8. The local name server sends the IP address of www.microsoft.com back to the original resolver.

![Diagram of recursive and iterative queries]

**Figure 7.4 Recursive and iterative queries**

**Inverse Queries**

In an inverse query, the resolver sends a request to a name server to resolve the host name associated with a known IP address. There is no correlation between host names and IP addresses in the DNS name space. Therefore, only a thorough search of all domains guarantees a correct answer.

To prevent an exhaustive search of all domains for an inverse query, a special domain called in-addr.arpa was created. Nodes in the in-addr.arpa domain are named after the numbers in the dotted-decimal representation of IP addresses. Because IP addresses get more specific from left to right and domain names get less specific from left to right, the order of IP address octets must be reversed when building the in-addr.arpa domain. With this arrangement, administration of lower limbs of the in-addr.arpa domain can be delegated to organizations as they are assigned their class A, B, or C IP addresses.

Once the in-addr.arpa domain is built, special resource records called pointer (PTR) records are added to associate the IP addresses and the corresponding host name. For example, to find a host name for the IP address 157.55.200.51, the resolver queries the DNS server for a PTR record for 51.200.55.157.inaddr.arpa. The PTR record found contains the host name and corresponding IP address 157.55.200.51. This information is sent back to the resolver. Part of the administration of a DNS name server is ensuring that PTR records are created for hosts.

**Caching and Time to Live**
When a name server is processing a recursive query, it may be required to send out several queries to find the answer. The name server caches all of the information that it receives during this process for a time that is specified in the returned data. This amount of time is referred to as the Time to Live (TTL). The name server administrator of the zone that contains the data decides on the TTL for the data. Smaller TTL values help ensure that data about the domain is more consistent across the network if this data changes often. However, this also increases the load on name servers.

Once data is cached by a DNS server, it must start decreasing the TTL from its original value so that it will know when to flush the data from its cache. If a query comes in that can be satisfied by this cached data, the TTL that is returned with the data is the current amount of time left before the data is flushed from the DNS server cache. Client resolvers also have data caches and honor the TTL value so that they know when to expire the data.

DNS Configuration Files

The DNS is a hierarchical, distributed database. The database itself consists of resource records, which primarily consist of a DNS name, a record type, and data values that are associated with that record type. For example, the most common records in the DNS database are address records, where the name of an address record is the name of a computer, and the data in the record is the TCP/IP address of that computer.

To resolve names, servers consult their zones (also called DNS database files, or simply, db files). The zones contain resource records (RRs) that make up the resource information associated with the DNS domain. For example, some RRs map friendly names to IP addresses, and others map IP addresses to friendly names.

Start of Authority Record

The first record in any database file must be the start of authority (SOA) record. The SOA defines the general parameters for the DNS zone. The following is an example of an SOA record:

```
@   IN  SOA     nameserver.example.microsoft.com.
postmaster.example.microsoft.com. ( 
    1            ; serial number
    3600         ; refresh   [1h]
    600          ; retry     [10m]
    86400        ; expire    [1d]
    3600 )       ; min TTL   [1h]
```

The following rules apply to all SOA records:

- The at symbol (@) in a database file indicates "this server."
- IN indicates an Internet record.
- Any host name not terminated with a period (.) will be appended with the root domain.
- The @ symbol is replaced by a period (.) in the e-mail address of the administrator.
Parentheses ( () ) must enclose line breaks that span more than one line.

Name Server Record

The name server (NS) record lists the additional name servers. A database file may contain more than one NS record. The following is an example of an NS record:

@ IN NS nameserver2.microsoft.com

Host Record

A host address resource record (A) statically associates a host name to its IP address. Host records will comprise most of the database file and will list all hosts within the zone. The following are examples of host records:

rhino IN A 157.55.200.143
localhost IN A 127.0.0.1

CNAME Record

A canonical name (CNAME) record enables you to associate more than one host name with an IP address. This is sometimes referred to as aliasing. The following is an example of a CNAME record:

FileServer1 CNAME rhino
www CNAME rhino
ftp CNAME rhino

The Reverse Lookup File

The reverse lookup file (z.y.x.w.in-addr.arpa) allows a resolver to provide an IP address and request a matching host name. A reverse lookup file is named like a zone file according to the in-addr.arpa zone for which it is providing reverse lookups. For example, to provide reverse lookups for the IP network 157.55.28.0, a reverse lookup file is created with a file name of 57.157.in-addr.arpa. This file contains SOA and NS records similar to other DNS database zone files, as well as PTR records.

This DNS reverse lookup capability is important because some applications provide the capabilities to implement security based on the connecting host names. For instance, if a browser sends a request to an Internet Information Server (IIS) Web server with this security arrangement, the Web server would contact the DNS server and do a reverse name lookup on the client’s IP address. If the host name returned by the DNS server is not in the access list for the Web site or if the host name was not found by DNS, then the request would be denied.

NOTE

Windows 2000 does not require reverse lookup zones to be configured. Reverse-lookup zones might be necessary for other applications or for administrative convenience.
The PTR Record

PTR records provide an address-to-name mapping within a reverse lookup zone. IP numbers are written in backward order and “in-addr.arpa” is appended to the end to create this PTR record. As an example, looking up the name for 157.55.200.51 requires a PTR query for the name 51.200.55.157.in-addr.arpa. An example might read


The Cache File

The CACHE.DNS file contains the records of the root domain servers. The cache file is essentially the same on all name servers and must be present. When the name server receives a query outside its zone, it starts resolution with these root domain servers. An example entry might read

.                       3600000     IN     NS     A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.     3600000     A             198.41.0.4

The cache file contains host information that is needed to resolve names outside of authoritative domains, and also contains names and addresses of root name servers. The default file provided with the Windows 2000 DNS Server has the current records for all of the root servers on the Internet, and is stored in the %SystemRoot%\System32\Dns folder. For installations not connected to the Internet, the file should be replaced to contain the name server's authoritative domains for the root of the private network.

The Boot File

The boot file is the startup configuration file on the Berkeley Internet Name Daemon-specific implementation of DNS. This file contains host information needed to resolve names outside of authoritative domains. The file is not defined in a Request for Comments (RFC) and is not needed to be RFC-compliant. It is supported by Windows 2000 to improve compatibility with traditional, UNIX-based DNS services. The Berkeley Internet Name Daemon boot file controls the startup behavior of the DNS server. Commands must start at the beginning of a line and no spaces can precede commands. Table 7.1 shows descriptions of some of the boot file commands supported by Windows 2000.

**Table 7.1 Windows 2000 Boot File Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Directory</td>
<td>Specifies a directory where other files referred to in the boot file can be found.</td>
</tr>
<tr>
<td>Cache</td>
<td>Specifies a file used to help the DNS service contact name servers for the root domain. This command and the file it refers to must be present. A cache file suitable for use on the Internet is provided with Windows 2000.</td>
</tr>
<tr>
<td>Primary</td>
<td>Specifies a domain for which this name server is authoritative and a data-base file that contains the resource records for that domain</td>
</tr>
</tbody>
</table>
Lesson Summary

When clients need to resolve a host name or IP address, they can make one of three queries to DNS servers: recursive, iterative, or inverse. A DNS server will only return the information it has in cache, including the potential of an error, when a client makes a recursive request. A more typical query is an iterative query. When a client makes an iterative query, the DNS server will return the requested information or provide the client with an alternative DNS server that will provide the correct information. The third type of query, inverse, provides reverse-lookup information. If a DNS client needs a host name resolved from a known IP address, an inverse query is sent to the DNS server.

DNS servers store their name and configuration information in four files: database, reverse lookup, cache, and boot files. Windows 2000 and the included DNS Manager allow you to configure these files with a graphical user interface that is described in more detail in Chapter 8, "Using Windows 2000 Domain Name Service."

Lesson 3: Planning a DNS Implementation

The configuration of your DNS servers depends on factors such as the size of your organization, organization locations, and fault-tolerance requirements. This lesson gives you an idea of how to configure DNS for your site. It contains scenarios that measure your network planning knowledge prior to installing DNS.
After this lesson, you will be able to

- Register a DNS server with the parent domain
- Estimate the number of DNS name servers, domains, and zones needed for a network

Estimated lesson time: 40 minutes

DNS Considerations

Although Windows 2000 and its name resolution require a DNS server, the DNS server itself does not have to reside on a server running Windows 2000. In addition, it does not even have to be on your local network. As long as Windows 2000 can be configured to reference a valid DNS server that supports the necessary record types, such as one hosted by your Internet service provider (ISP), you can provide the required name resolution capabilities to Windows 2000. However, with the increased functionality implemented in the version of DNS provided with Windows 2000, you may decide it is worthwhile to install and configure your own DNS server. For purposes of this lesson, it is assumed you have decided to implement your own Windows 2000 DNS server.

If your organization, regardless of size, wants to use a second-level domain, the Internet Network Information Center (InterNIC) must be informed of the domain name of your organization and the IP addresses of at least two DNS servers that service the domain. You can also set up additional DNS servers within your organization that are independent of the Internet.

For reliability and redundancy, Microsoft recommends that at least two DNS servers be configured per domain—a primary and a secondary name server. The primary name server maintains the database of information, which is replicated to the secondary name server. This replication allows name queries to be serviced even if one of the name servers is unavailable. The replication schedule can be configured depending on how often names change in the domain. Replication should be frequent enough so that changes are known to both servers. However, excessive replication can increase network traffic and increase load on the name server.

Registering with the Parent Domain

Once you have your DNS server or servers configured and installed, you need to register with the DNS server that is above you in the hierarchical naming structure of DNS. Figure 7.5 provides an example of registering your DNS server with the domain level above it. The parent system needs the names and addresses of your name servers and may require other information, such as the date that the domain becomes available and the names and mailing addresses of contact people.
Figure 7.5 Registering your DNS server with the domain level above it

If you are registering with a parent below the second-level domain, check with the administrator of that system to determine the information you need to supply.

Practice: Implementing DNS

In this practice, you work through three DNS implementation scenarios. In each scenario, you estimate the number of DNS name servers, domains, and zones needed for a network. Each scenario describes a company that is migrating to Windows 2000 and wants to implement directory services. You will answer some questions involved in drafting a DNS network design for each company using unique criteria. The purpose of these practices is to measure your network planning knowledge prior to installing DNS. This will serve as a baseline to measure how much you have learned at the completion of this course and will help you start thinking about DNS network design.

Scenario 1: Designing DNS for a Small Network

The Northwind Company is in the process of replacing its older mid-range computer with a computer running Windows 2000. Most employees access the mid-range system through terminal devices. Some users have 486 computers and a few have Pentium computers; these computers are not networked. The company has already purchased the hardware for the migration.

The network will be used for basic file and print sharing and will also have one Windows 2000-based server running Microsoft SQL Server 7. The majority of users will need access to the computer running SQL Server 7. Desktop applications will be installed on the local computers, but data files will be saved on the servers.

The Northwind Company would like to be connected to the Internet so employees can receive e-mail.

Draft a network design using the criteria shown in Table 7.3.

Table 7.3 Network Design Criteria
The design will take into account

- Number of users
- Number of administrative units
- Number of sites

Based on these design objectives, answer the following questions:

1. How many DNS domains will you need to configure?
2. How many subdomains will you need to configure?
3. How many zones will you need to configure?
4. How many primary name servers will you need to configure?
5. How many secondary name servers will you need to configure?
6. How many DNS cache-only servers will you need to configure?

**Answers**

**Scenario 2: Designing DNS for a Medium-Size Network**

You are consulting for the Northwind Company, which has 8,795 users. There are 8,000 users located in four primary sites, with the remaining employees located in 10 branch offices in major U.S. cities. The company has decided to upgrade its existing LANs to Windows 2000 Server. The organization has also decided to centralize all user accounts in a single location at the corporate headquarters.

As illustrated in Figure 7.6, the four primary sites are connected by T1 lines. The branch offices are connected to the nearest primary site by 56 Kilobit per second.
Three of the four primary sites are independent business units and operate independently of the others. The fourth is corporate headquarters. Branch offices have between 25 and 250 users needing access to all four of the primary sites but seldom needing access to the other branch offices.

In addition to the 10 branch offices, you have discovered that the company has a temporary research location employing 10 people. The site has one server that connects to Boston using dial-on-demand routers. This site is expected to be shut down within six months. It is a stand-alone operation requiring connectivity for messaging only.

Primary sites will continue to maintain their own equipment and the equipment of the branch offices connected to them. Currently, bandwidth utilization is at 60 percent during peak times. Future network growth is expected to be minimal for the next 12 to 18 months.

Draft a network design using the criteria shown in Table 7.4.

Table 7.4 Network Design Criteria

<table>
<thead>
<tr>
<th>Environmental Components</th>
<th>Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Users</td>
<td>8,795</td>
</tr>
<tr>
<td>Location(s)</td>
<td>Four primary sites, with 10 branch sites in major cities in the U.S. No plans for opening any international locations.</td>
</tr>
<tr>
<td>Administration</td>
<td>Full-time administrators at each of the four primary sites. Some of the smaller sites have part-time administrators.</td>
</tr>
<tr>
<td>Number of name servers</td>
<td>To be determined.</td>
</tr>
<tr>
<td>Number of cache servers</td>
<td>DNS cache servers are needed in each of the remote</td>
</tr>
</tbody>
</table>
Other branch offices include: Los Angeles, 40 users; Salt Lake City, 25 users; Montreal, 30 users; New Orleans, 25 users; Kansas City, 25 users; Washington, DC, 100 users; Denver, 200 users; Miami, 75 users.

The design must take into account:

- Number of users
- Number of administrative units
- Number of sites
- Speed and quality of links connecting sites
- Available bandwidth on links
- Expected changes to network
- Line of business applications

Based on these design objectives, answer the following questions:

1. How many DNS domains will you need to configure?
2. How many subdomains will you need to configure?
3. How many zones will you need to configure?
4. How many primary name servers will you need to configure?
5. How many secondary name servers will you need to configure?
6. How many DNS cache-only servers will you need to configure?
7. Use the following mileage chart to design a zone/branch office configuration based on the geographical proximity between each primary site and branch office. Branch offices should be in the same zone as the nearest primary site.

**Mileage Chart**

<table>
<thead>
<tr>
<th>City</th>
<th>Atlanta</th>
<th>Boston</th>
<th>Chicago</th>
<th>Portland, OR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dallas</td>
<td>807</td>
<td>1817</td>
<td>934</td>
<td>2110</td>
</tr>
</tbody>
</table>

**Answers**

Zones for each branch office (based on geographical proximity):
Scenario 3: Designing DNS for a Large Network

The Northwind Company has 60,000 users located around the world. The corporate headquarters are in Geneva, Switzerland. Headquarters for North and South America are located in New York City. The Australia and Asian headquarters are located in Singapore. Each of the regional headquarters will maintain total control of users within their areas. Users require access to resources in the other regional headquarters. The three regional headquarters sites are connected by T1 lines.

Each of the three regional headquarters has lines of business applications that need to be available to all sites within their areas, as well as the other regional headquarters. The Malaysian and Australian subsidiaries have major manufacturing sites to which all regional subsidiaries need access.

These lines of business applications are all running on Windows 2000-based servers. These computers will be configured as servers within the domains. The links among Singapore, Australia, and Malaysia are typically operating at 90 percent utilization. The Asia and Australia region has 10 subsidiaries comprising Australia, China, Indonesia, Japan, Korea, Malaysia, New Zealand, Singapore, Taiwan, and Thailand.

Due to import restrictions with some of the subsidiaries, it has been decided to give control of the equipment to each subsidiary and to have a resource domain in each subsidiary. Lately, most of the computers the subsidiaries have purchased are running Windows 2000 Professional. The company has authorized redundant hardware where you can justify it.

To keep this scenario reasonable, the questions and answers deal only with the Asia and Australia region.

Draft a network design using the criteria in Table 7.5.

Table 7.5 Network Design Criteria

<table>
<thead>
<tr>
<th>Environmental Components</th>
<th>Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Users in Asia and</td>
<td>25,000 evenly distributed across all of the subsidiaries</td>
</tr>
</tbody>
</table>
The design for the Asia and Australia region must take into account:

- Number of users
- Number of administrative units
- Number of sites
- Speed and quality of links connecting sites
- Available bandwidth on links
- Expected changes to network
- Line of business applications

Based on these design objectives, answer the following questions:

1. How many DNS domains will you need to configure?
2. How many subdomains will you need to configure?
3. How many zones will you need to configure?
4. How many primary name servers will you need to configure?
5. How many secondary name servers will you need to configure?
6. How many DNS cache-only servers will you need to configure?

Answers

Lesson Summary
Depending on the size of your organization and configuration, you may want to configure DNS for your site. Windows 2000 requires access to a DNS server to provide complete functionality. This DNS server can be on your local network or provided remotely by your ISP. However, the DNS implementation included in Windows 2000 has additional features beyond those of traditional DNS servers. For more information about these new features, please see Chapter 8, "Using Windows 2000 Domain Name Service."

Lesson 4: Installing DNS

Microsoft DNS is an RFC-compliant DNS server; as a result, it creates and uses standard DNS zone files and supports all standard resource record types. It is interoperable with other DNS servers and includes the DNS diagnostic utility NSLOOKUP. Microsoft DNS is tightly integrated with Windows Internet Name Service (WINS) and is administered through the graphical administration utility called DNS Manager. In this lesson, you will install the DNS service on Windows 2000.

After this lesson, you will be able to

- Install the Microsoft DNS Server service
- Troubleshoot DNS with NSLOOKUP

Estimated lesson time: 45 minutes

Before installing the Microsoft Windows 2000 DNS Server service, it is important that the Windows 2000 server's TCP/IP protocol be configured correctly. The DNS Server service obtains the default settings for the host name and domain name through the Microsoft TCP/IP Properties dialog box. The DNS Server service will create default SOA, host, and NS records based on the specified domain name and host name. If the host name and domain name are not specified, only the SOA record is created.

Practice: Installing the DNS Server Service

In this procedure, you install the Microsoft DNS Server service. You will configure DNS in a later lesson.

Before you continue with the lesson, run the Ch07a.exe demonstration file located in the Media folder on the Supplemental Course Materials CD-ROM that accompanies this book. The file provides an overview of installing the DNS server service.
NOTE

Complete this procedure from the computer you designate as the DNS server.

Before configuring DNS, verify that your DNS client settings are correct.

- **To verify DNS client settings**

  1. Right-click My Network Places, then click Properties.

     The Network And Dial-Up Connections dialog box appears.

  2. Right-click the connection (typically the Local Area Network properties) for which you want to configure the DNS server, then click Properties.

     The Connection Properties dialog box appears.

  3. Click Internet Protocol (TCP/IP), then click Properties.

     The Internet Protocol (TCP/IP) Properties dialog box appears.

  4. On the Internet Protocol (TCP/IP) Properties page, enter the IP address of the existing DNS server in the Preferred DNS Server field.

     You can also add the IP address of an alternate DNS server in the Alternate DNS Server field.

  5. If you need to specify more than one alternate DNS server, click Advanced, click the DNS tab, then enter the servers in the DNS Server Addresses box.

  6. Click OK to close the TCP/IP Properties dialog box.

  7. Click OK to close the Connection Properties dialog box.

- **To install the DNS Server service**

  1. In Control Panel, double-click Add/Remove Programs, then click Add/Remove Windows Components.

     The Windows Components Wizard appears.

  2. Click Networking Services, then click Details.

     The Networking Services dialog box appears.

  3. If it is not already selected, select the check box next to Domain Name System (DNS) as illustrated in Figure 7.7, then click OK.

  4. Click Next.

     Windows 2000 installs DNS.
5. Click Finish.

![Figure 7.7 Domain Name System (DNS) check box in Networking Services](image)

**Troubleshooting DNS with NSLOOKUP**

NSLOOKUP is a useful tool for troubleshooting DNS problems such as host name resolution. When you start NSLOOKUP, it shows the host name and IP address of the DNS server that is configured for the local system, and then displays a command prompt for further queries. If you type a question mark (?), NSLOOKUP shows all available commands. You can exit the program by typing exit. To look up a host's IP address using DNS, type the host name and press Enter. NSLOOKUP defaults to using the DNS server configured for the computer on which it is running, but you can focus it on a different DNS server by typing `server <name>` (where `<name>` is the host name of the server you want to use for future lookups). Once another server is specified, anything entered after that point is interpreted as a host name.

**NSLOOKUP Modes**

NSLOOKUP has two modes: interactive and noninteractive. If a single piece of data is needed, use noninteractive or command-line mode. If more than one piece of data is needed, interactive mode can be used.

**NSLOOKUP Syntax**

NSLOOKUP.EXE is a command line administrative tool for testing and troubleshooting DNS servers. The following syntax is used to run the NSLOOKUP utility:

```
nslookup [option ...] [computer-to-find | _ [server]]
```

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>–option ...</td>
<td>Specifies one or more NSLOOKUP commands. For a list of</td>
</tr>
</tbody>
</table>


To use NSLOOKUP in command mode

1. At a command prompt, modify the properties so that the command prompt window has a screen buffer size of 50.

   As illustrated in Figure 7.8, use the Layout property page to do this. You should apply this change to all future instances of the command prompt window; it will be needed in later lessons.

2. Type the following command:

   ```
   nslookup hostx
   ```

   where hostx is a host in your domain.

3. NSLOOKUP will return the IP address of the computer hostx because the information is stored in the DNS database.

4. Exit the command prompt.
To use NSLOOKUP in interactive mode

1. At a command prompt, type `nslookup` then press Enter.

   The > prompt appears.

2. Type `set all` at the > prompt.

   This command lists all of the current values of the NSLOOKUP options.

3. Use the following set commands to change the timeout value to 1 second and the number of retries to 7, as illustrated in Figure 7.9.

   `Set ti=1
   Set ret=7`

4. Use `Set All` to verify that the defaults were changed.

5. Type the names of the other computers, one at a time, at the > prompt. Press Enter after each name.

6. Exit the command prompt.
Lesson Summary

Microsoft DNS is interoperable with other DNS servers. Before installing the DNS Server service, you should make sure that the Windows 2000 server's TCP/IP protocol is configured correctly.

The NSLOOKUP utility is the primary diagnostic tool for DNS. It lets you display resource records on DNS servers.

Lesson 5: Configuring DNS

There are two ways to administrate the Microsoft DNS server: use the DNS Manager or manually edit the DNS configuration files. This lesson reviews the tools used to administer a DNS server.

After this lesson, you will be able to

- Administer a DNS server
- Create a zone file and populate it with resource records

Estimated lesson time: 60 minutes

Configuring DNS Server Properties

The primary tool that you use to manage Windows 2000 DNS servers is the DNS console, illustrated in Figure 7.10. Because the DNS server has no initial information about a user's network, the DNS server installs as a caching-only name server for the Internet. This means that the DNS server contains only information on the Internet root servers. For most DNS server configurations, additional information must be supplied to obtain the preferred operation.

- To open DNS
  1. Click Start, point to Programs, point to Administrative Tools, then click DNS.
Figure 7.10 DNS settings in Microsoft Management Console (MMC)

- **To configure a new DNS server**

  1. Click Start, point to Programs, point to Administrative Tools, then click DNS.
  3. Follow the instructions in the Configure DNS Server Wizard.

   In the Configure DNS Server Wizard, you can create one or more forward lookup zones. The type of zone you create can be

   - **Active Directory-integrated.** Active Directory-integrated DNS enables Active Directory storage and replication of DNS zone databases. Zone data is stored as an Active Directory object and is replicated as part of domain replication.

   - **Standard primary.** Standard primary zones are required to create and manage zones in your DNS name space if you are not using Active Directory.

   - **Standard secondary.** Standard secondary zones help balance the processing load of primary servers and provide fault tolerance.

  4. The next step in the New Zone Wizard is to create a forward or reverse lookup zone. If you select Forward lookup zone, you must provide a name for the new zone and then specify a zone file. If you select Reverse lookup zone, you must provide the network ID or zone name, and then specify a zone file.

  5. Click Finish to close the wizard.

**Manually Configuring DNS**

The DNS server can be configured manually by editing files in the default installation path `\system_root\System32\Dns`. Administration is identical to administration of
traditional DNS. These files can be modified using a text editor, as illustrated in Figure 7.11. The DNS service must then be stopped and restarted.

![Figure 7.11 Editing the CACHE.DNS file](image)

**Adding DNS Domains and Zones**

The first step in configuring the DNS server is to determine the hierarchy for your DNS domains and zones. Once the domain and zone information has been determined, this information must be entered into the DNS configuration using the DNS console.

**Adding Primary or Secondary Zones**

You add primary and secondary zones through the DNS console, as illustrated in Figure 7.12. After you enter your zone information, DNS Manager will construct a default zone file name. If the zone file already exists in the DNS directory, DNS console will automatically import these records.

![Figure 7.12 Creating a new zone with the DNS console](image)

A primary zone stores name-to-address mappings locally. When you configure a
primary zone, you need no information other than the zone name.

Secondary zones obtain name-to-address mappings from a master server by zone transfer. When you configure a secondary zone, you must supply the names for the zone and master name server.

Once all zones have been added to the server, subdomains under the zones can be added. If multiple levels of subdomains are needed, create each successive subdomain. There is a key written to the DNS registry entry for each zone for which the DNS will be authoritative. The keys are located under HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\DNS\Zones.

Each zone has its own key that contains the name of the database file, which indicates whether the DNS server is a primary or secondary name server. For example, for the zone dev.volcano.com, there is the following registry entry:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\DNS\Zones\dev.volcano.com.

**Configuring Zone Properties**

After you have successfully added a zone, you can configure and modify the zone properties (described in Table 7.6).

**Table 7.6 Zone Properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>Configures the zone file in which the resource records are stored and specifies whether this is a primary or secondary name server.</td>
</tr>
<tr>
<td>SOA record</td>
<td>Configures zone transfer information and the name server administrator mailbox.</td>
</tr>
<tr>
<td>Notify</td>
<td>Specifies the secondary servers to be alerted when the primary server data-base changes. Also, additional security can be applied to the name server by specifying that only the listed secondary servers can contact this server.</td>
</tr>
<tr>
<td>WINS lookup</td>
<td>Enables the name server to query WINS to resolve names. A list of WINS servers can be configured in this dialog. The WINS servers can be set on a per-name-server basis by selecting the Settings Only Affect Local Server check box. If this is not selected, secondary servers will also use the configured WINS servers.</td>
</tr>
</tbody>
</table>

**Practice: Configuring a DNS Server**

In this procedure, you configure the DNS server by adding a primary zone. Complete this procedure from the DNS server computer.
Before you continue with the lesson, run the Ch07b.exe demonstration file located in the Media folder on the Supplemental Course Materials CD-ROM that accompanies this book. The file provides an overview of configuring the DNS server service.

- **To add a zone to a server**

  1. Right-click your computer name, then click New Zone.
     
     The New Zone Wizard appears.
  2. Click Next, select Standard Primary, then click Next.
  3. Select Forward Lookup Zone, then click Next.
  4. In the Name box, type `zone1.org` (where `zone1.org` is your zone name).
  5. Click Create A New File With This File Name, then click Next.
     
     `zone1.org.dns` will be the file name (where `zone1.org` is your zone name).
  6. Click Finish to create the new zone.

The Forward Lookup Zones folder now contains your new zone, as illustrated in Figure 7.13.

![Figure 7.13 Zone added to the Forward Lookup Zones folder](image)

**Adding Resource Records**

Once the zones and subdomains are configured, resource records can be added. To create a new host, right-click a zone or subdomain and then click New Host, as illustrated in Figure 7.14. Simply type the host name and click Add Host, and the host record will be created.
To create a record of a different type, right-click a zone or subdomain and then click Other New Records. Next, select which resource record type to create. A dialog box displays various fields specific to record type, as illustrated in Figure 7.15.

Figure 7.15 Selecting a type of record to create

Configuring Reverse Lookup

To find a host name, given the host’s IP address, a reverse lookup zone must be created...
for each network on which hosts in the DNS database reside. Adding a reverse lookup zone is procedurally identical to adding any other type of zone, except for the zone name. For example, if a host has an address of 198.231.25.89, it would be represented in the in-addr.arpa domain as 89.25.231.198.in-addr.arpa. Furthermore, to enable this host to appear to a client who has its IP address, a zone would need to be added to the DNS for 25.231.198.in-addr.arpa. All PTR records for the network 198.231.25.0 would be added to this reverse lookup zone.

**Lesson Summary**

The first step in configuring Windows 2000 DNS server is to determine the hierarchy for your DNS domains and zones. Once the zones and subdomains are configured, resource records can be added. To find a host name, given the host's IP address, a reverse lookup zone must be created for each network on which hosts in the DNS database reside.

[Previous] [Next]

**Review**

Answering the following questions will reinforce key information presented in this chapter. If you are unable to answer a question, review the appropriate lesson and then try the question again. Answers to the questions can be found in Appendix A, "Questions and Answers."

1. Name the three components of the DNS.
2. Describe the difference between primary, secondary, and master name servers.
3. List three reasons to have a secondary name server.
4. Describe the difference between a domain and a zone.
5. Describe the difference between recursive and iterative queries.
7. Describe the purpose of the boot file.

*Answers*