Windows 2000 Server

Chapter 6 - Windows 2000 DNS

Microsoft® Windows® 2000 DNS is compliant with the standard Domain Name System (DNS) as described in the Request for Comments (RFC) documents of the Internet Engineering Task Force (IETF). DNS is the de facto naming system for Internet Protocol (IP)–based networks and the naming service that is used to locate computers on the Internet. Because Windows 2000 DNS is RFC-compliant, it interoperates with most of the other DNS server implementations, such as those DNS servers that use the Berkeley Internet Name Domain (BIND) software. This chapter describes the new features and enhancements of Windows 2000 DNS and explains how to set up and configure some of the features. For more information about DNS-related RFC standards that are supported by Windows 2000, see "Introduction to DNS" in this book.

In This Chapter

Related Information in the Resource Kit

- For more information about TCP/IP, see "Introduction to TCP/IP" in this book.
- For more information about the Windows Internet Name Service, see "Windows Internet Name Service" in this book.
- For information about Domain Name System concepts, see "Introduction to DNS" in this book.
- For more information about Active Directory, see "Active Directory Logical Structure" in *Microsoft® Windows® 2000 Server Resource Kit Distributed Systems Guide*.

Introduction to the Windows 2000 Implementation of DNS

The Windows 2000 DNS server and resolver have several new features and improvements over those of Microsoft® Windows NT® version 4.0. This chapter describes the following features:

Support for Active Directory as a Locator Service for Domain Controllers

DNS is required for support of Active Directory. You can also use another DNS server implementation solution to support Active Directory deployment.

Integration with Active Directory

You can integrate DNS zones into Active Directory, providing increased fault tolerance and security. Every Active Directory-integrated zone is replicated among all domain controllers within the Active Directory domain. All DNS servers running on these domain controllers can act as primary servers for the zone, accepting dynamic updates. Also, Active Directory replicates on a per-property basis, propagating only relevant changes.

Support for Dynamic Updates

The DNS service allows client computers to dynamically update their resource records in DNS. This improves DNS administration by reducing the time needed to manually manage zone records. The dynamic update feature can be used in conjunction with Dynamic Host Configuration Protocol (DHCP) to dynamically update resource records when a computer's IP address is released and renewed. Computers that run Windows 2000 can send dynamic updates.

Support for Aging and Scavenging of Records

The DNS service is capable of aging and scavenging records. When enabled, this feature can prevent stale records from remaining in DNS.

Support for Secure Dynamic Updates in Active Directory–Integrated Zones

You can configure Active Directory–integrated zones for secure dynamic update. With secure dynamic update, only authorized users can make changes to a zone or record.

Improved Ease of Administration

The DNS console offers an improved graphical user interface (GUI) for managing the DNS service. Also, Windows 2000 Server provides several new configuration wizards and other tools to help you manage and support DNS servers and clients on your network.

Administration from the Command Prompt

You can use the command-line tool Dnscmd.exe to perform most of the tasks that you can perform from the DNS console. For example, you can create, delete, and view zones and records; reset server and zone properties; and perform routine administration operations such as updating the zone, reloading the zone, refreshing the zone, writing the zone back to a file or Active Directory, pausing and resuming the zone, clearing the cache, stopping and starting the DNS service, and viewing statistics.

You can also use Dnscmd.exe to write scripts and for remote administration. For more information about Dnscmd.exe, see Windows 2000 Support Tools Help. For information about installing and using the Windows 2000 Support Tools and Support Tools Help, see the file Sreadme.doc in the directory \Support\Tools on the Windows 2000 operating system CD.

Enhanced Name Resolution

The Windows 2000 resolver generally tries to resolve names with DNS before trying to do so with Network Basic Input/Output System (NetBIOS). Also, it can query different servers based on the adapters to which they are assigned.

Enhanced Caching and Negative Caching

You can now view and flush the resolver cache by using the command-line tool Ipconfig, and you can flush the server cache from within the DNS console. Also, the resolver performs *negative caching*, which stores the information that a name or type of record does not exist. Negative caching reduces lookup time when the user queries for a name that the resolver has already determined does not exist. For more information about caching, see "Windows 2000 Resolver" later in this chapter.

Additional Client Enhancements

The cache can be preloaded with Hosts file entries. Also, the resolver server list can be dynamically reordered to prioritize responsive DNS servers.

Support for a Pure DNS Environment

If all of the computers on your network are running Windows 2000, you do not need any WINS servers. Even in a mixed environment,

you do not need to configure WINS on your Windows 2000–based clients if you have configured WINS lookup. By using WINS lookup, you can direct DNS to query WINS for name resolution, so that DNS clients can look up the names and IP addresses of WINS clients.

Interoperability with Other DNS Server Implementations

Because the Windows 2000 DNS server is RFC-compliant, it interoperates with other DNS server implementations, such as BIND.

Integration with Other Network Services

The Windows 2000 DNS server is integrated with DHCP and WINS.

Incremental Zone Transfer

In addition to performing full zone transfers (sending a copy of the entire zone), the DNS server can now send and receive incremental zone transfers, in which only changes to the zone are transferred. This can reduce the amount of time and bandwidth required for zone transfers.

Support for New Resource Record Types

Windows 2000 includes support for two new record types: the SRV resource record, which is used by computers to locate domain controllers, and the ATMA resource record.

Naming Hosts and Domains

In Windows NT 4.0 and earlier, a computer is identified primarily by a NetBIOS name — it is by this name that the computer is known on the network. In Windows 2000, a computer is identified primarily by its full computer name, which is a DNS fully qualified domain name (FQDN). The same computer could be identified by more than one FQDN. However, only the FQDN that is a concatenation of the host name and the primary DNS suffix is the full computer name. In this chapter, the first label of the full computer name is known as the *host name*, and the remaining labels form a primary DNS suffix.

By default, the *primary DNS suffix* of a computer that is running Windows 2000 is set to the DNS name of the Active Directory domain to which the computer is joined. The primary DNS suffix can also be specified by Group Policy, discussed later in this section.

Note You can set and view the FQDN from the **Network Identification** tab of the **System Properties** dialog box, which you can go to by right-clicking **My Computer**, and then clicking **Properties**. To change the host name, click **Properties** and then to change the primary DNS suffix, click **More**.

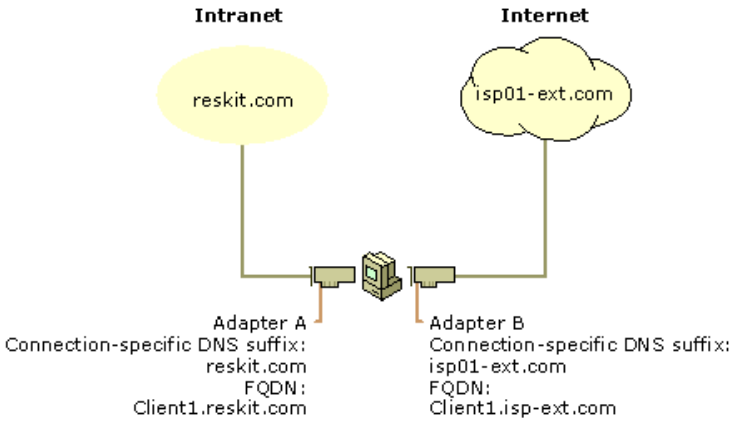
Suppose that you have a WINS client named Client1. The name "Client1" will be the computer's NetBIOS name. Next, suppose that you replace WINS with DNS on your network and make Client1 a DNS client in the domain eu.reskit.com. The name Client1 is now also the computer's host name, and it is by default concatenated with the primary DNS suffix eu.reskit.com to make the FQDN Client1.eu.reskit.com.

The NetBIOS name is derived from the host name, but the two names might not be identical. The NetBIOS name is a 16-byte string that uniquely identifies a computer or service for network communication. It is used by all the Windows 2000 network services to uniquely identify themselves. If the DNS host name is 15 or fewer bytes, the NetBIOS name is the host name plus enough spaces to form a 15-byte name, followed by a unique identifier, the sixteenth byte, that specifies the network service. If the DNS host name is longer than 15 bytes, then by default, the NetBIOS name is the host name, truncated to 15 bytes, plus the service identifier. If you try to create two DNS host names and the first 15 bytes are the same, you are prompted to enter a new name for NetBIOS.

Note Because host names are encoded in UTF-8 format, they do not necessarily have only 1 byte per character. ASCII characters are 1 byte each, but the size of extended characters is more than 1 byte.

Windows 2000 also allows each adapter to have its own DNS suffix, which is known as a connection-specific DNS suffix. The connection-specific DNS suffix is usually assigned by a DHCP server that leases an IP address to the adapter. On computers that are running Windows 2000, in addition, an administrator can assign a connection-specific DNS suffix to statically configured adapters.

Depending on the configuration, the connection-specific DNS suffix can be appended to the host name to create an FQDN that is registered in DNS. For example, suppose that the computer Client1 has the primary DNS suffix reskit.com, and Client1 is connected to both the Internet and the corporate intranet. For each connection, you can specify a connection-specific DNS suffix. For the connection to the corporate intranet, you specify the name reskit.com, and the FQDN is then Client1.reskit.com. For the connection to the Internet, you specify the name isp01-ext.com, and the FQDN is then Client1.isp01-ext.com. Figure 6.1 shows this configuration.



If your browser does not support inline frames, click here to view on a separate page.

Figure 6.1 Connection-Specific Domain Names

You can specify connection-specific DNS suffixes for statically configured adapters and adapters configured by DHCP on the **DNS** tab in the **Advanced TCP/IP Settings** dialog box. In that dialog box, you can also specify whether the client uses its connection-specific DNS suffix in addition to its primary DNS suffix when it registers its FQDN. For more information about configuring DHCP clients for DNS, see "Dynamic Update" later in this chapter.

Caution If you have any multihomed dynamic update clients and at least one adapter is using DHCP, configure the DHCP server to update resource records according to the request of the client. For more information about how to configure DHCP servers to update resource records, see "Dynamic Update" later in this chapter. If the DHCP server is configured to register both A and PTR resource records, the DHCP server replaces all A resource records for the name it attempts to register. As a result, A resource records that correspond to the IP addresses for the computer's other addresses might be deleted.

Table 6.1 summarizes the differences between each kind of name using the example FQDN client1.reskit.com.

Table 6.1 DNS and NetBIOS Names

Name Type	Description
NetBIOS name	The NetBIOS name is used to uniquely identify the NetBIOS services listening on the first IP address that is bound to an adapter. This unique NetBIOS name is resolved to the IP address of the server through broadcast, WINS, or the LMHosts file. By default, it is the same as the host name up to 15 bytes, plus any spaces necessary to make the name 15 bytes long, plus the service identifier. The NetBIOS name is also known as a *NetBIOS* computer name. For example, a NetBIOS name might be Client1.
Host name	The term *host name* can mean either the FQDN or the first label of an FQDN. In this chapter, *host name* refers to the first label of an FQDN. For example, the first label of the FQDN client1.reskit.com is client1.
Primary DNS suffix	Every Windows 2000–based computer can be assigned a primary DNS suffix to be used in name resolution and name registration. The primary DNS suffix is specified on the **Network Identification** tab of the properties page for **My Computer**. The primary DNS suffix is also known as the primary domain name and the domain name. For example, the FQDN client1.reskit.com has the primary DNS suffix reskit.com.
Connection-specific DNS suffix	The connection-specific DNS suffix is a DNS suffix that is assigned to an adapter. The connection-specific DNS suffix is also known as an *adapter-specific DNS suffix*. For example, a connection-specific DNS suffix might be acquired01-ext.com.
Full computer name	The full computer name is a type of FQDN. The same computer could be identified by more than one FQDN. However, only the FQDN that is a concatenation of the host name and the primary DNS suffix is the full computer name.
Fully qualified domain name	The FQDN is a DNS name that uniquely identifies the computer on the network. By default, it is a concatenation of the host name, the primary DNS suffix, and a period. For example, an FQDN might be client1.reskit.com.

Complying With Name Restrictions for Hosts and Domains

Different DNS implementations impose different character and length restrictions. Table 6.2 shows the restrictions for each implementation.

Table 6.2 Name Restrictions

Restriction	Standard DNS (Including Windows NT 4.0)	DNS in Windows 2000	NetBIOS
Characters	Supports RFC 1123, which permits "A" to "Z", "a" to "z", "0" to "9", and the hyphen (-).	Several different configurations are possible, as described at the end of this section.	Unicode characters, numbers, white space, symbols: ! @ # $ % ^ & ') (. - _ { } ~
Fully qualified domain name length	63 bytes per label and 255 bytes for an FQDN	63 bytes per label and 255 bytes for an FQDN; domain controllers are limited to 155 bytes for an FQDN.	15 bytes

Note Although you can create long, complex DNS names, it is recommended that you create shorter, user-friendly names.

According to RFC 1123, the only characters that can be used in DNS labels are "A" to "Z", "a" to "z", "0" to "9", and the hyphen ("-"). (The period [.] is also used in DNS names, but only between DNS labels and at the end of an FQDN.) Many DNS servers, including Windows NT 4.0–based DNS servers, follow RFC 1123.

However, adherence to RFC 1123 can present a problem on Windows 2000 networks that still use NetBIOS names. NetBIOS names can use additional characters, and it can be time consuming to convert all the NetBIOS names to standard DNS names.

To simplify the migration process to Windows 2000 from Windows NT 4.0, Windows 2000 supports a wider character set. RFC 2181, "Clarifications to the DNS Specification," enlarges the character set allowed in DNS names. It states that a DNS label can be any binary string, and it does not necessarily need to be interpreted as ASCII. Based on this definition, Microsoft has proposed that the DNS name specification be readjusted to accommodate a larger character set: UTF-8 character encoding, as described in RFC 2044. UTF-8 character encoding is a superset of ASCII and a translation of the UCS-2 (also known as Unicode) character encoding. The UTF-8 character set includes characters from most of the world's written languages; this enables a far greater range of possible names. The Windows 2000 DNS service includes support for UTF-8 character encoding.

However, before using additional characters, consider the following issues:

- Some third-party resolver software supports only the characters listed in RFC 1123. If you have any third-party resolver software, that software is probably not able to look up computers with names that have non-standard characters.
- A DNS server that does not support UTF-8 encoding might accept a zone transfer of a zone containing UTF-8 names, but it cannot write back those names to a zone file or reload those names from a zone file. Therefore, you must not transfer a zone that contains UTF-8 characters to a DNS server that does not support them.

You can configure the Windows 2000 DNS server to allow or disallow the use of UTF-8 characters on your Windows 2000 server. You can do so on a per-server basis from within the DNS console. From the **Advanced** tab of the server properties page, set **Name checking** to one of the following:

- *Strict RFC (ANSI)*. Allows "A" to "Z", "a" to "z", the hyphen (-), the asterisk (*) as a first label; and the underscore (_) as the first character in a label.
- *Non RFC (ANSI)*. Allows all characters allowed when you select **Strict RFC (ANSI)**, and allows the underscore (_) anywhere in a name.
- *Multibyte (UTF-8)*. Allows all characters allowed when you select **Non RFC (ANSI)**, and allows UTF-8 characters.
- *Any character*. Allows any character, including UTF-8 characters.

Note If you enter a DNS name that includes UTF-8 or underscore characters that are not listed in RFC 1123 when you are modifying a host name or DNS suffix or creating an Active Directory domain, a warning message appears explaining that some DNS server implementations might not support these characters.

Using Group Policy to Specify a DNS Suffix

When a Group Policy exists, the suffix set in the Group Policy supersedes the local primary DNS suffix, which by default is the same as the Active Directory domain name. Users can still enter a suffix in the **System Properties** dialog box, but the suffix is not used unless the Group Policy is disabled or unspecified.

If you make the primary DNS suffix of the computer different from the Active Directory domain name, however, you must perform additional configuration in order to enable the modified full computer name to be registered in the DNS host name attribute and the Service Principal Name attribute for the computer object in Active Directory.

By default, the name registered in those attributes must have the following syntax:

<NetBIOS name>.<Active Directory domain name>

where *NetBIOS name* is the NetBIOS name of the computer and *Active Directory domain name* is the DNS name of the Active Directory domain. To enable registration of the modified full computer name, you must modify the access control list (ACL) for the appropriate domain by following the steps in the following procedure. You must also perform this procedure if any computers joined to the domain have host names of more than 15 bytes.

To modify the ACL to enable registration of the full computer name

1. Click **Start**, highlight **Programs**, highlight **Administrative Tools**, and then click **Active Directory Users and Computers**.
2. In the **View** menu, click **Advanced Features**.
3. Right-click the domain you want to modify, and then click **Properties**.
4. Click the **Security** tab.
5. Click **Add**, click **SELF**, click **ADD**, and then click **OK**. This adds the SELF group to the ACL.
6. Click the **Advanced** button.
7. Click **SELF** and then click **View/Edit**.
8. Click the **Properties** tab.
9. In the **Apply onto** box, click **Computer objects**.
10. In the **Permissions** box, check **Allow** next to **Write dNSHostName**, and then click **OK** until you have closed the **Active Directory Users and Computers** dialog box.

Caution If you modify the ACL to enable registration of the modified full computer name, any computer in the domain can register itself under a different name.

Windows 2000 Resolver

Windows 2000 DNS includes a *caching resolver* service. The caching resolver reduces DNS network traffic and speeds name resolution by providing a local cache for DNS queries. For troubleshooting purposes, this service can be viewed, stopped, and started like any other Windows 2000 service by using the **Component Services** console; but the caching resolver is enabled by default.

The Windows 2000 resolver performs the following tasks:

- Name resolution.
- General caching of queries.
- Negative caching.
- Keeps track of transient (Plug and Play) network adapters and their IP configurations.
- Keeps track of connection-specific domain names.
- When a server fails to respond to a query, the resolver ceases to query that server for a certain amount of time.
- When the resolver receives multiple A resource records from a DNS server, it prioritizes them based on their IP address.

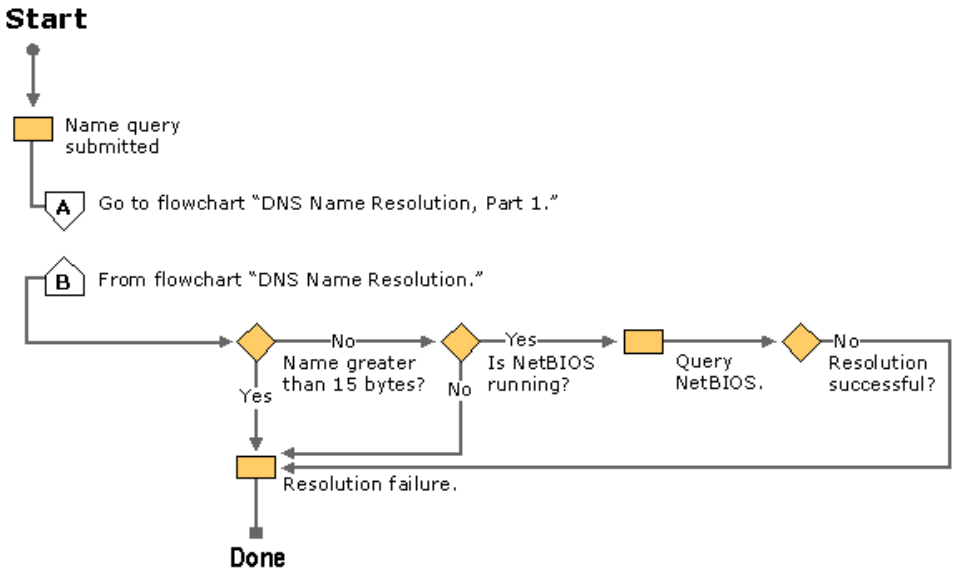
Name Resolution

Name resolution in Windows 2000 differs significantly from name resolution in Windows NT 4.0. In Windows NT 4.0, the resolver generally tried NetBIOS name resolution first and then DNS name resolution. In Windows 2000, however, the resolver generally tries DNS name resolution first, and then it tries NetBIOS name resolution. Windows 2000 also includes improvements for multihomed computers.

When the GetHostByName API is used, the Windows 2000 resolver first submits the name query to DNS. If DNS name resolution fails, the resolver checks whether the name is longer than 15 bytes. If it is longer, resolution fails. If not, the resolver then checks whether NetBIOS is running. If it is not running, resolution fails. If it is running, the resolver then tries NetBIOS name resolution. For information about NetBIOS name resolution and flowcharts for NetBIOS name resolution, see "Windows 2000 TCP/IP" in this book.

Figure 6.2 shows an overview of the process.

Note The flowchart in Figure 6.2 directs you to other flowcharts in other figures. To locate the correct flow chart, see the figure captions.



If your browser does not support inline frames, click here to view on a separate page.

Figure 6.2 Overview of Name Resolution

DNS Name Resolution

When DNS name resolution begins, the resolver first checks what kind of name was submitted. Three types of names can be submitted:

- Fully qualified domain names

  These names are terminated with a period. For example:

  host.reskit.com.

- Single-label, unqualified domain names

  These names contain no periods. For example:

  host

- Multiple-label, unqualified domain names

  These names contain one or more periods but are not terminated with a period. For example:

  host.reskit.com

  – Or –

  host.reskit

When a user enters an FQDN, the resolver queries DNS using that name. Likewise, when a user enters a multiple-label, unqualified (not terminated with a period) name, the DNS resolver adds a terminating period and queries DNS using that name.

However, if the user enters a multiple-label, unqualified name and it fails to resolve as an FQDN, or if the user enters a single-label, unqualified name, the resolver systematically appends different DNS suffixes to the name that the user entered, adding periods to make them FQDNs, and resubmitting them to DNS.

If the user has not entered a domain suffix search list, the resolver appends the following names:

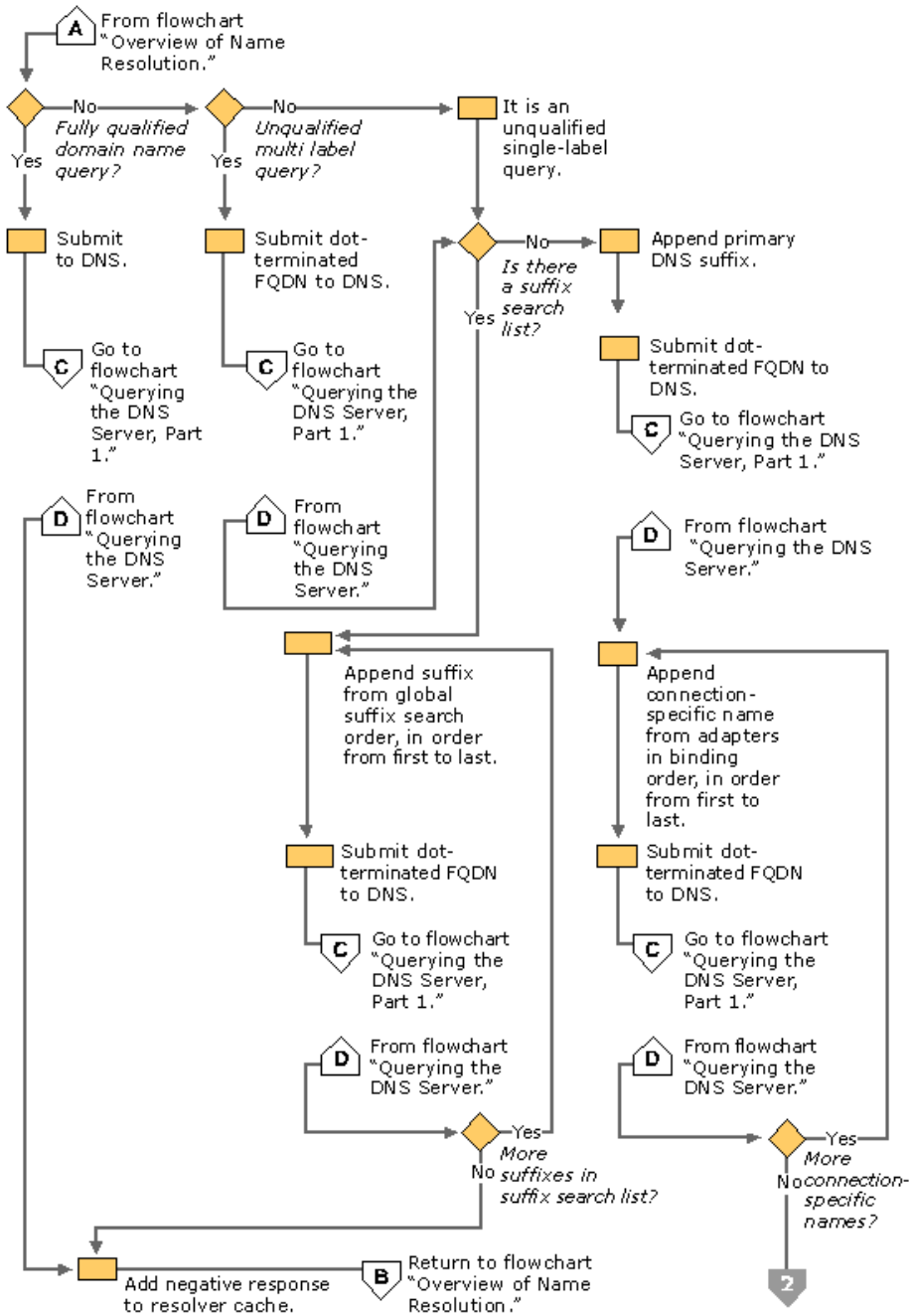
1. The primary DNS suffix, which is specified on the **Network Identification** tab of the **System Properties** dialog box in the properties for **My Computer**. Click the **Properties** button, and then click **More**.
2. If resolution is not successful, the resolver appends each connection-specific DNS suffix. This suffix can be dynamically assigned by the DHCP server. You can also specify suffixes on the **DNS** tab in the **Advanced TCP/IP Settings** dialog box for each connection. You open the **Advanced TCP/IP Settings** dialog box by right-clicking the connection and then clicking **Properties** to reach the properties from the connection, then double-clicking **Internet Protocol (TCP/IP)** to reach the **Internet Protocol (TCP/IP) Properties** dialog box, and then clicking **Advanced**.

If resolution is still not successful, the resolver devolves the FQDN by appending the parent suffix of the primary DNS suffix name, and the parent of that suffix, and so on, until only two labels are left. For example, if the user enters the name **client** and the primary DNS suffix is eu.reskit.com, the resolver will try client.eu.reskit.com and then client.reskit.com.

On the other hand, if the user has entered a domain suffix search list on the **DNS** tab in the **Advanced TCP/IP Settings** dialog box in the properties for the network connection, both the primary DNS suffix and the connection-specific domain name are ignored, and neither is appended to the host name before the FQDN is submitted to DNS. Instead, the resolver appends each suffix from the search list in order and submits it to the DNS server until it finds a match or reaches the end of the list.

Figures 6.3 and 6.4 show how FQDNs are formed. Figure 6.5 shows what happens when a name is submitted to DNS.

Note The flowcharts in Figures 6.3 and 6.4 direct you to other flowcharts in other figures. To locate the correct flow chart, see the figure captions.



If your browser does not support inline frames, click here to view on a separate page.

Figure 6.3 DNS Name Resolution, Part 1

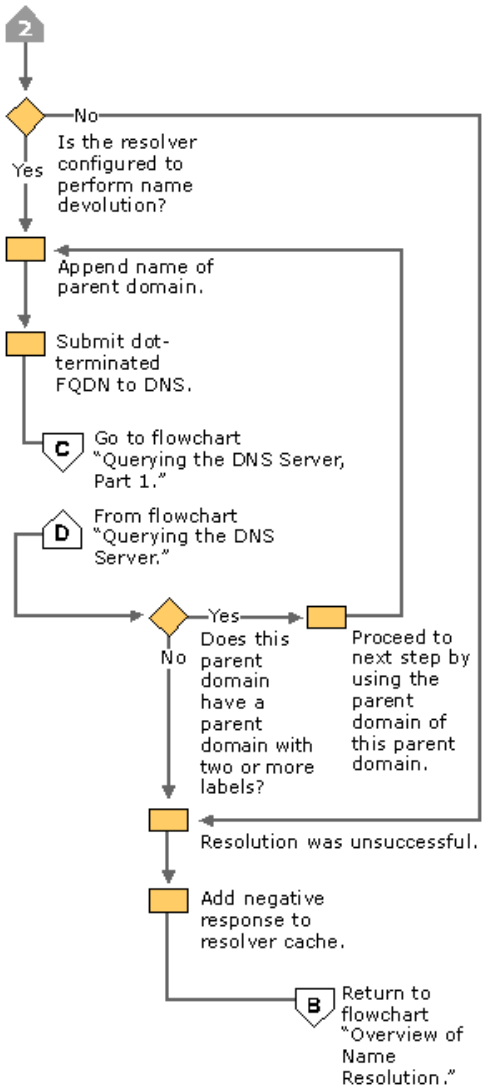


Figure 6.4 DNS Name Resolution, Part 2

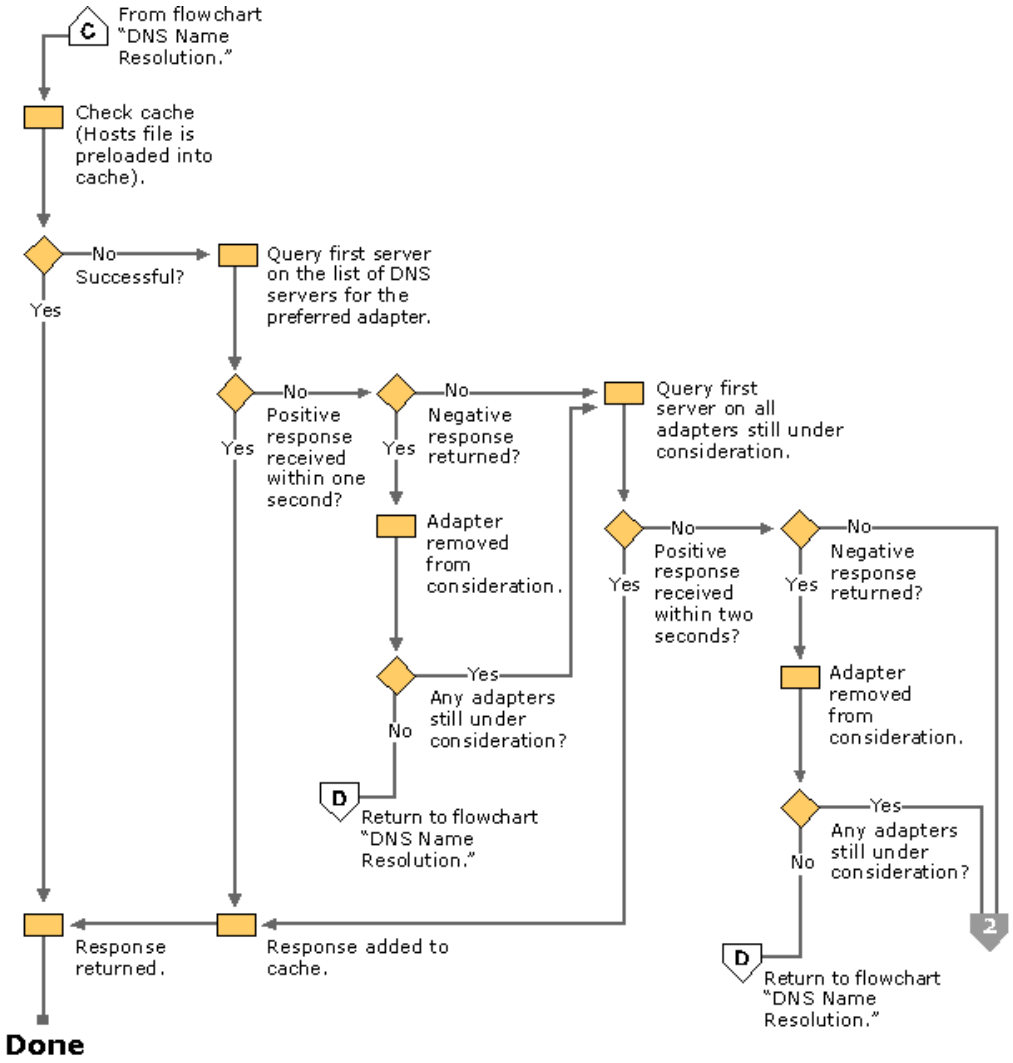
DNS Queries

When a name is submitted to DNS, if the resolver is caching names, the resolver first checks the cache. If the name is in the cache, the data is returned to the user. If the name is not in the cache, the resolver queries the DNS servers that are listed in the TCP/IP properties for each adapter.

The resolver can query through all adapters in the computer, including remote access adapters. In Windows NT 4.0, the resolver queried all servers through all adapters. In Windows 2000, however, you can specify a list of DNS servers to query for each adapter.

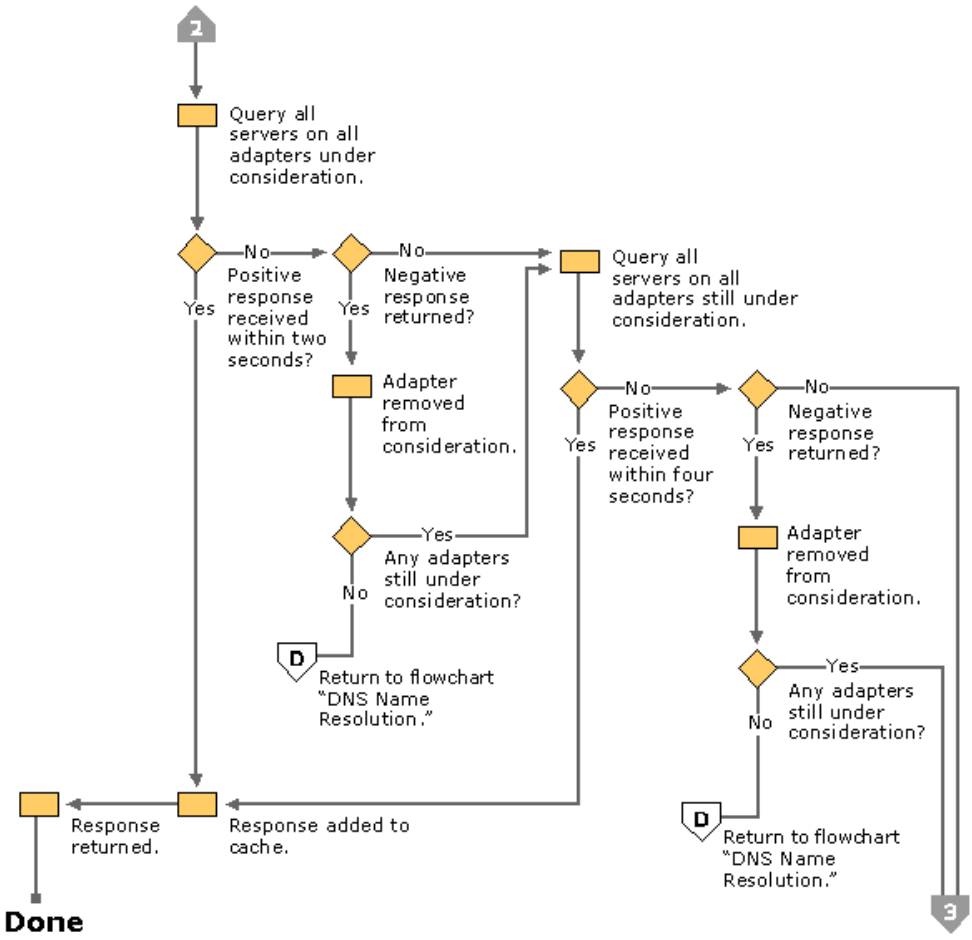
Figures 6.5, 6.6, and 6.7 illustrate the process by which the resolver queries the servers on each adapter.

Note The flowcharts in Figures 6.5, 6.6, and 6.7 direct you to other flowcharts in other figures. To locate the correct flow chart, see the figure captions.



If your browser does not support inline frames, click here to view on a separate page.

Figure 6.5 Querying the DNS Server, Part 1



If your browser does not support inline frames, click here to view on a separate page.

Figure 6.6 Querying the DNS Server, Part 2

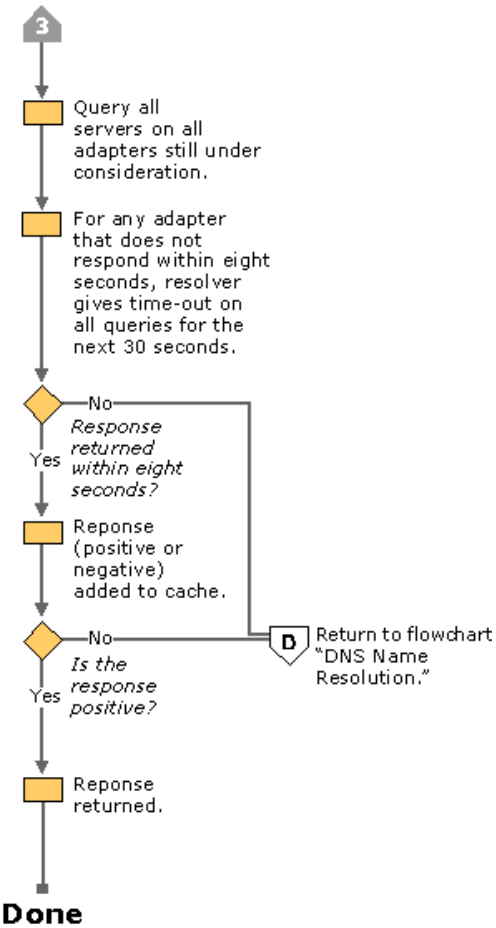


Figure 6.7 Querying the DNS Server, Part 3

The resolver queries the DNS servers in the following order:

1. The resolver sends the query to the first server on the preferred adapter's list of DNS servers and waits for one second for a response.
2. If the resolver does not receive a response from the first server within one second, it sends the query to the first DNS servers on all adapters that are still under consideration and waits two seconds for a response.
3. If the resolver does not receive a response from any server within two seconds, the resolver sends the query to all DNS servers on all adapters that are still under consideration and waits another two seconds for a response.
4. If the resolver still does not receive a response from any server, it sends the query to all DNS servers on all adapters that are still under consideration and waits four seconds for a response.
5. If it still does not receive a response from any server, the resolver sends the query to all DNS servers on all adapters that are still under consideration and waits eight seconds for a response.

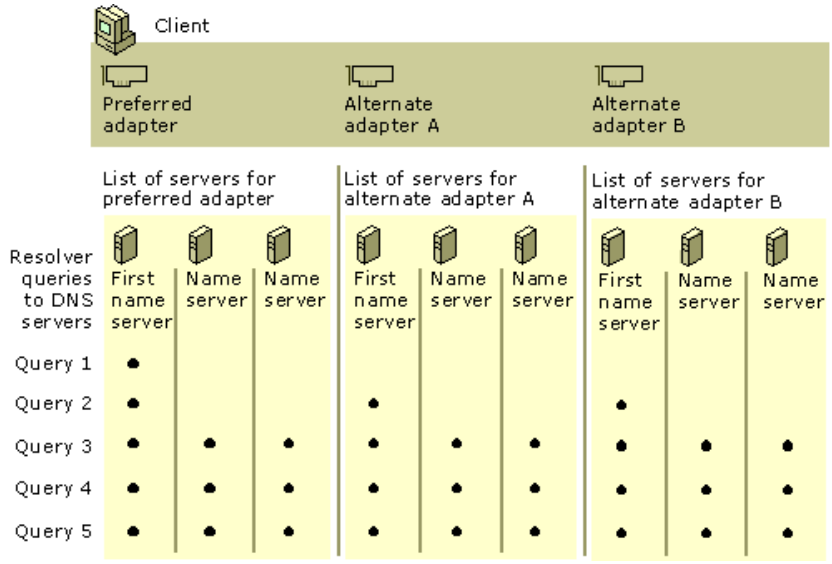
If the resolver receives a positive response, it stops querying for the name, adds the response to the cache and returns the response to the client.

If the resolver has not received a response from any server by the end of the eight-second time period, the resolver responds with a time-out. Also, if it has not received a response from any server on a specified adapter, then for the next 30 seconds, the resolver responds to all queries destined for servers on that adapter with a time-out and does not query those servers. This time-out is sent only by computers running Windows 2000 Professional.

If at any point the resolver receives a negative response from a server, it removes every server on that adapter from consideration during this search. For example, if in step 2, the first server on Alternate Adapter A gave a negative response, the resolver would not send the query to any other server on the list for Alternate Adapter A.

The resolver keeps track of which servers answer queries more quickly, and it might move servers up or down on the list based on how quickly they reply to queries.

Figure 6.8 shows how the resolver queries each server on each adapter.



If your browser does not support inline frames, click here to view on a separate page.

Figure 6.8 Multihomed Name Resolution

Configuring Query Settings

The resolver attaches DNS suffixes to a name that you enter in a query if either of the following conditions is true:

- Name is a single-label unqualified name.
- Name is a multiple-label unqualified name, and the resolver did not resolve it as an FQDN.

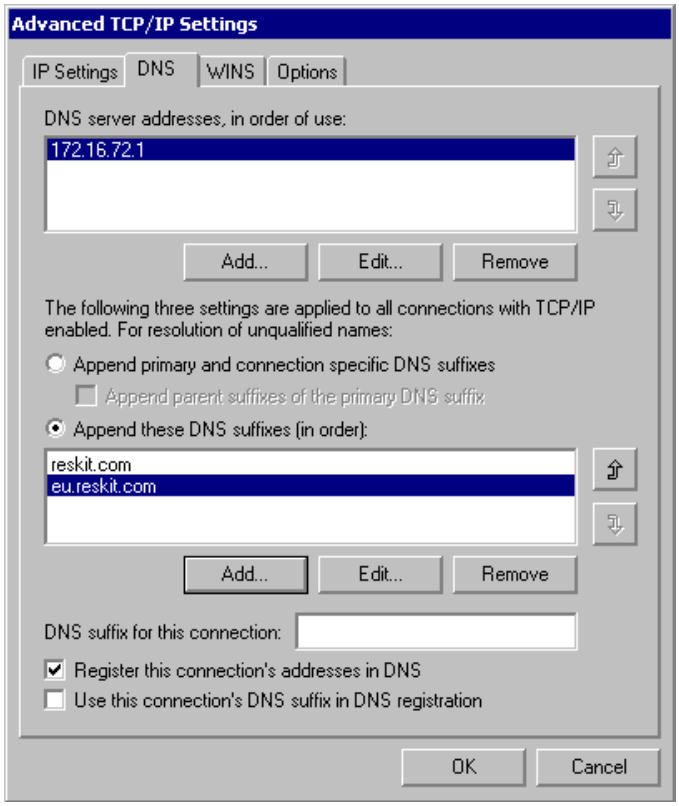
Note For information about what happens when FQDNs and multiple-label domain names are submitted to DNS, see "DNS Name Resolution" earlier in this chapter.

You can configure which suffixes are added to queries from within the **Advanced TCP/IP Settings** dialog box.

To view the Advanced TCP/IP Settings dialog box

1. Right-click **My Network Places**, and then click **Properties**.
2. Right-click the connection that you want to view, and then click **Properties**.
3. Click **Internet Protocol (TCP/IP)**, and then click **Properties**.
4. Click **Advanced**, and then click the **DNS** tab.

Figure 6.9 shows the **Advanced TCP/IP Settings** dialog box.



If your browser does not support inline frames, click here to view on a separate page.

Figure 6.9 DNS Tab in the Advanced TCP/IP Settings Dialog Box

The box labeled **Append these DNS suffixes (in order)** lets you specify a list of DNS suffixes to try, called a DNS suffix search list. If you enter a DNS suffix search list, the resolver adds those DNS suffixes in order and does not try any other domain names. For example, if the **Append these DNS suffixes (in order)** box includes the names listed in Figure 6.9 and you submit the unqualified, single-label query "coffee," the resolver queries in order for the following FQDNs:

coffee.reskit.com.

coffee.eu.reskit.com.

If you do not enter a DNS suffix search list, the resolver first appends the primary DNS suffix, which you specify on the **Network Identification** tab of the **System Properties** dialog box. For example, if your primary DNS suffix is fareast.isp01-ext.com, the resolver queries for the following FQDN:

coffee.fareast.isp01-ext.com.

Next, if that query fails and if a connection-specific DNS suffix is specified in the **DNS suffix for this connection** box or assigned by the DHCP server, the resolver appends that suffix. For example, if you entered the name noam.reskit.com in the **DNS suffix for this connection** box and then queried for the unqualified, single-label name "coffee," the resolver queries for the following FQDN:

coffee.noam.reskit.com.

Next, if you select the check box **Append parent suffixes of the primary DNS suffix**, the resolver performs name devolution on the primary DNS suffix. It strips off the leftmost label and tries the resulting domain name until only two labels remain. For example, if your primary DNS suffix is mfg.fareast.isp01-ext.com, and you selected the check box **Append parent suffixes of the primary DNS suffix** and then queried for the unqualified, single-label name "coffee," the resolver queries in order the following FQDNs:

coffee.fareast.isp01-ext.com.

coffee.isp01-ext.com.

You can disable name devolution by clearing the check box **Append parent suffixes of the primary DNS suffix**.

Configuring Caching and Negative Caching

When the Windows 2000 resolver receives a positive or negative response to a query, it adds that positive or negative response to its cache. The resolver always checks the cache before querying any DNS servers, so if a name is in the cache, the resolver uses the name from the cache rather than querying a server. This expedites queries and decreases network traffic for DNS queries.

You can use the command-line tool Ipconfig to view and flush the cache.

To view the resolver cache

- At the command prompt, type the following and then press ENTER:

  ipconfig /displaydns

Ipconfig displays the contents of the DNS resolver cache, including names that are preloaded from the Hosts file and any recently

queried names resolved by the system.

After a certain amount of time, specified by the time to live (TTL) associated with the name, the resolver discards the name from the cache. You can view and change the TTL associated with the record from within the DNS console.

To view the TTL for a record

1. In the DNS console, point to **View** and click **Advanced** to select Advanced View.

   This step is not necessary to view the TTL for a start of authority (SOA) record.

2. Right-click the record, and click **Properties**.

You can also flush the cache manually. After you flush the cache, the computer needs to query DNS servers.

To flush the cache manually by using Ipconfig

- At the command prompt, type the following and then press ENTER:

  ipconfig /flushdns

The local Hosts file is preloaded into the resolver's cache and reloaded into the cache whenever the local Hosts file is updated.

Note The resolver cache and server cache are maintained separately. For information about the server cache, see Windows 2000 Server Help.

The length of time for which a positive or negative response is cached on a DNS client depends on the values in the following registry key:

HKEY_LOCAL_MACHINE \SYSTEM \CurrentControlSet \Services \DNSCache \Parameters

Caution Do not use a registry editor to edit the registry directly unless you have no alternative. The registry editors bypass the standard safeguards provided by administrative tools. These safeguards prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your system. Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. To configure or customize Windows 2000, use the programs in Control Panel or Microsoft Management Console (MMC) whenever possible.

Positive responses are cached for the number of seconds specified in the query response that the resolver received, but never for longer than the value of the **MaxCacheEntryTtlLimit** (DWORD) registry entry. The default value is 86,400 seconds.

Windows 2000 supports negative caching, as specified in RFC 2308 with some modifications in the resolver cache. In the resolver cache, negative responses are cached for the number of seconds specified in the **NegativeCacheTime** value (DWORD). The default data is 300 seconds. If you do not want negative responses to be cached at all, set the value of **NegativeCacheTime** to 0.

Note The Windows 2000 DNS server caches negative responses according to the minimum TTL in the SOA record. However, it cannot be less than one minute or greater than 15 minutes. Thus, if the minimum TTL in the SOA record is 20 minutes, the negative response is cached for only 15 minutes. You can use the DNS console or Dnscmd.exe to change the minimum TTL.

If all DNS servers on an adapter are queried and none of them reply, either positively or negatively, all subsequent name queries to any server listed on that adapter fail instantly and continue to fail for a default of 30 seconds. This feature decreases network traffic. It is available only on Windows 2000 Professional.

Configuring Subnet Prioritization

If the resolver receives multiple A resource records from a DNS server, and some have IP addresses from networks to which the computer is directly connected to, the resolver orders those resource records first. This reduces network traffic across subnets by forcing computers to connect to network resources that are closer to them.

For example, suppose that you have three Web servers that all host the Web page for www.reskit.com, and they are all located on different subnets. On the name server, you can create the following resource records:

```
www.reskit.com. IN A 172.16.64.11
www.reskit.com. IN A 172.17.64.22
www.reskit.com. IN A 172.18.64.33
```

When users query for www.reskit.com, the resolver puts first in the list IP addresses from networks to which the computer is directly connected. For example, if a user with the IP address 172.17.64.93 queries for www.reskit.com, the resolver returns the resource records in the following order:

```
www.reskit.com. IN A 172.17.64.11
www.reskit.com. IN A 172.16.64.22
www.reskit.com. IN A 172.18.64.33
```

Subnet prioritization prevents the resolver from using the round robin feature, defined in RFC 1794. Using the round robin feature, the server rotates the order of resource record data returned in a query answer in which multiple resource records of the same type exist for a queried DNS domain name. Thus, in the example described earlier, if a user queries for www.reskit.com, the name server replies to the first client request by ordering the addresses as the following:

```
172.16.64.11
172.17.64.22
172.18.64.33
```

The name server replies to the second client response with the addresses ordered as follows:

```
172.17.64.22
172.18.64.33
172.16.64.11
```

If clients are configured to use the first IP address in the list that they receive, different clients use different IP addresses; so the load is balanced among multiple network resources that have the same name. However, if the resolvers are configured for subnet prioritization, the resolvers reorder the list to favor IP addresses from networks to which they were directly connected; so the effectiveness of the round robin feature is reduced.

Although subnet prioritization does reduce network traffic across subnets, in some cases you might prefer to have the round robin feature work as described in RFC 1794. If so, you can disable the subnet prioritization feature on your clients by adding the **PrioritizeRecordData** registry entry with a value of 0 (REG_DWORD) to the following registry subkey:

HKEY_LOCAL_MACHINE \SYSTEM \CurrentControlSet \Services \DnsCache \Parameters

Caution Do not use a registry editor to edit the registry directly unless you have no alternative. The registry editors bypass the standard safeguards provided by administrative tools. These safeguards prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your system. Editing the registry directly can have serious, unexpected consequences that

can prevent the system from starting and require that you reinstall Windows 2000. To configure or customize Windows 2000, use the programs in Control Panel or Microsoft Management Console (MMC) whenever possible.

Configuring Subnet Prioritization on the Server

In addition to configuring the resolver to perform subnet prioritization for records that it receives, you can configure the server to do the same for records that it sends. How the server behaves depends on the setting of the **Enable round robin** option on the **Advanced** tab of the server **Properties** dialog box in the DNS snap-in, and the value of the **LocalNetPriority** (REG_DWORD) registry entry in the following registry subkey:

HKEY_LOCAL_MACHINE \SYSTEM \CurrentControlSet \Services \DNS \Parameters

You can also change the round robin setting from the registry; however, do so from the DNS snap-in instead.

If **Enable round robin** is selected (the default) and the value of **LocalNetPriority** is 1, the server rotates among the A resource records that it returns in the order of their similarity to the IP address of the querying client.If **Enable round robin** is deselected and the value of **LocalNetPriority** is 1, the server returns the records in local net priority order. It does not rotate among available addresses.

If **Enable round robin** is selected and the value of **LocalNetPriority** is 0 (the default), the server rotates among the available records in the order in which the records were added to the database.If **Enable round robin** is deselected and the value of **LocalNetPriority** is 0 (the default), the server returns the records in the order in which they were added to the database. The server does not attempt to sort them or rotate the records it returns.

Caution Do not use a registry editor to edit the registry directly unless you have no alternative. The registry editors bypass the standard safeguards provided by administrative tools. These safeguards prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your system. Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. To configure or customize Windows 2000, use the programs in Control Panel or Microsoft Management Console (MMC) whenever possible.

Preventing the Resolver from Accepting Responses from Non-Queried Servers

By default, the resolver accepts responses from the servers that it did not query. This feature speeds performance but can be a security risk.

Caution Do not use a registry editor to edit the registry directly unless you have no alternative. The registry editors bypass the standard safeguards provided by administrative tools. These safeguards prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your system. Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. To configure or customize Windows 2000, use the programs in Control Panel or Microsoft Management Console (MMC) whenever possible.

Setting Up DNS for Active Directory

Basic Concepts of DNS and Active Directory

Active Directory is the Windows 2000 directory service. A directory service consists of the following components:

- An information repository used to store information about objects
- The services that make that information available to users and applications

Like DNS, Active Directory is a distributed database that can be partitioned and replicated. Active Directory domains are identified with DNS names. Active Directory uses DNS as its *location service*, enabling computers to find the location of domain controllers. To find a domain controller in a particular domain, a client queries DNS for SRV and address (A) resource records that provide the names and IP addresses of the Lightweight Directory Access Protocol (LDAP) servers for the domain. LDAP is the protocol used to query and update Active Directory, and all domain controllers run an LDAP server. For more information about A and SRV resource records, see "Introduction to DNS" in this book. For more information about the domain locator service, see "Active Directory Logical Structure" in the *Microsoft® Windows® 2000 Server Resource Kit Distributed Systems Guide*.

For information about how to set up DNS to support Active Directory, see "Setting Up DNS for Active Directory" later in this chapter.

You cannot install Active Directory without having DNS on your network, because Active Directory uses DNS as its location service. However, you can install DNS separately, without Active Directory. If you install DNS on a domain controller, you can also choose whether or not to use Active Directory to provide storage and replication for DNS. Using Active Directory for storage and replication provides the following benefits:

- Increased fault tolerance
- Security
- Easier management
- More efficient replication of large zones

For DNS to function as a location service for Active Directory, you must have a DNS server to host the locator records (A, SRV, and CNAME). For more information about the locator, see "Active Directory Logical Structure" in the *Microsoft® Windows® 2000 Server Resource Kit Distributed Systems Guide*.

You can configure your Windows 2000 DNS server automatically by using the Active Directory Installation wizard, which is a wizard provided in Windows 2000 that installs and configures Active Directory. The Active Directory Installation wizard can perform all the installation and configuration necessary for DNS, and the Netlogon service adds the necessary locator records. For more information about the Active Directory Installation wizard, see "Using the Active Directory Installation Wizard" later in this chapter.

Unless you are using a DNS server other than Windows 2000 or you want to perform special configuration, you do not need to manually configure DNS to support Active Directory. However, if you want to set up a configuration other than the default configuration that the Active Directory Installation wizard sets up, you can manually configure DNS. In Windows 2000, you can configure DNS by using the DNS console. For information about the DNS console and when you might want or need to use it, see "Using the Configure DNS Server Wizard" later in this chapter.

If you are using a third-party DNS server, you must also perform manual configuration. For information about issues related to configuring DNS when you are using a third-party DNS server, see "Configuring Non-Windows 2000 DNS Servers to Support Active Directory" later in this chapter.

Using the Active Directory Installation Wizard

The Active Directory Installation wizard promotes the computer to the role of domain controller, installs Active Directory, and can install and configure the DNS server. For more information about the Active Directory Installation wizard, see "Active Directory Data Storage" in the *Microsoft® Windows® 2000 Server Resource Kit Distributed Systems Guide*.

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 Active Directory domain and then checks whether that server is going to accept dynamic updates. If the test is positive, the wizard does not install and configure a local DNS server.

If the Active Directory Installation wizard cannot find the DNS server that is authoritative for the name, or if the server it finds does not support dynamic updates or is not configured to accept dynamic updates, the Active Directory Installation wizard asks you whether you want the wizard to automatically install and configure a local DNS server. If you answer yes, the wizard automatically installs and configures the DNS Server service.

During automatic configuration, the Active Directory Installation wizard adds to the DNS server the forward lookup zone that will host the locator records and configures the DNS server to accept dynamic updates. (A *forward lookup zone* is a zone that contains information needed to resolve names within the DNS domain.) In some cases, it also primes the root hints with the names of the root servers. The wizard uses the following process to determine whether to prime the root hints:

The Active Directory Installation wizard examines the TCP/IP configuration of the computer and checks whether the computer is configured to use any DNS servers. If so, the Active Directory Installation wizard queries for the root servers. If it finds root DNS servers, it primes the root hints with the names of the root DNS servers.

If the resolver is not configured to use any DNS servers, the Active Directory Installation wizard queries for the root DNS servers specified in the file Cache.dns. By default, these are the Internet root servers. If it finds root DNS servers, it primes the root hints with the names of the root DNS servers. If it does not find any root servers, it creates a root zone on the DNS server, making it a root server.

After the Active Directory Installation wizard finishes, you are prompted to restart the computer. After the computer restarts, Netlogon attempts to add locator resource records to the DNS server by sending a dynamic update request to the authoritative DNS server. Locator resource records are necessary for other computers to locate this domain controller.

Note You can also invoke the Active Directory Installation wizard by executing an answer file that contains all of the settings that you need to configure. An *answer file* is a file that a wizard uses to provide answers to questions. For more information about the answer file for the Active Directory Installation wizard, see "Active Directory Data Storage" in the *Microsoft® Windows® 2000 Server Resource Kit Distributed Systems Guide*.

Follow the steps below to install and configure DNS and Active Directory. For more information about installing and configuring Active Directory, see "Active Directory Data Storage" in the *Microsoft® Windows® 2000 Server Resource Kit Distributed Systems Guide*.

To configure DNS and Active Directory

1. Log on as Administrator.
2. Check the TCP/IP settings of your computer to make sure it is configured to use a DNS server. If your computer is the first DNS server on the network, you can configure your computer to use itself as a DNS server.
3. If the Windows 2000 Configure Your Server wizard is not already open on your computer, click **Start**, point to **Programs** and **Administrative Tools**, and then click **Configure Your Server**.
4. Use the Windows 2000 Configure Your Server wizard to install and configure Active Directory. The Windows 2000 Configure Your Server wizard asks you questions about your configuration and then starts the Active Directory Installation wizard, which installs and configures Active Directory. If it's necessary, the Active Directory Installation wizard also guides you through the installation and configuration of the DNS server component.
5. When directed to do so, restart your computer.

After you have run the Active Directory Installation wizard, you might need to add a delegation in the parent zone of the zone you created. If this server is a root DNS server, there is no parent zone; therefore, you do not need to add a delegation. However, if there are other DNS servers that are running on the network, you must add a delegation.

To add a delegation

1. Locate the zone that the Active Directory Installation wizard created. The Active Directory Installation wizard automatically creates a zone with the same name as the Active Directory domain you created.
2. Locate the parent zone for this zone.
3. On the parent zone, add the delegation.

Using the Configure DNS Server Wizard

In most cases, you do not need to manually configure DNS to support Active Directory; you can let the Active Directory Installation wizard automatically configure DNS. However, you can use the Configure DNS Server wizard to configure DNS if you want a DNS configuration other than the default configuration that the Active Directory Installation wizard sets up. For example, you might want your DNS server to be different from your domain controller.

If you plan to use the Configure DNS Server wizard to configure your DNS server, perform the following tasks before running the wizard:

- If the DNS server is not already installed, install it.
- If this server will not be the root DNS server, configure its network connections to point to one or more DNS servers in your network.

While you are running the wizard or after you have completed the wizard, you must create a forward lookup zone that is authoritative for the locator records that Netlogon will add.

After you have completed configuration of your DNS server by using the wizard, you must perform the following tasks:

- Enable dynamic updates on that zone.
- Unless this is a root zone, add a delegation to the new forward lookup zone in its parent zone.
- Make sure that the server that will be a domain controller has network connectivity to this server.

To configure a DNS server that is not running on a domain controller, you must be a member of the Administrators group for that computer.

To configure a DNS server that is running on a domain controller, you must be a member of at least one of the groups listed in the access control list (ACL) of the MicrosoftDNS container in Active Directory. The group must also have Full Control permissions. By default, the following groups are listed in the ACL:

- DNS Administrators
- Domain Administrators
- Enterprise Administrators

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

To verify DNS client settings

1. Right-click **My Network Places**, and then click **Properties**.
2. Right-click the connection for which you want to configure the DNS server, and then click **Properties**.
3. Click **Internet Protocol (TCP/IP)** and then click **Properties**.
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, and then enter the servers in the **DNS server addresses** box.

The Configure DNS Server wizard uses the DNS client information to determine whether there are any root DNS servers on the network. For more information about setting the DNS server IP address, see Windows 2000 Server Help.

Also, you must install the DNS server before configuring the server. To install and configure the DNS server, perform the following procedures:

To install the DNS server

1. In Control Panel, double-click **Add/Remove Programs**, and then click **Add/Remove Windows Components**.
2. Click **Components**, and then click **Next**.
3. Click **Networking Services**, and then click **Details**.
4. If it is not already selected, select the check box next to **Domain Name System (DNS)**, and then click **OK**.
5. Click **Next**. Windows 2000 installs DNS.
6. Click **Finish**.

To configure the DNS server

1. In Control Panel, double-click **Administrative Tools** and then double-click **DNS**.
2. Click the DNS server to expand it.
3. Right-click the name of the server, and select **Configure the server** from the context menu. The Configure DNS Server wizard starts and guides you through the process of setting up DNS. In some cases, this includes creating a reverse lookup zone. For more information about creating a reverse lookup zone, see "Adding a Reverse Lookup Zone" later in this chapter.
4. Optionally, if Active Directory has already been installed, integrate the zone with Active Directory. For information about integrating the zone with Active Directory, see "Active Directory Integration and Multimaster Replication" later in this chapter.

The Configure DNS Server wizard prompts you for all the information needed to create the appropriate forward and reverse lookup zones.

The Configure DNS Server wizard also primes the root hints and creates a root zone, if necessary, exactly as the Active Directory Installation wizard does. However, it does not create a reverse lookup zone, so you must do that later. For more information about creating reverse lookup zones, see "Adding a Reverse Lookup Zone" later in this chapter.

If you are creating an Active Directory domain, you must perform some additional configuration.

To configure the DNS server to support Active Directory

1. Make sure that you have a forward lookup zone that is authoritative for the resource records registered by Netlogon.
2. Configure the forward lookup zone to enable dynamic update.
3. Unless this DNS server is a root DNS server, from the parent server, delegate the forward lookup zone to this server.

Adding a Reverse Lookup Zone

The Active Directory Installation wizard does not automatically add a reverse lookup zone and PTR resource records, because it is possible that another server, such as the parent server, controls the reverse lookup zone. You might want to add a reverse lookup zone to your server if no other server controls the reverse lookup zone for the hosts listed in your forward lookup zone. Reverse lookup zones and PTR resource records are not necessary for Active Directory to work, but you need them if you want clients to be able to resolve FQDNs from IP addresses. Also, PTR resource records are commonly used by some applications to verify the identities of clients.

The following sections explain where to put reverse lookup zones and how to create, configure, and delegate them. For information about any of the IP addressing concepts discussed in the following sections, see "Introduction to TCP/IP" in this book.

Planning for Reverse Lookup Zones

To determine where to place your reverse lookup zones, first gather a list of all the subnets in your network, and then examine the class (A, B, or C) and type (class-based or subnetted) of each subnet.

To simplify administration, create as few reverse lookup zones as possible. For example, if you have only one class C network identifier (even if you have subnetted your network), it is simplest to organize your reverse lookup zones along class C boundaries. You can add the reverse lookup zone and all the PTR resource records on an existing DNS server on your network.

Subdomains do not need to have their own reverse lookup zones. If you have multiple class C network identifiers, for each one you can configure a reverse lookup zone and PTR resource records on the primary name server closest to the subnet with that network identifier.

However, organizing your reverse lookup zones along class C boundaries might not always be possible. For example, if your organization has a small network, you might have received only a portion of a class C address from your ISP. Table 6.3 shows how to configure your network with each type of subnet.

Table 6.3 Planning Reverse Lookup Zones

Network Type	Recommended Action	See Section in This Chapter
Class A network	Configure your reverse lookup zone on the primary name server for the top-level domain.	"Configuring a Standard Reverse Lookup Zone"
Class B network	Configure your reverse lookup zone on the primary name server for the top-level domain.	"Configuring a Standard Reverse Lookup Zone"
Class C network	Configure your reverse lookup zone on the primary name server for the top-level domain.	"Configuring a Standard Reverse Lookup Zone"

Subnetted class A network	Divide your network into class B or C networks.	"Configuring a Standard Reverse Lookup Zone"
Subnetted class B network	Divide your network into class C networks.	"Configuring a Standard Reverse Lookup Zone"
Subnetted class C network, owner of class C network manages the reverse lookup zone	Rely on the owner of the class C network to manage the reverse lookup zone.	Not applicable.
Subnetted class C network, owner of class C network has delegated the reverse lookup zone for your network to you	Configure a classless In-addr.arpa reverse lookup zone.	"Configuring and Delegating a Classless In-addr.arpa Reverse Lookup Zone"

Configuring a Standard Reverse Lookup Zone

The following procedures describe how to add a reverse lookup for a class C network ID.

To add a reverse lookup zone

1. In Control Panel, double-click **Administrative Tools** and then double-click **DNS**.
2. Optionally, if the server to which you want to add a reverse lookup zone does not appear in the list, right-click **DNS**, click **Connect to Computer**, and then follow the instructions to add the desired server.
3. To display the zones, click the server name.
4. Right-click the **Reverse Lookup Zones** folder, and click **New Zone**. A zone configuration wizard appears.

Windows 2000-based clients and Windows 2000 DHCP servers can automatically add PTR resource records, or you can configure PTR resource records at the same time as when you create A resource records; otherwise, you might want to add PTR resource records manually.

To add PTR resource records

1. In Control Panel, double-click **Administrative Tools** and then double-click **DNS**.
2. To display the zones, click the server name.
3. Right-click the zone in the **Reverse Lookup Zones** folder, point to **New**, and then point to **Pointer**.
4. To create the PTR resource record, follow the instructions in the dialog box.

Note If you can't select the **Pointer** field because it is shaded, double-click the zone.

Configuring and Delegating a Classless In-addr.arpa Reverse Lookup Zone

Many organizations divide class C networks into smaller portions. This process is referred to as "subnetting a network." If you have subnetted a network, you can create corresponding subnetted reverse lookup zones, as specified in RFC 2317. Although your network has been subnetted, you do not need to create corresponding subnetted reverse lookup zones. It is an administrative choice. DNS servers and zones are independent of the underlying subnetted infrastructure.

However, in certain situations, you might want to create and delegate classless reverse lookup zones. If you own one class C address, and you want to distribute the addresses in the range to several different groups (for example, branch offices), but you do not want to manage the reverse lookup zones for those addresses, you would create classless reverse lookup zones and delegate them to those groups. For example, suppose that an ISP has a class C address and has given the first 624 addresses to Reskit. The ISP can include records in its zone indicating that the name server on Reskit has information about that portion of the namespace. Reskit can then manage that portion of the namespace by including resource records with the IP address–to–host mappings, also known as a *classless in-addr.arpa reverse lookup zone*.

The following sections, explain the syntax of classless reverse lookup zones and describe how to delegate and configure reverse lookup zones by using the preceding example. For more information about delegating reverse lookup zones, see the Request for Comments link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. Search for RFC 2317, "Classless in-addr.arpa delegation."

Note Dynamic update does not work with classless in-addr.arpa zones. If you need to dynamically update PTR resource records, do not use classless zones.

Syntax of a Classless In-addr.arpa Reverse Lookup Zone

You can use the following notation to specify the name of the classless in-addr.arpa reverse lookup zone:

<subnet-specific label>.*<octet>*.*<octet>*.*<octet>*.**in-addr.arpa**

where *octet* specifies an octet of the IP address range. The octets are specified in reverse order of the order in which they appear in the IP address.

Although *subnet-specific label* could be comprised of any characters allowed by the authoritative DNS server, the most commonly used formats include the following:

- *<minimum value of the subnet range>*-*<maximum value of the subnet range>*
- *<subnet>* **/** *<subnet mask bit count>*
- *<subnet ID>*

Subnet specifies which segment of the class C IP address this network is using. *Subnet mask bit count* specifies how many bits the network is using for its subnet mask. *Subnet ID* specifies a name the administrator has chosen for the subnet.

For example, suppose that an ISP has a class C address 192.168.100.0 and has divided that address into four subnets of 624 hosts per network, with a subnet mask of 255.255.255.192, and given the first 624 host addresses to a company with the DNS name Reskit.com. The name of the classless reverse lookup zone can use any of the following syntax lines:

- 0-26.100.168.192.in-addr.arpa
- 0/26.100.168.192.in-addr.arpa
- Subnet1.100.168.192.in-addr.arpa

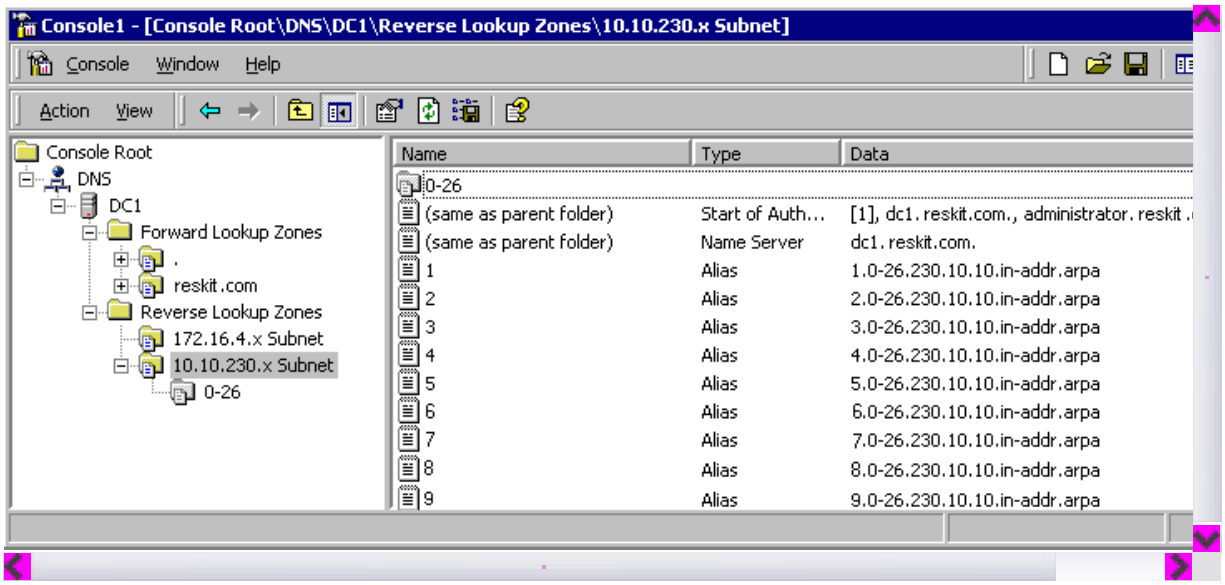
You can use any of this syntax in Windows 2000 DNS by entering the zones into a text file. For more information about creating and delegating subnetted reverse lookup zones through text files, see the Microsoft TechNet link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. Search Microsoft TechNet using the phrases "subnetted reverse

lookup zone" and "Windows NT."

Delegating a Classless Reverse Lookup Zone

You never need to delegate a classless reverse lookup zone, even if your network is subnetted. However, there are a few cases in which you might want to delegate a classless reverse lookup zone. For example, you might want to do so if you gave a merged organization a portion of your class C address, or if you had a remote subnetted network and wanted to avoid sending replication or zone transfer traffic across a wide area link.

Figure 6.10 shows how an administrator for a class C reverse lookup zone would then configure its DNS server.



If your browser does not support inline frames, click here to view on a separate page.

Figure 6.10 Reverse Lookup Delegations

You can delegate and create classless reverse lookup zones from within the DNS console.

To delegate a classless reverse lookup zone

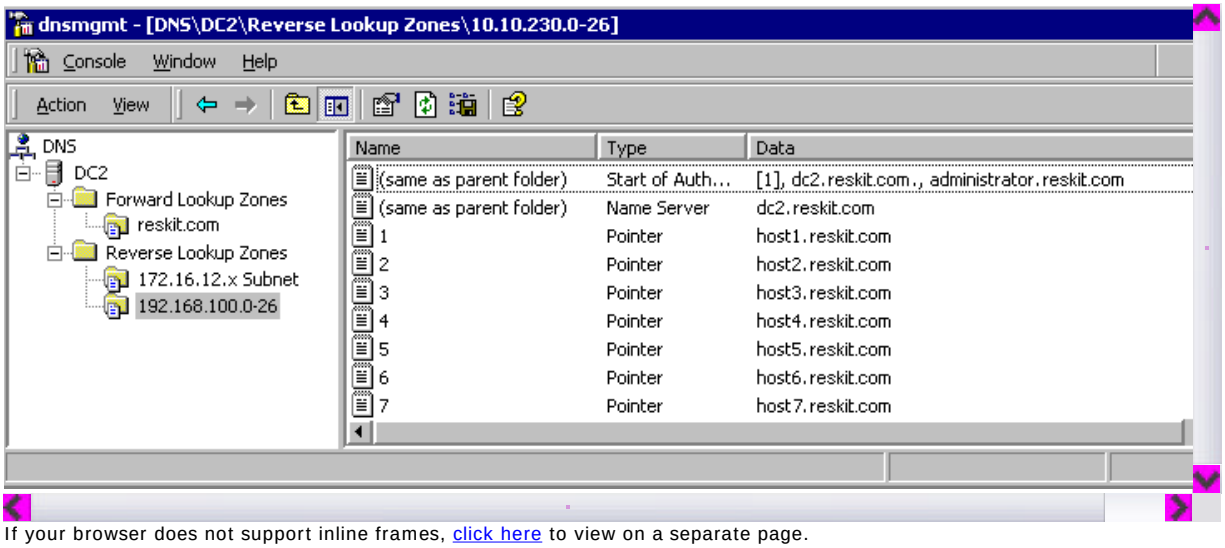
1. On the DNS server for your domain, create a reverse lookup zone.

   For the preceding example, create the reverse lookup zone 100.168.192.in-addr.arpa. The reverse lookup zone is added on the server for ISP.com, not Reskit.com.
2. Right-click the reverse lookup zone that you created, point to **New Delegation**.
3. In the New Delegation wizard, enter the name of the delegated domain and the name and IP address of the delegated name server. In the preceding example, the delegated domain is 0-26.
4. Right-click the reverse lookup zone and click **New alias**.
5. Add CNAME records for all the delegated addresses.

   For example, for the IP address 192.168.100.5, create a CNAME record of 5 that points to 5.0-26.100.168.192.in-addr.arpa.
6. Create the classless reverse lookup zone in the subdomain, by following the procedure in the following section.

Configuring a Classless In-addr.arpa Reverse Lookup Zone

You must configure a classless reverse lookup zone if one has been delegated to you. In the preceding example, an administrator for an ISP delegated a reverse lookup zone to Reskit.com, and an administrator for Reskit.com must therefore configure a classless reverse lookup zone. Figure 6.11 shows how Reskit.com would configure its classless reverse lookup zone.



If your browser does not support inline frames, click here to view on a separate page.

Figure 6.11 Classless Reverse Lookup Zone

To create a classless reverse lookup zone

1. In the DNS console, click the server name to display configuration detail it, right-click the **Reverse Lookup Zones** folder, and then click **Create a New Zone**. The Add New Zone wizard appears.

2. When you reach the **Network ID** page, in the field named **Enter the name of the zone directly**, enter the name of the classless reverse lookup zone.

   For example, type **0-26.100.168.192.in-addr.arpa**.

Then add any necessary PTR resource records in that zone.

Active Directory Integration and Multimaster Replication

In addition to storing zone files on DNS servers, you can store a primary zone in Active Directory. When you store a zone in Active Directory, zone data is stored as Active Directory objects and replicated as part of Active Directory replication.

Active Directory replication provides an advantage over standard DNS alone. With standard DNS, only the primary server for a zone can modify the zone. With Active Directory replication, all domain controllers for the domain can modify the zone and then replicate the changes to other domain controllers. This replication process is called *multimaster replication* because multiple domain controllers, or *masters*, can update the zone.

Although Active Directory–integrated zones are transferred by using Active Directory replication, you can also perform standard zone transfers to secondary servers as you can with standard DNS zones.

Active Directory–integrated storage provides the following benefits:

Fault Tolerance Although you can still perform standard zone transfers with Active Directory–integrated zones, Active Directory multimaster replication provides greater fault tolerance than using standard zone transfers alone. Standard zone transfers and updates rely on a single primary DNS server to update all the secondary servers. With Active Directory replication, however, there is no single point of failure for zone updates.

Security You can limit access to updates for any zone or record, preventing insecure dynamic updates. For more information about configuring secure dynamic update, see "Dynamic Update and Secure Dynamic Update" later in this chapter.

Simpler Management Because Active Directory performs replication, you do not need to set up and maintain a separate replication topology (that is, zone transfers) for DNS servers.

More Efficient Replication of Large Zones Active Directory replicates on a per-property basis, propagating only relevant changes. This is more efficient than full zone transfers.

Integrated Storage

When you configure a primary zone to be Active Directory–integrated, the zone is stored in Active Directory.

Figure 6.12 shows this configuration.

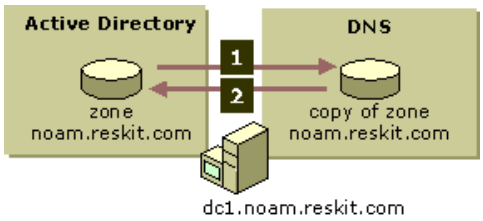
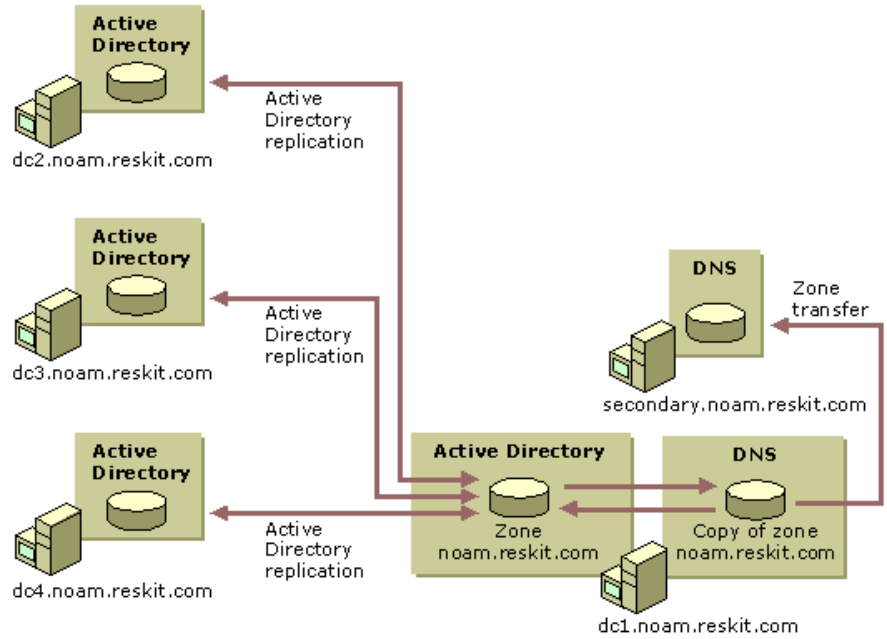


Figure 6.12 Active Directory–Integrated Zone

The DNS server component contains only a copy of the zone. When it starts up, it reads a copy of the zone from Active Directory (step 1). Then, when the DNS server receives a change, it writes the change to Active Directory (step 2).

Through Active Directory replication, the zone is replicated to other domain controllers. Also, through standard zone transfer, the DNS server can send its copy of the zone to any secondary DNS servers that request it. The DNS server can perform both incremental and full zone transfers. Figure 6.13 shows how the same zone can be replicated by using both Active Directory replication and standard zone transfer.



If your browser does not support inline frames, click here to view on a separate page.

Figure 6.13 Replication and Zone Transfer

By default, when an Active Directory–integrated DNS server starts up, it checks whether Active Directory is available and if it contains any DNS zones. If Active Directory does have zones, the DNS server loads zones from a location specified by the setting of **Load data on startup** in the properties page for the server within the DNS console. The DNS server can load zones from the following locations:

- If **Load data on startup** is set to **From registry**, the DNS server loads all local standard zone files and Active Directory–integrated zones specified in the following registry subkey:

  HKEY_LOCAL_MACHINE \SYSTEM \CurrentControlSet \Services \DNS \Zones

- If **Load data on startup** is set to **Boot From File**, the DNS server uses a BIND-style boot file to determine the location of the zone files.

  Note The DNS server automatically writes back to the boot file at regular intervals. You can also update the boot file by clicking on the server from within the DNS console and then by clicking the **Action** menu and selecting **Update Server Data Files**. Alternatively, you can stop and restart the server to update the boot file by right-clicking on the server from within the DNS console, pointing to **All Tasks** in the context-sensitive menu, and then clicking **Restart**.

- If **Load data on startup** is set to **From Active Directory and registry** (the default), the DNS server loads all Active Directory–integrated zones in the directory and all local standard zone files specified in the registry. (The DNS server must load *all* the files in the directory; you cannot configure the DNS server to load only some of the zones.)

The DNS server also loads the root hints and server and zone parameters from different locations depending on the **Load data on startup** setting. Table 6.4 shows the locations from which the DNS server loads and to which it writes zones, root hints, and server and zone parameters depending on the setting of **Load data on startup**.

Table 6.4 How the DNS Server Loads Zones, Root Hints, and Parameters

	Load Data on Startup: **Boot from File**	**Load Data on Startup:** **Boot from Registry**	**Load Data on Startup:** **Boot from Active Directory and Registry**
Read root hints from:	Root hints file	If available, the root hints file. Otherwise, if the Directory is available and contains root hints, the Directory.	If the Directory is available and contains root hints, from the Directory. Otherwise, the root hints file.
Write root hints to:	Root hints file	Root hints file.	If the Directory is available, the Directory.
Read zones from:	Boot file	Registry.	The Directory (for Active Directory–integrated zones) and the registry.
Write zones to:	Boot file and the registry	Registry and, if the zone is Active Directory–integrated, the Directory.	Registry and, if the zone is Active Directory–integrated, the Directory.
Read server and zones parameters from:	Boot file and the registry	Registry and (for Active Directory–integrated zones) the Directory.	The Directory (for Active Directory–integrated zones) and the registry.
Write server and zones parameters to:	Boot file and the registry	Registry (for all zones) and (for Active Directory–integrated zones) the Directory.	The Directory (for Active Directory–integrated zones) and the registry.

If you change the setting of **Load data on startup**, the DNS server first writes the root hints file, zones, and parameters to the locations specified in the original setting of **Load data on startup** and then reads them from the new setting.

If the server has loaded Active Directory–integrated zones, it periodically polls Active Directory for changes to those zones. The server also checks for the addition of new zones or the deletion of existing zones.

The DNS server can modify Active Directory if an administrator makes a change to the zone, or if the server is configured to accept dynamic updates and a dynamic update occurs. (Dynamic Update is described in "Dynamic Update and Secure Dynamic Update" later in this chapter.)

DNS servers update Active Directory by using the following procedure:

1. When an update occurs, the DNS server polls Active Directory to make sure that the copy of the zone in the memory of the DNS server is up to date. If not, the DNS server polls for any changes and incorporates those changes in the in-memory copy.
2. Next, the server verifies that all prerequisites are satisfied. Prerequisites are conditions that must be satisfied before records can be updated.
3. Finally, to accept the change, it updates the primary zone data in Active Directory.

Storage Location

The Active Directory directory service is an object-oriented database that organizes network resources in a hierarchical structure. Every resource is represented by an object.

Each object has attributes that define its characteristics.

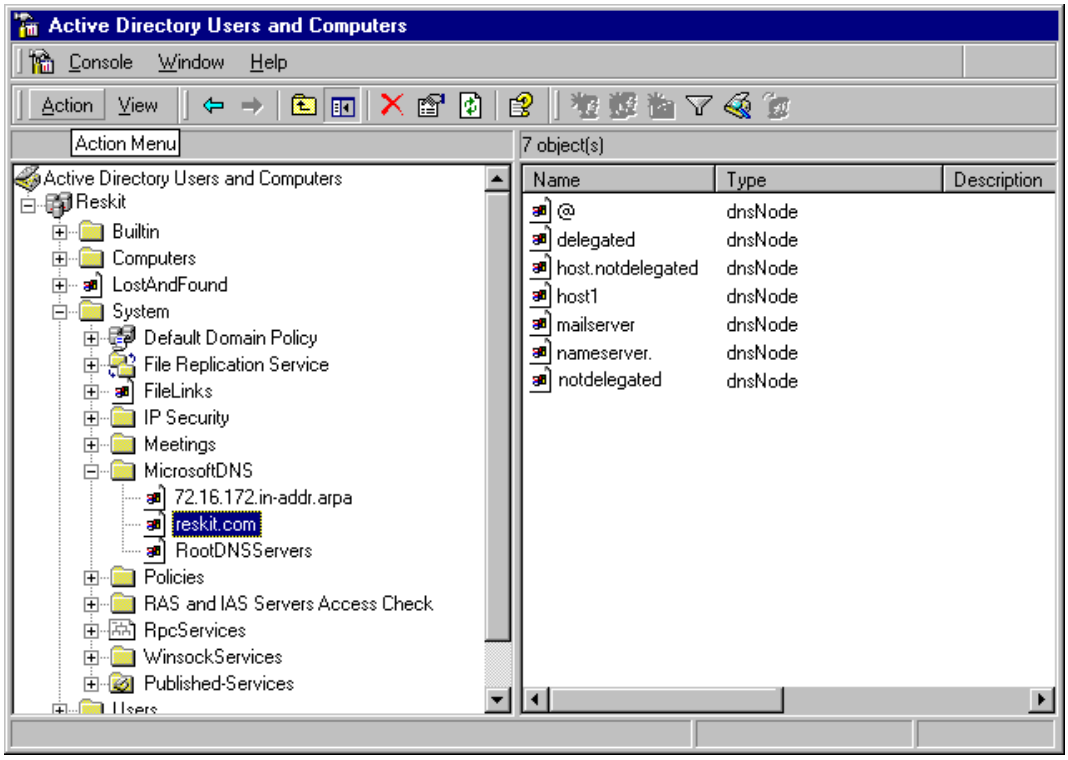
The classes of objects and the attributes of each object are defined in the Active Directory schema.

Table 6.5 shows the DNS objects in Active Directory.

Table 6.5 DNS Objects in Active Directory

Object	**Description**
dnsZone	Container created when a zone is stored in Active Directory
dnsNode	Leaf object used to map and associate a name in the zone to resource data
dnsRecord	Multivalued attribute of a dnsNode object used to store the resource records associated with the named node object
dnsProperty	Multivalued attribute of a dnsZone object used to store zone configuration information.

Figure 6.14 shows how DNS objects are represented in Active Directory.



If your browser does not support inline frames, click here to view on a separate page.

Figure 6.14 DNS Objects in Active Directory

Within the MicrosoftDNS container object are the dnsZone container objects. In Figure 6.14, MicrosoftDNS contains the following dnsZone objects:

- The reverse lookup zone, 72.16.172.in-addr.arpa
- The forward lookup zone, reskit.com
- The root hints, RootDNSServers

The dnsZone container object contains a dnsNode leaf object for every unique name within that zone. Figure 6.14 shows the following dnsNode objects within the dnsZone container object for reskit.com:

- **@**, which signifies that the node has the same name as the dnsZone object.
- **delegated**, a delegated subdomain.
- **host.notdelegated**, a host in the domain notdelegated.reskit.com, a domain that is controlled by the zone on reskit.com.
- **host1**, a host in the domain reskit.com.
- **mailserver**, the mail server in the domain reskit.com.
- **nameserver**, the name server in reskit.com.
- **notdelegated**, the domain notdelegated.reskit.com, which is controlled by the zone on reskit.com.

The dnsNode leaf object has a multivalued attribute called dnsRecord with an instance of a value for every record associated with the object's name. In this example, the dnsNode leaf object mailserver.reskit.com has an "A" attribute containing the IP address.

You can view the DNS objects from within the Active Directory Users and Computers console.

To view zones stored in Active Directory

1. Click **Start**, point to **Programs** and **Administrative Tools**, and then click **Active Directory Users and Computers**.
2. In the **View** menu, click **Advanced Features**.
3. Double-click the Domain object, the System object, and then the MicrosoftDNS object to display the dnsZone objects.
4. Double-click the zone that you want to view.

Although you can see the zone objects from within the Active Directory Users and Computers component, the Active Directory Users and Computers component cannot interpret the values of the dnsRecord attribute. If you want to view the DNS domain hierarchy and associated records, you do so from within the DNS console. For information about the DNS console, see "Setting Up DNS for Active Directory" earlier in this chapter. Alternatively, if you want to view the zones, you can retrieve them by using Nslookup. For more information about Nslookup, see "Troubleshooting" later in this chapter.

Creating, Converting, and Deleting Zones

You can store any number of zones in Active Directory. Zones stored in Active Directory act like primary zones: Any DNS server running on a domain controller in the domain can modify the zone.

To store a zone in Active Directory, you can either create an Active Directory–integrated zone or convert a primary or secondary zone to be Active Directory–integrated. You can also convert Active Directory–integrated zones back to standard primary or secondary zones. This section explains issues you need to consider when you create, convert, and delete zones. For information about how to create, convert, and delete zones, see Windows 2000 Server Help.

Creating an Active Directory–Integrated Zone

Any zone you create is automatically replicated to all domain controllers in the zone. Therefore, do not create the same zone on more than one domain controller.

Caution If you create a zone on one domain controller, and then create the same zone on a second domain controller before Active

Directory has replicated the zone, Active Directory deletes the zone on the first domain controller. As a result, you lose any changes that you made to the version of the zone that you created on the first domain controller.

Converting a Standard Zone to an Active Directory–Integrated Zone

You can convert either a standard primary or secondary zone to an Active Directory–integrated zone. When you integrate a zone with Active Directory, consider the following issues:

- For a DNS server to use an Active Directory–integrated zone, that server must be running on a domain controller.
- You cannot load Active Directory–integrated zones from other domains. If you want your DNS server to be authoritative for an Active Directory–integrated zone from another domain, the server can only be a secondary server for that zone.
- There is no such thing as an Active Directory–integrated secondary zone. When you store a zone in Active Directory, all domain controllers can update the zone.
- You cannot have at the same time both an Active Directory–integrated zone and a standard primary copy of the same zone.

Converting an Active Directory–Integrated Zone to a Standard Zone

You can convert an Active Directory–integrated zone to either a standard primary or standard secondary zone.

If you convert an Active Directory–integrated zone to a standard *secondary* zone, the zone is copied to the name server on which you converted the zone. That server no longer loads the zone from Active Directory, but it has its own secondary copy of the zone. It requests zone transfers from whatever server you specified as the primary server for the zone.

If you convert an Active Directory–integrated zone to a standard *primary* zone, the zone is copied to a standard file on that server and is deleted from Active Directory. The zone no longer appears on other Active Directory–integrated DNS servers.

Deleting Zones

If you delete an Active Directory–integrated zone from a domain controller and **Load data on startup** is set to **Registry**, the DNS console asks you whether you also want to delete the zone from Active Directory. If you click **Yes**, the zone is completely deleted from Active Directory and is no longer available to be loaded onto any domain controllers. If you click **No**, the zone is removed from the registry but remains in Active Directory. The next time that the DNS server polls the directory for changes, if **Load data on startup**, on the **Advanced** tab of the DNS server properties page in the DNS console, is set to **From Active Directory and registry**, the zone reappears. If **Load data on startup** is set to **Registry**, on the other hand, the zone does not reappear.

If you delete a standard secondary zone from a domain controller, it is generally deleted from that domain controller. However, if a corresponding Active Directory–integrated zone exists, and you have configured the DNS server to load data on startup from Active Directory and the registry, the zone reappears as an Active Directory–integrated primary zone. You can then delete the Active Directory–integrated zone from the computer or from Active Directory.

Creating a Secondary Copy of an Active Directory–Integrated Zone

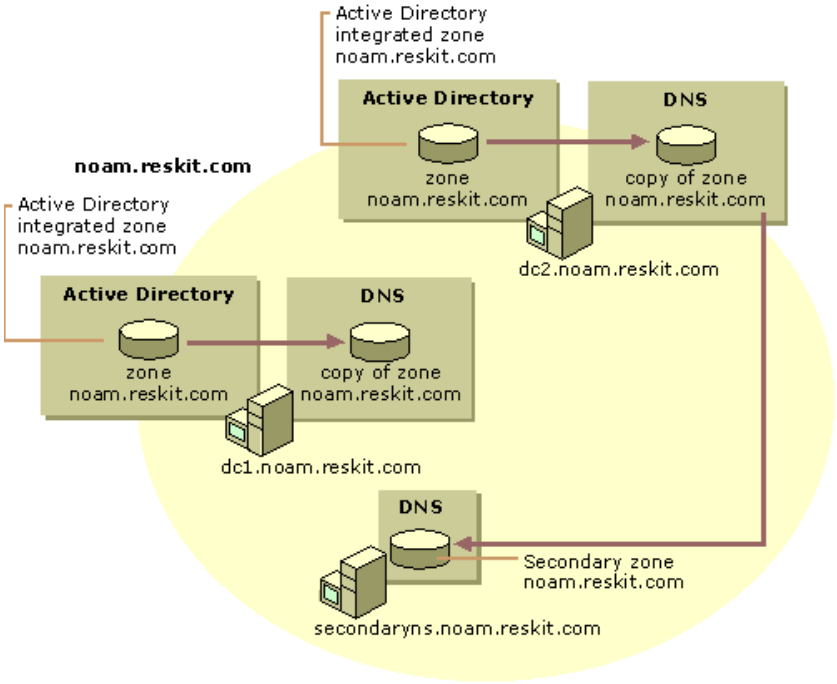
It is possible to integrate a zone in Active Directory and then add a secondary copy of the zone on another DNS server. You might want to create a secondary copy of an Active Directory–integrated zone; for example, if you have a remote site from which your users need to be able to resolve names, but you do not want to increase your network traffic by adding a domain controller, you might want to create a secondary copy of the zone.

Preventing Problems When Converting or Deleting Zones

When you delete a zone, or convert an Active Directory–integrated zone to a standard secondary zone, you can cause configuration errors. For example, if you delete a copy of the zone from a server and a secondary server is configured to pull zone transfers from that server, the secondary server is no longer able to pull zone transfers.

In another example, if you convert an Active Directory–integrated zone to a standard *primary* zone, the DNS server loading the new primary zone becomes the single master of the zone. Therefore, Active Directory removes the converted zone from Active Directory, which means that the zone is deleted from all domain controllers.

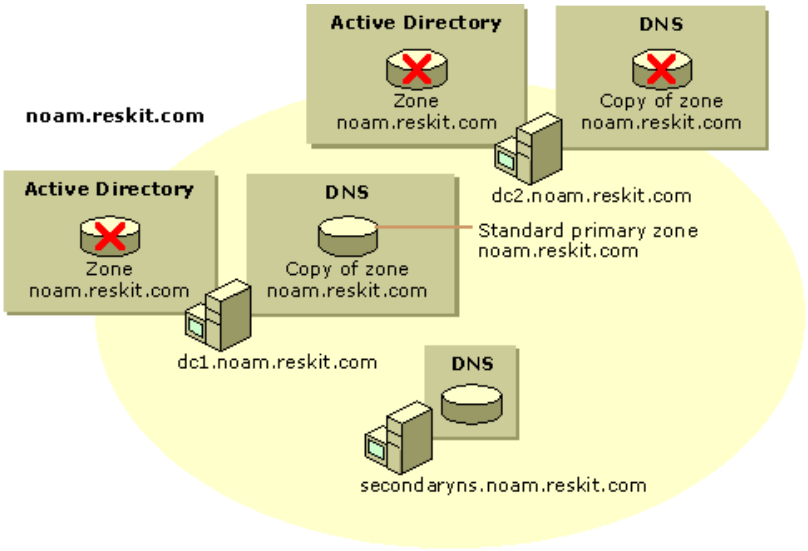
This can cause problems for secondary servers in some configurations. For example, suppose domain the noam.reskit.com has two Active Directory–integrated name servers, DC1.noam.reskit.com and DC2.noam.reskit.com; the domain has one secondary name server, SecondaryNS.noam.reskit.com, that has a secondary copy of the zone for noam.reskit.com and that points to DC2.noam.reskit.com as the master server for the zone. Figure 6.15 shows this configuration.



If your browser does not support inline frames, click here to view on a separate page.

Figure 6.15 Sample Domain Structure

Now, suppose that a user with the proper permissions logs on to DC1.noam.reskit.com and converts the zone from an Active Directory–integrated zone to a standard primary zone. As Figure 6.16 shows, DC1.noam.reskit.com will have a standard primary zone, and DC2.noam.reskit.com will not have a copy of the zone. Even though the zone is deleted from DC2.noam.reskit.com, SecondaryNS.noam.reskit.com still points to DC2.noam.reskit.com as the master server from the zone, and SecondaryNS.noam.reskit.com has no way to get a copy of the zone by using zone transfers.



If your browser does not support inline frames, click here to view on a separate page.

Figure 6.16 Orphaned Secondary Server

To prevent this problem, be sure to update all secondary servers for the zone that you are converting from an Active Directory–integrated zone to a standard primary zone.

This problem occurs only if you delete a zone from a server or you are converting an Active Directory–integrated zone to a standard primary zone, and a secondary server is pointing at a server from which the zone was deleted. The problem will not occur if you are converting an Active Directory–integrated zone to a standard secondary zone, because converting an Active Directory–integrated zone to a standard secondary does not cause the zone to be deleted from any server.

Multimaster Replication

Active Directory supports *multimaster replication*, which is replication in which any domain controller can send or receive updates of information stored in Active Directory. Replication processing is performed on a per-property basis, which means that only relevant changes are propagated. Replication processing differs from DNS full zone transfers, in which the entire zone is propagated. Replication processing also differs from incremental zone transfers, in which the server transfers all changes made since the last change. With Active Directory replication, however, only the final result of all changes to a record is sent.

When you store a primary zone in Active Directory, the zone information is replicated to all domain controllers within the Active Directory domain. Every DNS server running on a domain controller is then authoritative for that zone and can update it.

Name Collisions

Because all domain controllers in the domain can make changes to the same zone, it is possible for someone to update a property of an Active Directory object on one domain controller and someone else to update the same property on another domain controller simultaneously (or nearly simultaneously), thus making the information about the property on one domain controller inconsistent with that on the other domain controller. When a property changes in a second domain controller before a change from the first server replica has been propagated, a *replication collision* occurs.

Replication collisions can affect Active Directory–integrated DNS zones. Suppose that the same name is simultaneously created within the same domain and on two different domain controllers. The changes replicate, and Active Directory determines that there are two different dnsNode objects that have the same name. To solve the problem, the replication subsystem of Active Directory changes the name of the object that was created first by adding to the name a special character and a globally unique identifier (GUID), which is a unique 128-bit number that Active Directory associates with an object to make the object unique. This "disambiguates" the name of the object so that the two objects have different names. The next time that the DNS server pulls changes from Active Directory, the DNS server deletes the copy of the host object with the GUID. Thus, DNS accepts the last name to be created.

If you simultaneously modify a name object on two different server replicas, Active Directory must decide which change (attribute value) will be accepted and which will be discarded. To do so, Active Directory selects the attribute value that has the highest version number. If the version numbers are the same, Active Directory selects the attribute value that has the latest timestamp. Thus, DNS accepts the second change. For more information about replication collisions, see "Active Directory Replication" in the *Microsoft® Windows® 2000 Server Resource Kit Distributed Systems Guide*.

Causing Immediate Replication

When setting up DNS or troubleshooting replicas, you might not want to wait for the normal replication cycle. If so, you can cause replication to take place immediately. Keep in mind that your network performance affects how long it takes to update the target domain controller.

To cause immediate replication

1. Click **Start**, point to **Programs**, point to **Administrative Tools**, and then click **Active Directory Sites and Services**.
2. Double-click the **Sites** icon to expand it.

   All sites are displayed — including the first site, labeled **Default-First-Site-Name** — and any other site that has been manually configured.
3. Double-click the site that you want to expand.
4. Under the site you want, double-click the **Servers** icon to expand it, and then expand the icon for the computer. The **NTDS**

Settings icon is displayed.

5. Click the **NTDS Settings** icon.

   One or more objects are listed in the right pane. One of those objects is a link to the domain controller on which you want to cause immediate replication.

6. Right-click the object that links to the domain controller on which you want to cause immediate replication, and then click **Replicate Now**.

Dynamic Update and Secure Dynamic Update

Windows 2000 supports both dynamic update, defined in RFC 2136, and secure dynamic update, defined in the IETF Internet-Draft "GSS Algorithm for TSIG (GSS-TSIG)."

With dynamic update, clients can automatically send updates to the name server that is authoritative for the record they want to change. The authoritative name server then checks to make sure that certain prerequisites have been met. *Prerequisites* are resource records that must be present or absent before records can be updated. For more information about prerequisites, see "Introduction to DNS" in this book. If the prerequisites have been met, the authoritative name server makes the change. The change can be adding records, deleting records, or modifying records.

Note Both clients and servers can send dynamic updates.

Dynamic update provides the following benefits:

- Enables clients, including DHCP clients, to dynamically register A and PTR resource records with a primary server. This reduces the administrative resources needed to manually manage those records.
- Enables DHCP servers to register A and PTR resource records on behalf of DHCP clients. This reduces the time needed to manually manage those records and provides support for DHCP clients that cannot perform dynamic updates.
- Simplifies the setup of Active Directory by allowing domain controllers to be dynamically registered by using SRV records.

Secure dynamic update works like dynamic update, with the following exception: the authoritative name server accepts updates only from clients and servers that are authorized to make dynamic updates to the dnsZone and dnsNode objects.

Secure dynamic update provides the following benefits:

- Protects zones and resource records from being modified by users without authorization.
- Enables you to specify exactly which users and groups can modify zones and resource records.

Note Any primary zone can be configured for dynamic update. However, only Active Directory–integrated zones can be configured for secure dynamic update.

By default, the dynamic update client attempts a dynamic update first, and if it fails, negotiates a secure dynamic update. However, you can also configure it to always attempt insecure dynamic update or to always attempt secure dynamic update by adding the **UpdateSecurityLevel** registry entry to the following subkey:

```
HKEY_LOCAL_MACHINE \SYSTEM \CurrentControlSet \Services \Tcpip \Parameters
```

The value of **UpdateSecurityLevel** can be set to the decimal values 0, 16, or 256, which configure security as follows:

- *256*. Specifies the use of secure dynamic update only.
- *16*. Specifies the use of insecure dynamic update only.
- *0*. Specifies the use of secure dynamic update when an insecure dynamic update is refused. This is the default value.

Caution If you disable secure dynamic update, the client is not able to perform updates on zones that have been configured for secure dynamic update.

Also, if you configure a zone to use only secure dynamic update, make sure that the DHCP servers that update records in the zone are not installed on domain controllers. Otherwise, the DHCP server that performs registration of A resource records on behalf of any of its clients can take ownership of names that belong to computers that register their own records.

Dynamic Update

This section describes the Windows 2000 implementation of dynamic update. For information about the dynamic update standard specified in RFC 2136, see "Introduction to DNS" in this book.

Note Dynamic updates can be sent on behalf of different services such as the DHCP client, the DHCP server, Netlogon, and cluster services. The following sections describe only dynamic updates performed by the DHCP client and server.

In Windows 2000, clients can send dynamic updates for three different types of network adapters: DHCP adapters, statically configured adapters, and remote access adapters. Regardless of which adapter is used, the DHCP client service sends dynamic updates to the authoritative DNS server. The DHCP client service runs on all computers regardless of whether they are configured as DHCP clients.

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

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

By default, the dynamic update client automatically deregisters name–to–IP address mappings whenever the DHCP lease expires. You can configure the client not to register its name and IP address in DNS. If you configure the client not to automatically register name–to–IP address mappings and the DHCP server is running Windows 2000, and it is configured to register DNS resource records on behalf on clients that are running versions of Windows earlier than Windows 2000, the DHCP server attempts to update the mappings instead.

To prevent the client from registering name–to–IP address mappings

1. Double-click the **Network** icon in Control Panel.
2. Right-click the icon for the connection on which you want to disable registration of name–to–IP address mappings, and then click **Properties**.
3. Click **Internet Protocol (TCP/IP)**, and then click **Properties**.
4. Click **Advanced**, and then click the **DNS** tab.
5. Clear the check box **Register this connection's address in DNS**.

You can force a re-registration by using the command-line tool Ipconfig. For Windows 2000–based clients, type the following at the command prompt:

```
ipconfig /registerdns
```

For Windows NT 4.0–based clients, type the following:

```
ipconfig /release
ipconfig /renew
```

For Microsoft® Windows® 98–based and Microsoft® Windows® 95–based clients, type the following:

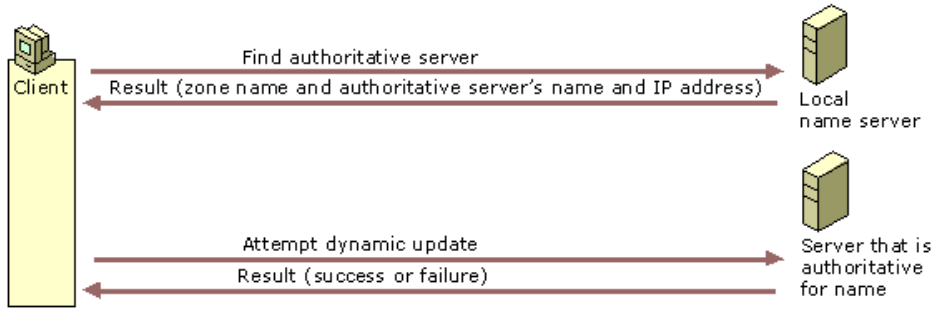
```
winipcfg /renew
```

Dynamic Update Process

In a dynamic update, the following events occur:

1. The client queries its local name server (using the process described in "DNS Queries," earlier in this chapter) to find the primary name server and the zone that is authoritative for the name it is updating. The local name server then performs the standard name resolution process to discover the primary name server that is authoritative for the name. (The local name server can also be the server that is authoritative for the name.) Then it responds with the name of the authoritative server and zone.
2. The client sends a dynamic update request to the primary server that is authoritative for the zone. The dynamic update request can include a list of prerequisites that must be fulfilled before the update can be made. The authoritative server then begins the dynamic update process. (For information about what happens if the zone has been configured for secure dynamic update, see "Secure Dynamic Update" later in this chapter.) The authoritative server then checks whether the prerequisites have been fulfilled. If they have, the server performs the update, then replies to the client.

Figure 6.17 shows a typical dynamic update process.



If your browser does not support inline frames, click here to view on a separate page.

Figure 6.17 Dynamic Update Process

Updates can fail for the following reasons:

- The primary server that is authoritative for the name does not respond.

  The primary server might not respond if it is down or if the local name server has an incorrect or outdated name server listed in its SOA resource record. DNS servers with standard zones (including secondary servers for Active Directory–integrated zones) can cause problems by sending incorrect or outdated SOA records when dynamic update clients request them. However, DNS servers with Active Directory–integrated zones always include their name in the SOA records, so DNS servers with Active Directory–integrated zones do not send incorrect or outdated SOA records.

  If the primary server does not respond but the zone is replicated through multimaster replication, the client attempts to register the name with the other primary DNS servers that are authoritative for the name.

  If the update fails because the server is not available, the client logs a message in the event log, which you can view by using Event Viewer. You can also configure the server log, Dns.log, to show a failure. For more information about Event Viewer, see "Troubleshooting" later in this chapter.
- The server is not accepting dynamic updates because the zone is being transferred.
- The server accepts only secure dynamic updates, and the insecure dynamic update operation failed.

  For more information about secure dynamic update, see "Secure Dynamic Update" later in this chapter.
- The prerequisites have not been met. For example, the dynamic update client might be trying to update a name for which no records currently exist.

The following sections describe the dynamic update process for adapters configured by DHCP, statically configured adapters (adapters for which a user or administrator has manually entered the IP address), and remote access adapters.

DHCP Clients and Servers

Windows 2000 DHCP clients are dynamic update–aware and can initiate the dynamic update process. A DHCP client negotiates the process of dynamic update with the DHCP server when the client leases an IP address or renews the lease, determining which computer will update the A and PTR resource records of the client for the FQDN (which can contain a connection-specific DNS suffix). Depending on the negotiation process, the DHCP client, the DHCP server, or both, update the records by sending a dynamic update request to a primary DNS server that is authoritative for the name that is to be updated.

Clients and servers that are running versions of Windows earlier than Windows 2000 do not support dynamic update. However, Windows 2000 DHCP servers can perform dynamic updates on behalf of clients that do not support the FQDN option (which is described in the following section). For example, clients that are running Windows 95, Windows 98, and Windows NT do not support the FQDN option. To enable this functionality, in the **DNS** tab of the server properties for the DHCP console, select the option **Enable updates for DNS clients that do not support dynamic updates**. The DHCP server first obtains the name of legacy clients from the DHCP REQUEST packet. It then appends the domain name given for that scope and registers the A and PTR resource records.

For information about how security for clients that do not support the FQDN option is implemented through secure dynamic update, see "Secure Dynamic Update" later in this chapter.

In some cases, stale PTR or A resource records can appear on DNS servers when the lease of a DHCP client expires. For example, when a Windows 2000 DHCP client tries to negotiate a dynamic update procedure with a Windows NT 4.0 DHCP server, the Windows 2000 DHCP client must register both A and PTR resource records itself. Later, if the Windows 2000 DHCP client is improperly removed from the network, the client cannot deregister its A and PTR resource records; thus, they become stale.

If a stale A resource record appears in a zone that allows only secure dynamic updates, no person or computer is able to use the name in that A resource record.

To prevent problems with stale PTR and A resource records, you can enable the aging and scavenging feature. For more information about the aging and scavenging feature, see "Aging and Scavenging" later in this chapter.

To provide fault tolerance, consider integrating with Active Directory those zones that accept dynamic updates from Windows 2000–based clients. If you want to speed up the discovery of authoritative servers, you can configure each client with a list of preferred and alternate DNS servers that are authoritative for that directory-integrated zone. If a client fails to update with its preferred server because the server is unavailable, the client can try an alternate server. When the preferred server becomes available, it loads the updated, directory-integrated zone that includes the update from the client.

Dynamic Update Process for Adapters Configured by DHCP

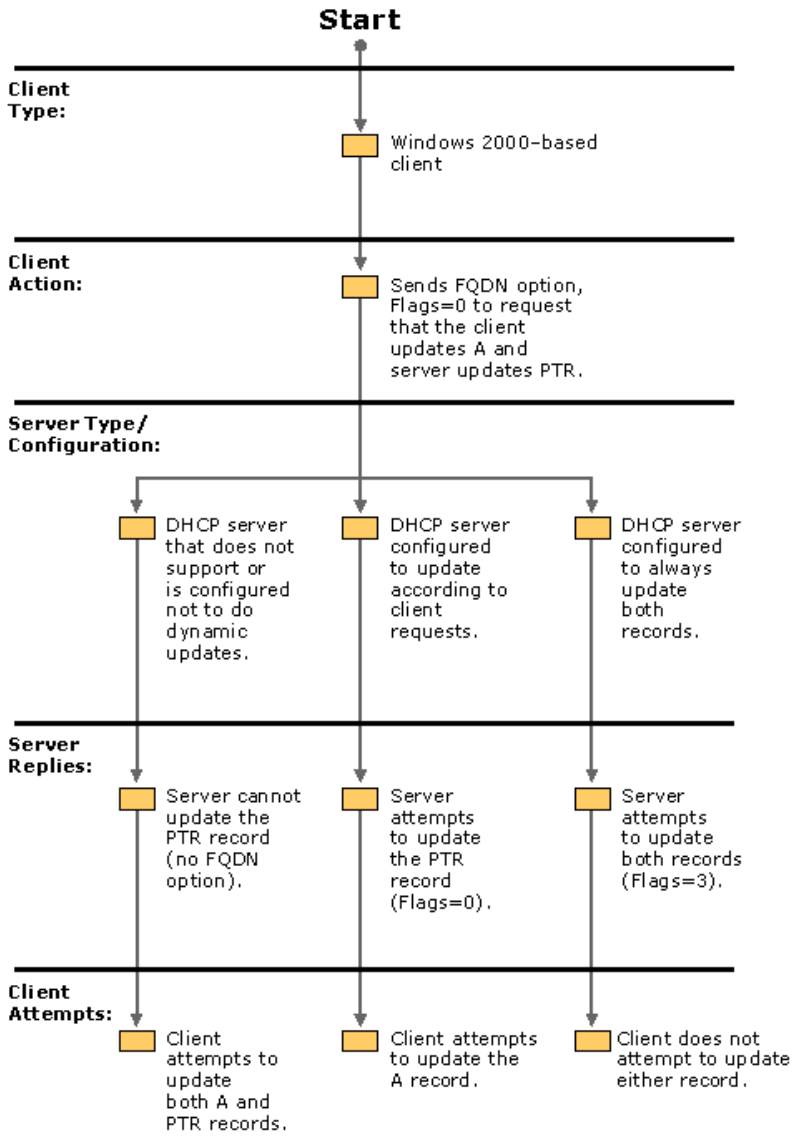
To negotiate the dynamic update process, the DHCP client sends its FQDN to the DHCP server in the DHCPREQUEST packet by using the FQDN option. The server then replies to the DHCP client by sending a DHCP acknowledgment (DHCPACK) message by using the FQDN option.

Table 6.6 lists the fields of the FQDN option of the DHCPREQUEST packet.

Table 6.6 Fields in the FQDN Option of the DHCPREQUEST Packet

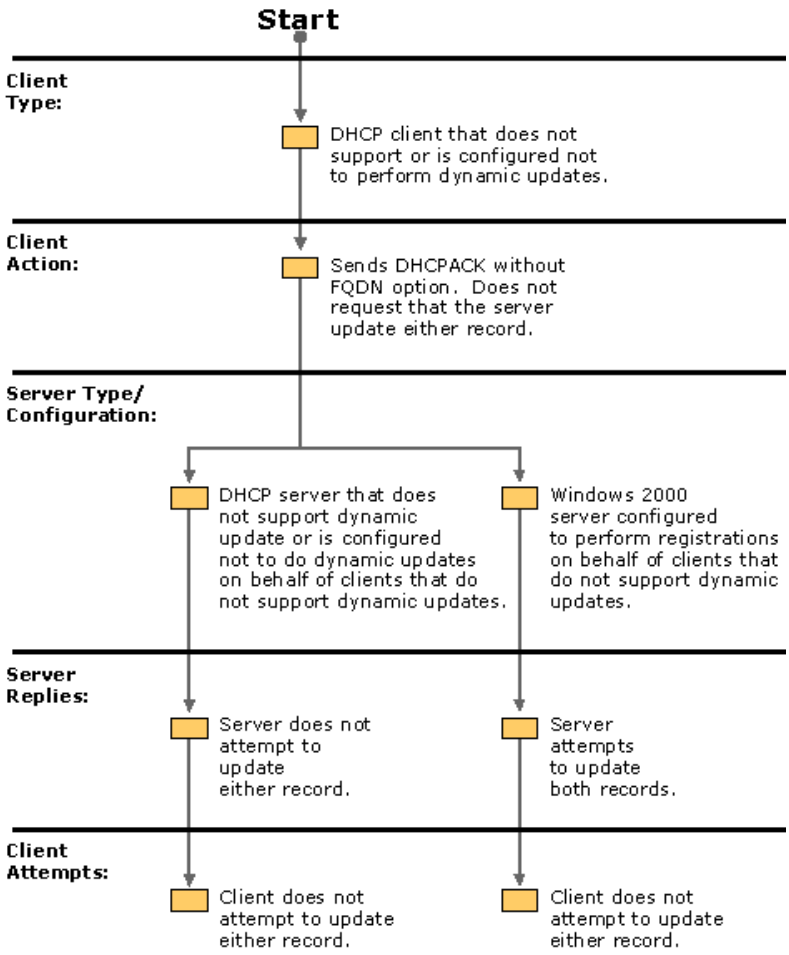
Field	Explanation
Code	Specifies the code for this option (81).
Len	Specifies the length of this option (minimum of 4).
Flags	Can be one of the following values:: *0*. Client wants to register the A resource record and requests that the server update the PTR resource record. *1*. Client wants server to register the A and PTR resource records. *3*. DHCP server registers the A and PTR resource records regardless of the request of the client.
RCODE1 and RCODE 2	The DHCP server uses these fields to specify the response code from an A resource record registration performed on the client's behalf and to indicate whether it attempted the update before sending DHCPACK.
Domain Name	Specifies the FQDN of the client.

As Figures 6.18 and 6.19 show, the conditions under which DHCP clients send the FQDN option and the action taken by DHCP servers depend on the operating system that the client and server are running and how the client and server are configured.



If your browser does not support inline frames, click here to view on a separate page.

Figure 6.18 Windows 2000–based Client



If your browser does not support inline frames, click here to view on a separate page.

Figure 6.19 Client That Does Not Perform Dynamic Updates

Whether the client requests dynamic update depends on whether the client is running Windows 2000 or another version of Windows. It also depends on and how the client is configured. Clients can take any of the following actions:

1. By default, the Windows 2000 DHCP client sends the FQDN option with the Flags field set to 0 to request that the client update the A resource record, and the DHCP server updates the PTR resource record. After the client requests the update, it waits for a response from the DHCP server. Unless the DHCP server sets the Flags field to 3, the client then initiates an update for the A resource record. If the DHCP server does not support or is not configured to perform registration of the DNS record, the client attempts registration of the A and PTR resource records.
2. DHCP clients that are running Windows operating systems of a version earlier than Windows 2000 and Windows 2000 DHCP clients configured not to register DNS resource records do not send the FQDN option. In this case, the client does not try to update either record.

Depending on what the client requests, the server can take different actions. If the DHCP client sends a DHCPREQUEST message without the FQDN option, what happens depends on the type of server and how the server is configured. The server can update both records anyway. The server does so if it is configured to update records on behalf of clients that do not support the FQDN option.

Alternatively, the server might do nothing. In the following cases, the server does nothing:

1. The server does not support dynamic update (for example, a Windows NT 4.0 server).
2. The server is running Windows 2000 and is configured not to do dynamic updates for clients that do not support the FQDN option.
3. The server is running Windows 2000 and configured not to register DNS resource records.

If the Windows 2000–based DHCP client requests that the server updates the PTR resource record but not the A resource record, what happens depends on the type of server and how it is configured. The server can perform any of the following actions:

1. If the server is running either Windows NT 4.0 or Windows 2000 and is configured not to perform dynamic updates, the server does not reply using the FQDN option and does not update either record. If this happens, the DHCP client attempts to update both the A and PTR resource records.
2. If the server is running Windows 2000 and is configured to update according to the request of the client, the server attempts to update the PTR resource record. The server sends a DHCPACK message to the client. The client then attempts to update the A resource record.
3. If the server is running Windows 2000 and is configured to always update both records, the server attempts to update both resource records. It sends a DHCPACK message to the client. If the client requested that the server update the PTR resource record but not the A resource record, the server also sets the Flags field to 3. In this case, the client does not attempt to update either resource record.

Configuring Dynamic Update for DHCP Clients and Servers

By default, Windows 2000 DHCP clients are configured to send the FQDN option with the Flags field set to 0, to request that the client register the A resource record and the server register the PTR resource record. The name used in the DNS registration is a concatenation of the host name and the primary DNS suffix of the computer. You can change this default from within the TCP/IP properties of your network connection.

Note From this page, you can specify whether to use the connection-specific DNS suffix in DNS registration and whether to register the

connection's IP address at all.

To change the dynamic update defaults on the dynamic update client

1. Right-click **My Network Places**, and then click **Properties**.
2. Right-click the connection you want to configure, and then click **Properties**.
3. Select **Internet Protocol (TCP/IP)**, click **Properties**, and click **Advanced**, and select the **DNS** tab.
4. By default, **Register this connection's address in DNS** is selected and **Use this connection's DNS suffix in DNS registration** is not selected, causing the client to request that the server update the PTR resource record and the client updates the A resource record using the primary DNS suffix.

   To configure the client to register the connection-specific DNS suffix as well as the primary DNS suffix, select **Use this connection's DNS suffix in DNS registration**.

   To configure the client not to register its IP address in DNS, deselect **Register this connection's addresses in DNS**.

You can configure the Windows 2000 DHCP server to do one of the following: update whichever records the client requests that it update; always update both A and PTR resource records, regardless of the request of the client; or to not update any DNS records.

To configure dynamic update for the Windows 2000 DHCP server

1. Click **Start**, point to **Programs** and **Administrative Tools**, and then click **DHCP**.
2. Expand the tree next to the name of the server.
3. Right-click the scope you're configuring, and then click **Properties**.
4. Click the **DNS** tab.
5. If it is not already selected, select **Automatically update DHCP client information in DNS**.
6. If you want the server to register whichever records the client requested that it register, select the option **Update DNS only if DNS client requests**.
7. If you want the server to always register both A and PTR resource records, select the option **Always update DNS**.
8. If you want the server to always register both A and PTR resource records on behalf of clients that do not support the FQDN option, select **Enable updates for DNS clients that do not support dynamic update**.

Caution If you have any multihomed dynamic update clients and at least one adapter is using DHCP, select the option **Update according to client request** (the default). If the DHCP server is configured to register both A and PTR resource records, the DHCP server replaces all A resource records for the name it attempts to register.

To update A or PTR resource records, the DHCP server sends a dynamic update request to the DNS server. If the DHCP server updated an A or PTR resource record, it removes that record when the lease of the client expires. You can also configure the server to remove the A resource record of the client when the lease of the client expires, even if the DHCP client and not the server registered the A resource record. When the DHCP lease is renewed, DHCP clients re-register their resource records.

To configure the Windows 2000 DHCP server to remove A resource records when the lease expires

1. Click **Start**, point to **Programs** and **Administrative Tools**, and then click **DHCP**.
2. Expand the tree next to the name of the server.
3. Right-click the scope you're configuring, and then click **Properties**.
4. Click the **DNS** tab.
5. Select the option **Discard forward (name-to-address) lookups when leases expire**.

For more information about the FQDN option and integration between DNS and DHCP, see the Internet Engineering Task Force (IETF) link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. Search for the IETF Internet-Draft "Interaction Between DHCP and DNS."

Statically Configured and Remote Access Clients

Statically configured clients and remote access clients do not rely on the DHCP server for DNS registration. Statically configured clients dynamically update their A and PTR resource records every time they start, or every 24 hours if the computer stays up longer than a day, in case the records become corrupted or need to be refreshed in the DNS database. Remote access clients dynamically update A and PTR resource records when a dial-up connection is made. They also attempt to deregister the A and PTR resource records when the user closes down the connection. However, if a remote access client fails to deregister a resource record within four seconds, it closes the connection, and the DNS database will contain a stale record. If the remote access client fails to deregister a resource record, it adds a message to the event log, which you can view by using Event Viewer. The remote access client never deletes stale records. However, the RRAS server attempts to deregister the PTR resource record when the client is disconnected.

Multihomed Clients

If a dynamic update client is multihomed (has more than one adapter and associated IP address), by default it registers the first IP address for each adapter. If you do not want it to register these IP addresses, you can configure it to not register IP addresses for one or more adapters from the properties page for the network connection.

To prevent the computer from registering an IP address for an adapter

1. Right-click **My Network Places**, and then click **Properties**.
2. Select the connection you want to configure, and then click **Properties**.
3. Select **Internet Protocol (TCP/IP)**, click **Properties**, click **Advanced**, and then select the **DNS** tab.
4. Clear the check box **Register this connection's address in DNS**.

The dynamic update client does not register all IP addresses with all DNS servers. For example, Figure 6.20 shows a multihomed computer, client1.noam.reskit.com, that is connected to both the Internet and the corporate intranet. Client1 is connected to the intranet by adapter A, a DHCP adapter with the IP address 172.16.8.7. Client1 is also connected to the Internet by adapter B, a remote access adapter with the IP address 10.3.3.9. Client1 resolves intranet names by using a name server on the intranet, NoamDC1, and resolves Internet names by using a name server on the Internet, ISPNameServer.

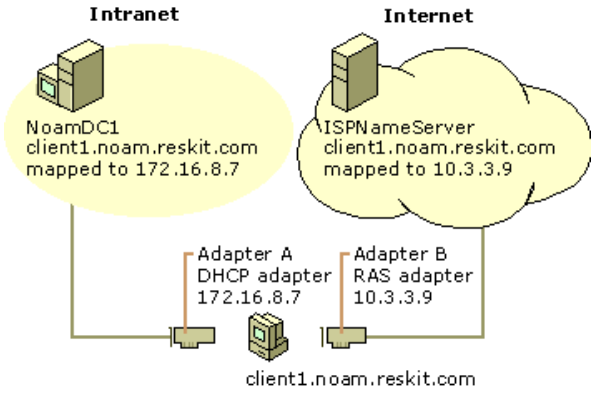


Figure 6.20 Dynamic Update for Multihomed Clients

Notice that although Client1 is connected to both networks, the IP address 172.16.8.7 is reachable only through adapter A, and the IP address 10.3.3.9 is reachable only through adapter B. Therefore, when the dynamic update client registers the IP addresses for Client1, it does not register both IP addresses with both name servers. Instead, it registers the name–to–IP address mapping for adapter A with NoamDC1 and the name–to–IP address mapping for adapter B with ISPNameServer.

By default, the computer registers a concatenation of the host name and primary DNS suffix. You can also configure the computer to register the domain name that is a concatenation of the host name and the connection-specific DNS suffix. For example, if you have a client that is connected to two different networks, and you want it to have a different domain name on each network, you can configure it to do so. For more information about configuring multiple domain names, see "Connection-Specific Domain Names" earlier in this chapter.

Time to Live

Whenever a dynamic update client registers in DNS, the associated A and PTR resource records include the TTL, which by default is set to 20 minutes. You can change the default setting by modifying the **DefaultRegistrationTTL** entry in the following registry subkey:

HKEY_LOCAL_MACHINE \SYSTEM \CurrentControlSet \Services \Tcpip \Parameters

The entry has a DWORD value and lists the TTL in seconds. A small value causes cached entries to expire sooner, which increases DNS traffic but decreases the risk of entries becoming stale. Expiring entries quickly is useful for computers that frequently renew their DHCP leases. A large value causes cached entries to be retained longer, decreasing DNS traffic but increasing the risk of entries becoming stale. Long retention times are useful for computers that renew their DHCP leases infrequently.

Caution Do not use a registry editor to edit the registry directly unless you have no alternative. The registry editors bypass the standard safeguards provided by administrative tools. These safeguards prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your system. Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. To configure or customize Windows 2000, use the programs in Control Panel or Microsoft Management Console (MMC) whenever possible.

Resolving Name Conflicts

If during dynamic update registration a client determines that its name is already registered in DNS with an IP address that belongs to another computer, by default the client attempts to replace the registration of the other computer's IP address with the new IP address. This means that for zones that are not configured for secure dynamic update, any user on the network can modify the IP address registration of any client computer. For zones that are configured for secure dynamic update, however, only authorized users are able to modify the resource record.

You can change the default setting so that instead of replacing the IP address, the client backs out of the registration process and logs the error in Event Viewer. To do so, add the **DisableReplaceAddressesInConflicts** entry with a value of 1 (DWORD) to the following registry subkey:

HKEY_LOCAL_MACHINE \SYSTEM \CurrentControlSet \Services \Tcpip \Parameters

The entry can be 1 or 0, which specify one of the following:

- *1*. If the name that the client is trying to create already exists, the client does not try to overwrite it.
- *0*. If the name that the client is trying to create already exists, the client tries to overwrite it. This is the default value.

Caution Do not use a registry editor to edit the registry directly unless you have no alternative. The registry editors bypass the standard safeguards provided by administrative tools. These safeguards prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your system. Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. To configure or customize Windows 2000, use the programs in Control Panel or Microsoft Management Console (MMC) whenever possible.

Secure Dynamic Update

You can configure any Active Directory–integrated zone for secure dynamic update, and then use the ACL to specify which users and groups have authority to modify the zone and records in the zone. The following sections describe the standards that comprise secure dynamic update, describe the secure dynamic update process, and explain how to configure secure dynamic update.

Note Secure dynamic update is available only on Active Directory–integrated zones.

Configuring Secure Dynamic Update

When you create an Active Directory–integrated zone, it is configured by default to allow only secure dynamic updates. If you created the zone as a standard primary zone and then converted it to an Active Directory–integrated zone, it is configured for non-secure dynamic updates or for no dynamic updates, depending on how the primary zone was previously configured.

To configure secure dynamic update

1. In the DNS console, right-click the zone for which you want to configure dynamic update, and then click **Properties**.
2. In the **Allow dynamic updates?** box, select **Only secure updates**.

Controlling Update Access to Zones

With secure dynamic update, only the computers and users you specify in an ACL can create or modify dnsNode objects within the zone. By default, the ACL gives Create permission to all members of the Authenticated User group, the group of all authenticated computers and users in an Active Directory forest. This means that any authenticated user or computer can create a new object in the zone. Also by default, the creator owns the new object and is given full control of it.

You can view and change the permissions for all DNS objects on the **Security** tab for the object, from within the Active Directory Users and Computers console or through the properties of zone and resource record in the DNS console.

To view the ACL for a dnsZone or dnsNode object

1. In the DNS console, right-click the zone or record you want to view, and then click **Properties**.
2. Click the **Security** tab.

Note ACLs are assigned on a per-name basis. Therefore, if you had two different records for the same FQDN, they map to the same object in Active Directory and have the same ACLs. For example, the following records have the same ACLs:

host1.reskit.com	A	172.16.15.9
host1.reskit.com	MX	mailer.reskit.com

Reserving Names

You can reserve FQDNs so that only certain users can use them. To do so, create the FQDN in the DNS console, then modify its ACL so that only particular computer, user, or users can change the set of records associated with the FQDN.

DNS Standards for Secure Dynamic Update

Windows 2000 supports secure dynamic updates through the Generic Security Service Application Program Interface (GSS-API, specified in RFC 2078) rather than Domain Name System Security Extensions (RFC 2535) or Secure Domain Name System Dynamic Update (RFC 2137). The GSS-API provides security services independently of the underlying security mechanism.

The GSS-API specifies a way to establish a security context by passing security tokens. The client generates the initial token and sends it to the server. The server processes the token and, if it is necessary, returns a subsequent token to the client. The process repeats until negotiation is complete and a security context has been established. After the security context has been established, it has a finite lifetime during which it can be used to create and verify the transaction signature on messages between the two parties.

Windows 2000 implements the GSS-API using an algorithm specified in the IETF Internet-Draft "GSS Algorithm for TSIG (GSS-TSIG)." This algorithm uses Kerberos v5 authentication protocol as its underlying security mechanism. Other security providers such as smart cards or certificates have not been tested. The algorithm uses the following resource records to provide security services:

TKEY. A resource record specified in the IETF Internet-Draft "Secret Key Establishment for DNS (TKEY RR)," as the vehicle to transfer security tokens between the client and the server and to establish secret keys to use with the TSIG resource record.

TSIG. A resource record specified in the IETF Internet-Draft "Secret Key Transaction Signatures for DNS (TSIG)," to send and verify signature-protected messages.

To see the TKEY and TSIG records being passed across the network, you can use Network Monitor. Versions 6.12 and later decode the resource records.

TKEY Resource Record

Table 6.7 describes the structure of the TKEY resource record, as described in the IETF Internet-Draft "Secret Key Establishment for DNS (TKEY RR)."

Table 6.7 TKEY Resource Record

Field	Data Type	Comment
NAME	domain name	Differs with mode and context
TTYPE	u_int16_t	TKEY
CLASS	u_int16_t	Ignored; should be zero
TTL	u_int32_t	Should be zero
RDLEN	u_int16_t	Length of **RDATA** field
RDATA		
Algorithm	domain name	Determines how the secret keying material exchanged by using the TKEY resource record is used to derive the algorithm-specific key
Inception	u_int	In number of seconds since January 1, 1970 GMT
Expiration	u_int32_t	In number of seconds since January 1, 1970 GMT
Mode	u_int16_t	Scheme for key agreement
Error	u_int16_t	Error code
Key size	u_int16_t	Size of **Key data** field in octets
Key data	octet stream	Differs with mode
Other size	u_int16_t	Not used
Other data	octet stream	Not used

TSIG Resource Record

Table 6.8 describes the structure of the TSIG resource record, as described in the IETF Internet-Draft "Secret Key Transaction Signatures for DNS (TSIG)," to send and verify signature-protected messages.

Table 6.8 Structure of TSIG Resource Record

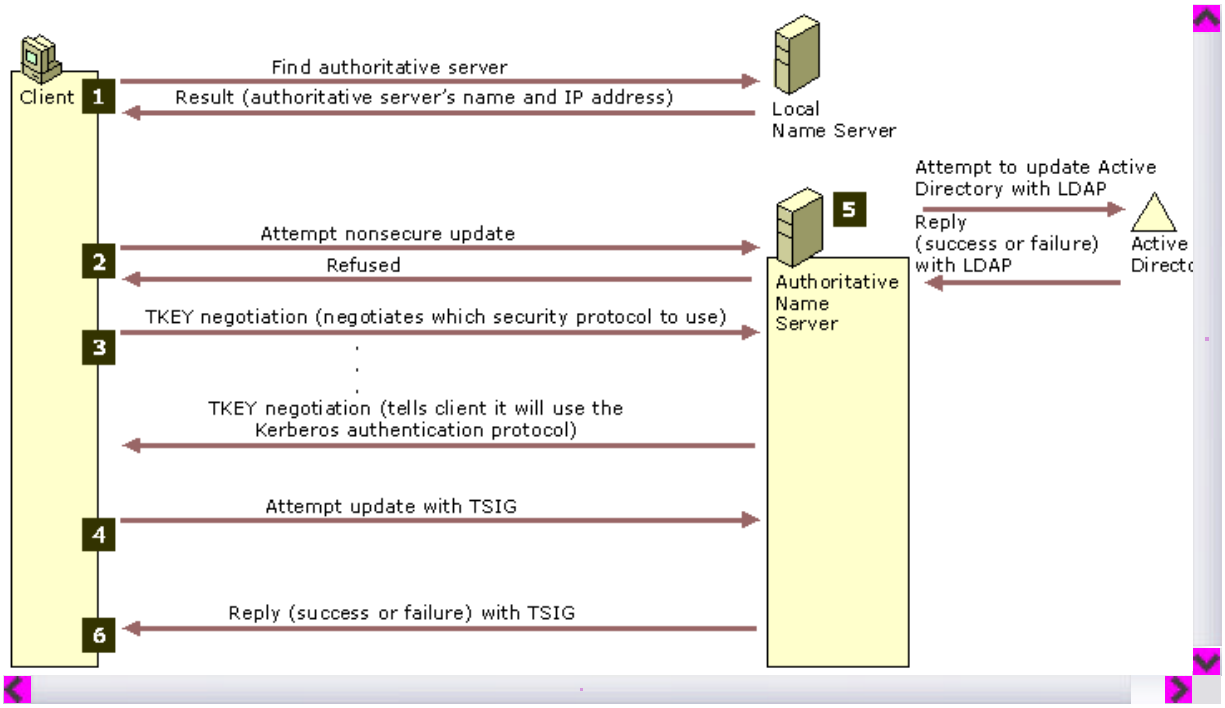
Field	Data Type	Comment

Algorithm name	domain name	Name of the algorithm, expressed as a domain name
Time signed	u_int48_t	Seconds since 1-Jan-70 UTC
Fudge	u_int16_t	Seconds of error permitted in **Time signed** field
Signature size	u_int16_t	Number of octets in **Signature** field
Signature	octet stream	Defined by **Algorithm name** field
Error	u_int16_t	Expanded RCODE covering signature processing
Other len	u_int16_t	Length, in octets, of **Other data** field
Other data	octet stream	Undefined

Secure Dynamic Update Process

To initiate a secure dynamic update, the client first initiates the TKEY negotiation process, to determine the underlying security mechanism and to exchange keys. Next, the client sends the dynamic update request containing resource records to add, delete, or modify to the server, signed with the TSIG resource record, and the server sends an acknowledgment. Finally, the server attempts to update Active Directory on behalf of the client.

Figure 6.21 shows the dynamic update process that takes place between a Windows 2000–based client and server, if both are configured with the default settings.



If your browser does not support inline frames, click here to view on a separate page.

Figure 6.21 Secure Dynamic Update Process

In step 1, the client queries the local name server to determine which server is authoritative for the name it is attempting to update (using the process described in "DNS Queries," found earlier in this chapter). The local name server responds with the name of the zone and the primary server that is authoritative for the zone.

In step 2, the client attempts a non-secure update, and the server refuses the non-secure update. Had the zone been configured for non-secure dynamic update rather than secure dynamic update, the server would have instead attempted to add, delete, or modify resource records in Active Directory.

In step 3, the client and server begin TKEY negotiation. First, the client and server negotiate an underlying security mechanism. Windows 2000 dynamic update clients and servers both propose the Kerberos protocol, so they decide to use it. Next, by using the security mechanism, they verify one another's identity and establish security context.

In step 4, the client sends the dynamic update request to the server, signed with the TSIG key that was generated by using the security context established in step 3. The DNS server verifies the origin of the dynamic update packet by using the security context and the TSIG key.

In step 5, the server attempts to add, delete, or modify resource records in Active Directory. Whether or not it can make the update depends on whether the client has the proper permissions to make the update and whether the prerequisites have been satisfied.

In step 6, the server sends a reply to the client stating whether or not it was able to make the update, signed with the TSIG key. If the client receives a spoofed reply, it throws it away and waits for a signed response.

Security for DHCP Clients That Do Not Support the FQDN Option

DHCP clients that do not support the FQDN option are not capable of dynamic updates. Therefore, if you want their A and PTR resource records dynamically registered in DNS, you must configure the DHCP server to perform dynamic updates on their behalf.

However, you do not want the DHCP server to perform *secure* dynamic updates on behalf of DHCP clients that do not support the FQDN option. If a DHCP server performs a secure dynamic update on a name, the DHCP server becomes the owner of that name, and only that DHCP server can update the name. This can cause problems in a few different circumstances. For example, suppose that the DHCP server DHCP1 created an object for the name nt4host1.reskit.com and then stopped responding, and that the backup DHCP server, DHCP2, tried to update the name; DHCP2 is not able to update the name because it does not own the name. In another example, suppose DHCP1 added an object for the name nt4host1.reskit.com, and then the administrator upgraded nt4host1.reskit.com to a Windows 2000–based computer. Because the Windows 2000–based computer did not own the name, it would not be able to update its own name.

Therefore, if you have enabled secure dynamic update, you might want to perform a special configuration for any DHCP server that will perform dynamic updates. Place the server in a special security group called DNSUpdateProxy. Objects created by members of the DNSUpdateProxy group have no security; therefore, any authenticated user can take ownership of the objects.

To add a DHCP Server to the DNSUpdateProxy group

1. Click **Start**, point to **Programs**, point to **Administrative Tools**, and then click **Active Directory Users and Computers**.
2. In the console tree, double-click the domain node.
3. Double-click the **Users** folder.
4. In the details pane, right-click the group and click **Properties**.
5. Click the **Members** tab, then click **Add**.
6. Click **Look in** to display a list of domains from which users and computers can be added to the group and click the domain containing the server you want to add.
7. Click the server to be added and then click **Add**.

Caution If you have installed the DHCP service on a domain controller, be absolutely certain not to make that server a member of the DNS Update Proxy group. Doing so would give any user or computer full control of the DNS records corresponding to the domain controllers, unless you manually modified the corresponding ACL. Moreover, if a DHCP server that is running on a domain controller is configured to perform dynamic updates on behalf of its clients, that DHCP server is able to take ownership of any record, even in the zones that are configured to allow only secure dynamic update. This is because a DHCP server runs under the computer account, so if it is installed on a domain controller it has full control over DNS objects stored in the Active Directory.

Windows 2000 DHCP clients register their own A resource records; therefore, putting a DHCP server in the DNSUpdateProxy group does not affect the security of the A resource records for Windows 2000 DHCP clients.

Note The A resource record corresponding to the DHCP server has no security if the server is placed in the DNSUpdateProxy group. However, you can manually modify the ACL through the DNS console.

For more information about interaction between DNS and DHCP, see the Windows 2000 Server Help.

Aging and Scavenging of Stale Records

With dynamic update, records are automatically added to the zone when computers and domain controllers are added. However, in some cases, they are not automatically deleted. For example, if a computer registers its own A resource record and is improperly disconnected from the network, the A resource record might not be deleted. If your network has many mobile users, this can happen frequently.

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

To solve these problems, the Windows 2000 DNS server can "scavenge" stale records; that is, it can search the database for records that have aged and delete them. Administrators can control aging and scavenging by specifying the following:

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

The DNS server uses an algorithm that ensures that it does not accidentally scavenge a record that must remain, provided that you configure all the parameters correctly. By default, the scavenging feature is off.

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

You can manually enable or disable aging and scavenging on a per-server, per-zone, or per-record basis. You can also enable aging for sets of records by using the command line tool Dnscmd.exe. (For information about Dnscmd.exe, see Windows 2000 Support Tools Help. For information about installing and using the Windows 2000 Support Tools and Support Tools Help, see the file Sreadme.doc in the directory \Support\Tools on the Windows 2000 operating system CD.) Keep in mind that if you enable scavenging on a record that is not a dynamic update record, the record will be deleted if it is not periodically refreshed, and you must recreate the record if it is still needed.

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

Aging and Scavenging Parameters

The Windows 2000 DNS server uses the timestamp that it gives each record, along with parameters that you configure, to determine when to scavenge records.

Table 6.9 lists the zone parameters that affect when records are scavenged. You configure these properties on the zone.

Table 6.9 Aging and Scavenging Parameters for Zones

Zone Parameter	Description	Configuration Tool	Notes
No-refresh interval	Time during which the server does not accept refreshes for the record. (The server still accepts updates.) This value is the interval between the last time a record was refreshed and the earliest moment it can be refreshed again.	DNS console and Dnscmd.exe	When an Active Directory–integrated zone is created, this parameter is set to the DNS server parameter **Default no-refresh interval**. This parameter replicates through Active Directory replication.
Refresh interval	The refresh interval comes after the no-refresh interval. At the beginning of the refresh interval, the server begins accepting refreshes. After the refresh interval expires, the DNS server can scavenge	DNS console and Dnscmd.exe	When an Active Directory–integrated zone is created, this parameter is set to the DNS server parameter **Default refresh interval**. This parameter is replicated by Active Directory.

	records that have not been refreshed during or after the refresh interval.		
Enable Scavenging	This flag indicates whether aging and scavenging is enabled for the records in the zone.	DNS console and Dnscmd.exe	When an Active Directory–integrated zone is created, this parameter is set to the DNS server parameter **Default enable scavenging**. This parameter is replicated by Active Directory.
ScavengingServers	This parameter determines which servers can scavenge records in this zone.	Only Dnscmd.exe	This parameter is replicated by Active Directory.
Start scavenging	This parameter determines when a server can start scavenging of this zone.	Not configurable	This parameter is not replicated by Active Directory.

Table 6.10 lists the server parameters that affect when records are scavenged. You set these parameters on the server.

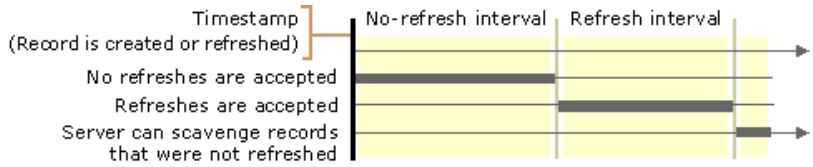
Table 6.10 Aging and Scavenging Parameters for Servers

Server Parameter	Description	Configuration Tool	Notes
Default no-refresh interval	This value specifies the no-refresh interval that is used by default for the Active Directory–integrated zone.	DNS console (shown as **No-refresh interval**) and Dnscmd.exe	By default, this is 7 days.
Default refresh interval	This value specifies the refresh interval that is used by default for the Active Directory–integrated zone.	DNS console (shown as **Refresh interval**) and Dnscmd.exe	By default, this is 7 days.
Default Enable Scavenging	This value specifies the Enable Scavenging parameter that is used by default for the Active Directory–integrated zone.	DNS console (shown as **Enable scavenging**)and Dnscmd.exe	By default, scavenging is disabled.
Enable scavenging	This flag specifies whether the DNS server can perform scavenging of stale records. If scavenging is enabled on a server, it automatically repeats scavenging as often as specified in the **Scavenging Period** parameter.	DNS console, Advanced View (shown as **Enable automatic scavenging of stale records**) and Dnscmd.exe	By default, scavenging is disabled.
Scavenging Period	This period specifies how often a DNS server enabled for scavenging can remove stale records.	DNS console, Advanced View (shown as **Scavenging Period**) and Dnscmd.exe	By default, this is 7 days.

Record Life Span

You can also invoke the Active Directory Installation wizard by executing an answer file that contains all of the settings that you need to configure. An *answer file* is a file that a wizard uses to provide answers to questions. For more information about the answer file for the Active Directory Installation wizard, see "Active Directory Data Storage" in the *Microsoft® Windows® 2000 Server Resource Kit Distributed Systems Guide*.

Figure 6.22 shows the life span of a scavengeable record.



If your browser does not support inline frames, click here to view on a separate page.

Figure 6.22 Life Span of a Scavengeable Record

When a record is created or refreshed on an Active Directory–integrated zone or on a standard primary zone for which scavenging is enabled, a timestamp is written.

Caution Because of the addition of the timestamp, a standard primary zone file for which scavenging is enabled is has a slightly different file format than a standard DNS zone. This does not cause any problems with standard zone transfer. However, you cannot copy a standard zone file for which scavenging is enabled to a non-Windows 2000 DNS server.

The value of the timestamp is the time the record was created or the record was last refreshed. By default, if the record is not dynamically updated, the timestamp equals zero, and the record is not scavengeable. Also, the timestamp is never changed if it contains a zero value. If the record belongs to an Active Directory–integrated zone, then every time the timestamp is refreshed, the record is replicated to the other domain controllers in the domain.

By default, the timestamps of records that are created by any method other than dynamic update are set to zero. A zero value indicates that the timestamp must not be refreshed and the record must not be scavenged.

After the record is refreshed, it cannot be refreshed again for the interval specified by the no-refresh interval. The no-refresh interval, a zone parameter, prevents unnecessary Active Directory replication traffic.

However, the record can still be updated during the no-refresh interval. If a dynamic update request requires modification to a record,

the request is considered an update. If the request requires no modifications, it is considered a refresh. Therefore, prerequisite-only updates, updates that include a list of prerequisites but no zone changes, are also considered refreshes.

The no-refresh interval is followed by the refresh interval. After the expiration of the no-refresh interval, the server begins to accept refreshes, and the server continues to accept refreshes for the life span of the record. The record can be refreshed as long as the current time is greater than the value of the timestamp plus the no-refresh interval. When the server accepts a refresh or an update, the value of the timestamp changes to the current time.

Next, after the expiration of the refresh interval, the server can scavenge the record if it has not been refreshed. The record can be scavenged if the current time is greater than the value of the timestamp plus the value of the no-refresh interval plus the value of the refresh interval. However, the server does not necessarily scavenge the record at that time. The time at which records are scavenged depends on several server parameters.

Server Behavior

You can configure the server to perform scavenging automatically, using a fixed frequency. In addition, you can manually trigger scavenging on a server to perform immediate scavenging. When scavenging starts, the server attempts to scavenge all primary zones and succeeds if all the following conditions are met:

- The **EnableScavenging** parameter is set to **1** on the server.
- The **EnableScavenging** parameter is set to **1** on the zone.
- Dynamic update is enabled on the zone.
- The zone parameter **ScavengingServers** is not specified or contains the IP address of this server.
- The current time is greater than the value of the zone parameter **StartScavenging**.

Note The zone parameter ScavengingServers is configurable only by using Dnscmd.exe. For more information about Dnscmd.exe, see Windows 2000 Support Tools Help.

The server sets **StartScavenging** whenever any of the following events occur:

- Dynamic update is turned on.
- **EnableScavenging** is set from **0** to **1** on the zone.
- The zone is loaded.
- The zone is resumed.

StartScavenging is equal to the time that one of the preceding events occur plus the amount of time specified in the refresh interval for the zone. This prevents a problem that can occur if the client is unable to refresh records because the zone isn't available — for example, if the zone is paused or the server is not working. If that happens and the server does not use **StartScavenging**, the server could scavenge the zone before the client has a chance to update the record.

When the server is ready to scavenge records, it examines all the records in the zone one by one. If the timestamp is not zero and the current time is later than the time specified in the timestamp for the record plus the no-refresh interval plus the refresh interval for the zone, it deletes the record.

Configuring Scavenging Parameters

This section discusses issues you must consider when configuring scavenging parameters.

To ensure that no records are deleted before the dynamic update client has time to refresh them, make certain that the refresh interval is greater than the refresh period for each record within a zone. Many different services might refresh records at different intervals; for example, Netlogon refreshes records once an hour, cluster servers generally refresh records every 15 to 20 minutes, DHCP servers refresh records at renewal of IP address leases, and Windows 2000–based computers refresh their A and PTR resource records every 24 hours.

Usually, the DHCP service requires the longest refresh interval of all services. If you are using the Windows 2000 DHCP service, you can use the default scavenging and aging values. If you are using another DHCP server, you might need to modify the defaults.

The longer you make the no-refresh and refresh intervals, the longer stale records remain. Therefore, you might want to make those intervals as short as is reasonable. However, if you make the no-refresh interval too short, you might cause unnecessary replication by Active Directory.

Integration with WINS

Windows Internet Name Service (WINS) provides dynamic name resolution for the NetBIOS namespace. Before Windows 2000, WINS was required on all clients and servers. The Windows NT 4.0 DNS server provided a feature called WINS lookup. With *WINS lookup*, you can direct DNS to query WINS for name resolution, so that DNS clients can look up the names and IP addresses of WINS clients. Windows 2000 still supports WINS lookup, although for DHCP clients, you can use dynamic update instead, provided that the DHCP server is running Windows 2000.

For more information about dynamic update, see "Dynamic Update and Secure Dynamic Update" earlier in this chapter. For more information about WINS, see "Windows Internet Name Service" in this book.

Note WINS is not required in a purely Windows 2000 environment.

To use WINS lookup integration, you add two special resource records — the WINS and WINS-R resource records — to your forward and reverse lookup zones, respectively. When a DNS server that is authoritative for that zone is queried for a name that it does not find in the authoritative zone, and the zone is configured to use WINS resolution, the DNS server queries the WINS server. If the name is registered with WINS, the WINS server returns the associated record to the DNS server.

Reverse lookups work slightly differently. When an authoritative DNS server is queried for a nonexistent PTR record, and the authoritative zone contains the WINS-R record, the DNS server uses a NetBIOS node adapter status lookup.

Finally, the DNS server returns the name or IP address in response to the original DNS request. Thus, DNS clients do not need to know whether a client is registered with WINS or DNS, nor do they need to query the WINS server.

Note For fault tolerance, you can specify multiple WINS servers. The server that is running the Windows 2000 Server DNS service tries to locate the name by searching the WINS servers in the order specified by the list.

Format of WINS and WINS-R Resource Records

The WINS resource record is used for forward lookups. When a resolver sends a query to the DNS server, requesting the corresponding A resource record, and the DNS server does not find the name in the forward lookup zone, it uses the WINS record to locate a WINS server that might be authoritative for the leftmost label of the FQDN. If present, the WINS record only applies for the topmost level within a zone and not for subdomains used in the zone. A WINS resource record has the following syntax:

<*domain*> <*class*> WINS [<*TTL*>] <*Local*> <*LookupTimeout*> <*CacheTimeout*> <*IP address of WINS server*>

where the placeholders have the following meanings:

domain. Domain name where the WINS record is found. It is always @.

class. Class is always IN for WINS records.

TTL. Time that a WINS record can be cached before it must be discarded.

Local. Specifies whether the record must be included in zone replication.

LookupTimeout. Time in seconds that a DNS server that uses WINS lookup waits before it gives up.

CacheTimeout. Time in seconds that a DNS server that uses WINS lookup can cache the response from the WINS server.

WINSServers. List of IP addresses of the WINS servers to be used.

The following is an example of a WINS resource record:

```
@ IN WINS LOCAL 5 3600 172.16.72.3
```

The WINS-R resource record is used for reverse lookups. When a resolver sends a query to the DNS server, requesting the corresponding PTR resource record, and the DNS server does not find the name in the authoritative reverse lookup zone, it uses a NetBIOS adapter node status query for the queried IP addresses. A WINS-R resource record has the following syntax:

<*domain*> <*class*> WINSR [<*TTL*>] <*Local*> <*LookupTimeout*> <*CacheTimeout*> <*NameResultDomain*>

where the placeholders have the following meanings:

domain. Domain name where the WINS record is found. It is always @.

class. Class the field is IN.

TTL. Time that a WINS record can be cached before it must be discarded.

Local. Specifies whether the record must be included into zone replication.

LookupTimeout. Time in seconds that a DNS server that uses WINS lookup waits before it gives up.

CacheTimeout. Time in seconds that a DNS server that uses WINS lookup can cache the response from the WINS server.

NameResultDomain. Domain to append to returned NetBIOS names.

The following is an example of a WINS-R resource record::

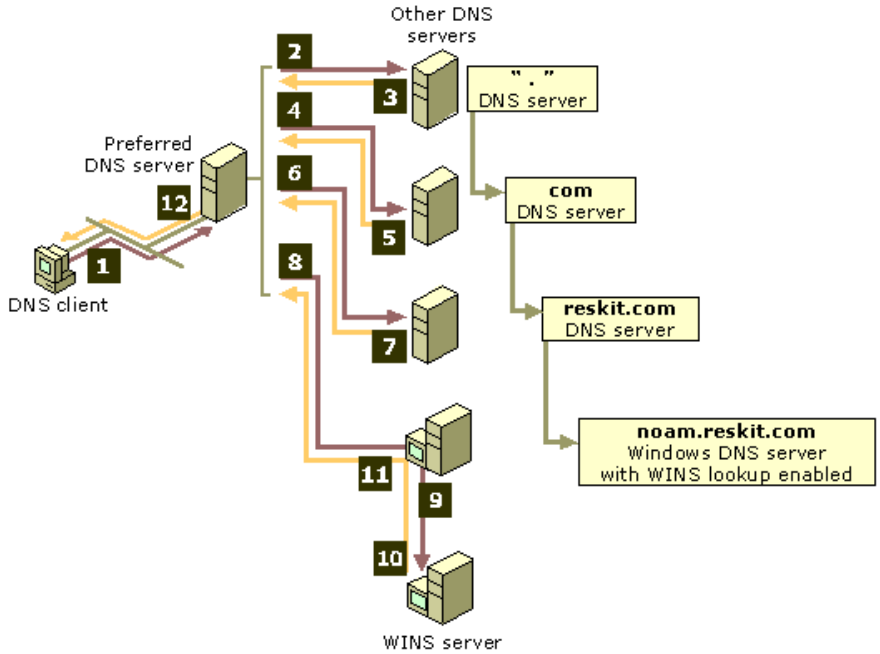
```
@ IN WINS-R LOCAL 5 3600 reskit.com.
```

Example of WINS Lookup

Suppose a user at a client workstation issues the following command:

net use \\host-a.noam.reskit.com.\public

This command establishes a connection between the client workstation and the Public folder on the computer host-a.noam.reskit.com, which is a client that is running Windows NT 4.0. However, before the connection can be established, the FQDN host-a.noam.reskit.com must be resolved by DNS — or, in this case, WINS — to an IP address. Figure 6.23 shows how this name is resolved, assuming that no server has cached the data and that no server is forwarding queries.



If your browser does not support inline frames, click here to view on a separate page.

Figure 6.23 Example of WINS Lookup

1. The client queries its preferred DNS server.
2. DNS servers carry out the normal process of recursion as the preferred DNS server queries other DNS servers in succession on behalf of the client. This process concludes at Step 8, when the DNS server for the noam.reskit.com zone is located through the previous chain of referral answers. At this point in the process, the server that is contacted is a Windows DNS server that is running either Windows NT Server 4.0 or Windows 2000 Server.

   When the Windows DNS server authoritative for the noam.reskit.com zone receives the query for "host-a," it looks in its configured

zone to see whether a matching A resource record can be found. If no A resource record is found and the zone is configured to use WINS lookup, the server does the following:

3. The DNS server separates the host part of the name (host-a) from the FQDN contained in the DNS query.

   The host part of the name is the first label.

4. The server then sends a NetBIOS name request to the WINS server using the host name host-a.
5. If the WINS server can resolve the name, it returns the IP address to the DNS server.
6. The Windows DNS server then returns this IP address information to the original preferred DNS server that was queried by the requesting client.
7. The preferred DNS server then passes the query answer back to the requesting client.

The client workstation establishes the session with host-a.noam.reskit.com and connects to the public folder.

In this example, only the last name server in the referral chain had knowledge of WINS. To the client resolver and all other name servers, it appears that DNS was responsible for the entire name resolution process. Furthermore, if the IP address changes for host-a.noam.reskit.com, WINS automatically handles it. Nothing needs to change with DNS.

A reverse lookup with WINS integration works a little differently than the previous example. Because the WINS database is not indexed by IP address, the DNS server cannot send a reverse name lookup to a WINS server to get the name of a computer given its IP address. The DNS server instead sends a node adapter status request directly to the IP address implied in the DNS reverse query. When the DNS server gets the NetBIOS name from the node status response, it appends the DNS domain name specified in the WINS-R record to the NetBIOS name provided in the node status response and forwards the result to the requesting client.

Configuring WINS Lookup

You can configure WINS lookup on both primary and secondary servers. To configure WINS lookup on a DNS server, perform the following steps:

To configure WINS lookup

1. In the DNS console, right-click the zone for which you are enabling WINS lookup, and then click **Properties**.
2. In the **Properties** dialog box for the zone, click the **WINS** tab.
3. Select the **Use WINS forward lookup** check box.
4. Under **IP address**, type the WINS Server IP address that will be used for resolution, and then click **Add**.

Repeat the procedure to add any other desired WINS servers. You can configure WINS lookup on both primary and secondary servers. You might want to configure WINS lookup on a secondary server, for example, if your primary and secondary servers are located at different sites and you want the secondary server to use local WINS servers. If you do so, however, you must disable replication from the primary server by clicking the check box **Do not replicate this record** on the **WINS** tab of the properties page for the zone.

Caution The WINS and WINS-R resource records are proprietary to the DNS service provided by Windows 2000 Server and earlier versions of Windows NT Server. It is best to make sure that all DNS servers that are authoritative for a zone are running Windows 2000 or any version of Windows NT; otherwise, resolvers can only look up WINS and WINS-R records intermittently. If you do make a server that is running another implementation of DNS authoritative for the zone, you must prevent these resource records from being included in zone transfers to other DNS server implementations by clicking the check box **Do not replicate this record** on the **WINS** tab of the zone properties. For more information about disabling WINS replication, see "WINS Lookup Interoperability Considerations" later in this chapter.

Advanced Parameters for WINS Lookups

You can use the following advanced timing parameters with the WINS and WINS-R records:

Cache Time-out Indicates to a DNS server how long to cache any of the information returned in a WINS lookup. By default, this value is set to 15 minutes.

Lookup Time-out Specifies how long to wait for a response from a WINS server before timing out and querying the next WINS server specified in a WINS record. By default, this value is 2 seconds.

You configure these parameters by using the Advanced button in the **Properties** dialog box for the zone. This button appears on either the WINS or WINS-R tabbed sheet, depending on whether the zone you are configuring is being used for forward lookup or reverse lookup.

Interoperability with Other DNS Servers

Windows 2000 DNS is RFC-compliant and interoperates with other DNS implementations. It has been tested to work with Windows NT 4.0, BIND 8.2, BIND 8.1.2, and BIND 4.9.7. However, Windows 2000 supports some features that other implementations of DNS do not support. Table 6.11 compares Windows 2000 to Windows NT 4.0, BIND 8.2, BIND 8.1.2, and BIND 4.9.7.

Table 6.11 Comparison of Features

Feature	Windows 2000	Windows NT 4.0	BIND 8.2	BIND 8.1.2	BIND 4.9.7
Support for the IETF Internet-Draft "A DNS RR for specifying the location of services (DNS SRV)." (SRV records)	Yes	Yes (with Service Pack 4)	Yes	Yes	Yes
Support for dynamic update	Yes	No	Yes	Yes	No
Support for secure dynamic update based on the GSS-TSIG algorithm	Yes	No	No	No	No
Support for WINS and WINS-R records	Yes	Yes	No	No	No
Support for fast zone transfer	Yes	Yes	Yes	Yes	Yes
Support for incremental zone transfer	Yes	No	Yes	No	No
Support for UTF-8 character encoding	Yes	No	No	No	No

The following sections describe issues to consider when implementing features that other DNS servers do not support, and describes how to set up DNS to support Active Directory when you are using third-party DNS servers.

Dynamic Update and Secure Dynamic Update Considerations

Clients and servers that are running versions of Windows earlier than Windows 2000 do not support dynamic update. However, Windows 2000 DHCP servers can perform dynamic updates on behalf of clients that do not support the FQDN option. If a Windows 2000 DHCP server must perform a secure dynamic update on behalf of clients that are running a version of Windows earlier than Windows 2000, you can place that DHCP server in a special security group called DNS Update Proxy. Objects created by the DNS Update Proxy group have no security, so they can be updated by any computer on the network.

For more information about these issues, see "Dynamic Update and Secure Dynamic Update" earlier in this chapter.

WINS Lookup Interoperability Considerations

For a zone that is configured for WINS lookup, WINS lookup works best if all authoritative servers are running Windows 2000 or Windows NT 4.0. WINS lookup requires the use of WINS and/or WINS-R resource records, two special, Windows-specific resource records. Computers that are running third-party implementations of DNS do not support WINS and WINS-R records. If you attempt to use a mixture of Microsoft and third-party DNS servers to host a zone, the mixture might cause data errors or failed zone transfers at the third-party DNS servers unless you configure the Windows 2000 server to disable replication of WINS and WINS-R records.

To disable replication of WINS and WINS-R records

1. In the DNS console, double-click your server to view its zones.
2. If you want to disable replication in a forward lookup zone, double-click the **Forward Lookup Zone** folder.

   -Or-

   If you want to disable replication in a reverse lookup zone, double-click the **Reverse Lookup Zone** folder.
3. Right-click the zone for which you want to disable replication of WINS and WINS-R records, and then click **Properties**.
4. Click the WINS tab.
5. Select the **Do not replicate this record** check box.

However, if you do disable replication of WINS and WINS-R records, queries directed at the primary and secondary servers return different results. When the authoritative primary server is queried for the name of a WINS client, it queries WINS, then returns the result to the client. However, when an authoritative secondary server is queried, it replies that the name was not be found.

The best way to prevent this problem is to configure your DNS servers to use WINS referral, described in the next section.

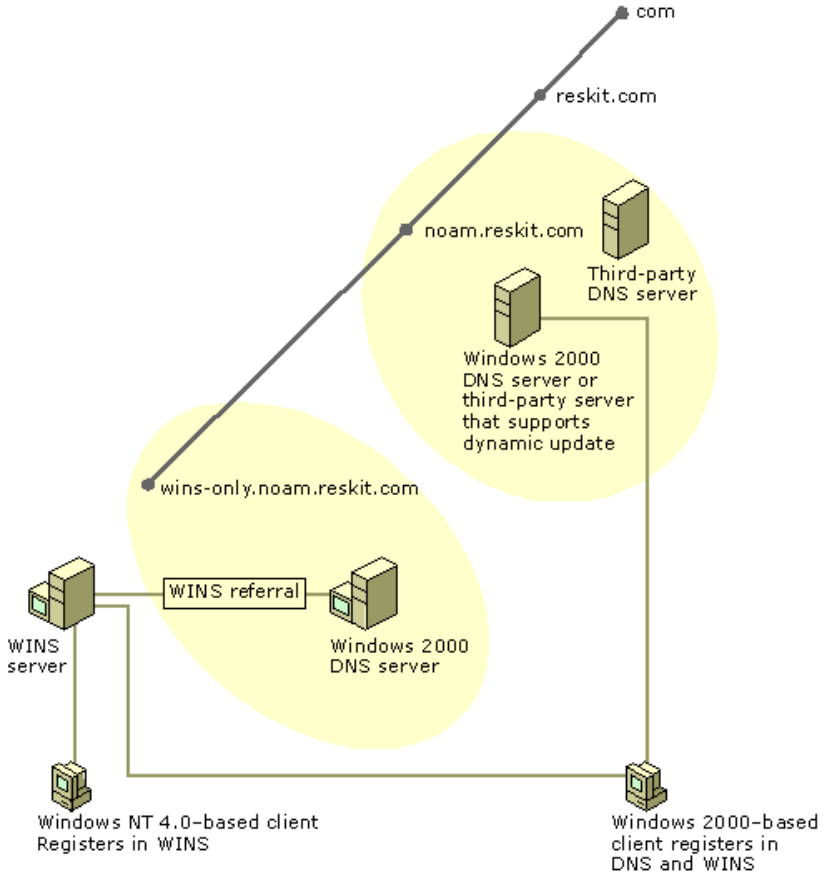
Using WINS Referral

If you have a domain that needs to contain WINS lookup resource records, but some of the authoritative name servers for that domain are running third-party DNS implementations, you can prevent interoperability problems by disabling WINS replication. Alternately, you can prevent interoperability problems by creating and delegating a WINS referral *zone*. This zone does not perform any registrations or updates, but only refers DNS lookups to WINS.

After you have created your WINS referral zone, you configure your DNS clients to append the WINS referral zone name to unqualified queries. The easiest way is to configure the DHCP server to assign a connection-specific DNS suffix to all DHCP adapters on all computers in your network. That suffix is appended to unqualified queries.

Alternatively, you can specify a domain suffix search list on each computer, as described in "Configuring the Resolver," earlier in this chapter. Keep in mind that when you specify a domain suffix search list, your primary DNS suffix and connection-specific DNS suffix are not used unless you specifically add them to the domain suffix search list.

Figure 6.24 shows an example of WINS referral in a network that includes servers that are running third-party implementations of DNS and that includes clients that are running both Windows 2000 and Windows NT 4.0.



If your browser does not support inline frames, click here to view on a separate page.

Figure 6.24 WINS Referral

In this example, the noam.reskit.com zone is stored and replicated between Windows 2000 servers and servers that are running other DNS implementations. To support WINS lookup, the network administrator created a new zone called wins-only.noam.reskit.com that is devoted to providing DNS-WINS integrated lookups for WINS clients. The network administrator enabled WINS lookup only on this zone, and did not add any resource records other than WINS resource records.

To register in DNS, Windows 2000–based clients send dynamic update requests to a DNS server that is authoritative for the domain noam.reskit.com. Both Windows 2000 and Windows NT 4.0–based clients register in WINS as well.

In this example, when a Windows 2000–based DNS client looks up a computer by its short name, it appends all the domain suffixes that it is configured to append, including the domain suffix wins-only.noam.reskit.com to produce the FQDN. For example, if the WINS host queried for is host-a, the client would use a DNS query for host-a.wins-only.noam.reskit.com.

Having only one WINS-integrated zone provides other advantages as well. When a DNS forward lookup for the host name of a computer uses WINS lookup, the DNS name specified and used in the query explicitly indicates that the source used to resolve the name was a DNS server that uses WINS lookup integration. This integrated solution can also prevent the confusing situation in which DNS queries for different FQDNs resolve to the same WINS client name and IP address. This result can easily occur if you add and configure multiple zones and enable each of them to use WINS lookup integration.

For example, suppose you have two zones, both configured to use WINS lookup. The zones are rooted and originate at the following DNS domain names:

- noam.reskit.com.
- eu.reskit.com.

With this configuration, a query for a WINS client named host-a can be resolved by using either of the following FQDNs:

- host-a.noam.reskit.com.
- host-a.eu.reskit.com.

Zone Transfer Considerations

Windows 2000 supports a method of zone transfer called fast zone transfer. With *fast zone transfer*, the Windows 2000 DNS server can send more than one resource record per message. This is more efficient than sending only one. However, some third-party DNS servers, including servers that are running versions of BIND earlier than 4.9.5 do not support fast zone transfer. If you use a secondary server that does not support fast zone transfer, disable fast zone transfers on the master server by selecting the check box **Bind secondaries** on the Advanced tab of the properties for your server, from within the DNS console.

Many DNS servers, including servers that are running versions of BIND earlier than 8.2, do not support *incremental zone transfer*, another method of zone transfer. With incremental zone transfer, instead of transferring a whole zone, a DNS server can transfer only those portions of the zone that changed since the last time the secondary server queried. However, this does not cause interoperability problems, because Windows 2000 can still use full zone transfer if any of the secondary servers do not support incremental zone transfer.

Windows 2000 also supports resource record types that other servers might not support, such as the WINS record and the WINS-R record. If you have a primary copy of a zone on a Windows 2000 DNS server and a secondary copy of a zone on a third-party DNS server, and the primary zone includes resource records the third-party server does not support, the secondary server might drop those resource records, or it might not be able to transfer the zone. For information about WINS records, see "WINS Considerations" earlier in this chapter.

It is also possible that a third-party DNS server might support a resource record type that Windows 2000 does not support, such as

resource records not listed in the RFCs. If you have a primary copy of the zone on a third-party DNS server and a secondary copy on a Windows 2000 server, and the primary zone includes resource records that the Windows 2000 DNS server does not support, the Windows 2000 DNS server drops those resource records. If it receives any circular CNAME records, it drops those as well. You can also configure your DNS server to halt a zone transfer when it receives a resource record it does not support.

For information about problems with zone transfer, see "Diagnosing Name Resolution Problems" later in this chapter.

Unicode Character Set Considerations

Windows 2000 supports RFC 2044, which enlarges the character set allowed in DNS names to include UTF-8 character encoding. However, many DNS servers, including Windows NT 4.0 DNS servers, follow RFC 1123, which permits a smaller character set. If you perform a zone transfer from a zone containing UTF-8 encoded characters to a third-party secondary server that does not support UTF-8 encoded characters, the secondary server might drop resource records, or the zone transfer might fail. Therefore, if you plan to use any characters from the UTF-8 character set, consider the issues described in "Naming Restrictions for Hosts and Domains," earlier in this chapter.

Configuring Non-Windows 2000 DNS Servers to Support Active Directory

For the domain controller locator to work properly, the primary DNS server that is authoritative for the names that are to be registered by the Netlogon service on the domain controller, must support the service location resource record (SRV RR). The SRV resource record is specified in the IETF Internet-Draft "A DNS RR for specifying the location of services (DNS SRV)." Other DNS servers that are authoritative for the domain must also support SRV records.

In addition, you can simplify administration by making sure that the DNS servers that are authoritative for the names that Netlogon registers support the dynamic update protocol, as described in RFC 2136. You can use as the primary master for the domain name a DNS server that does not support dynamic update. However, this is not recommended, because you will need to manually update the primary zone when you configure Active Directory. For information about how to configure and verify the DNS records that are used to support Active Directory, see "Verifying Your Basic DNS Configuration" later in this chapter.

If you are using a DNS server that does not support the IETF Internet-Draft "A DNS RR for specifying the location of services (DNS SRV)," you must upgrade your DNS server or add a DNS server that does supports those standards. The server supporting those standards must be the primary DNS server that is authoritative for the DNS names that will be registered by the Netlogon service on the domain controller. You must then perform special configuration on both DNS servers.

This section explains which DNS servers can be used to support Active Directory and how to configure DNS and Active Directory when you are using servers that cannot support Active Directory.

If you are using a DNS service other than the Windows 2000 DNS service, it is a good idea to test it for compatibility with Active Directory and DHCP.

Using Non-Microsoft DNS Servers to Support Active Directory

The following servers support SRV records:

- Windows 2000
- Windows NT 4.0 Service Pack 4 and later
- BIND 4.9.6 and later

The following servers support dynamic update:

- Windows 2000
- BIND 8

If you use a third-party server, however, you cannot use the DNS console or Dnscmd.exe, Active Directory integration, secure dynamic update, aging and scavenging of stale records, or remote administration.

Also, it is a good idea to verify your DNS configuration after you install Active Directory.

The DNS database must include locator resource records (SRV, CNAME, and A) to support each domain controller.

Using the Name of a Delegated Zone as an Active Directory Domain Name

If your organization already has a DNS domain (for example, reskit.com), and the primary DNS server that is authoritative for that domain does not support RFC 2136 and the IETF Internet-Draft "A DNS RR for specifying the location of services (DNS SRV)" — and you cannot upgrade the server to a server that does — you can still create an Active Directory domain. To provide DNS support for an Active Directory domain in such a situation, delegate a subdomain (for example, child.reskit.com) from your first DNS server to a second DNS server that does support these standards. Next, make that second DNS server authoritative for the subdomain, and create an Active Directory domain that has the same name as the DNS subdomain. Figure 6.25 shows an example of implementing the Windows 2000 DNS server and making it authoritative for a delegated subdomain.

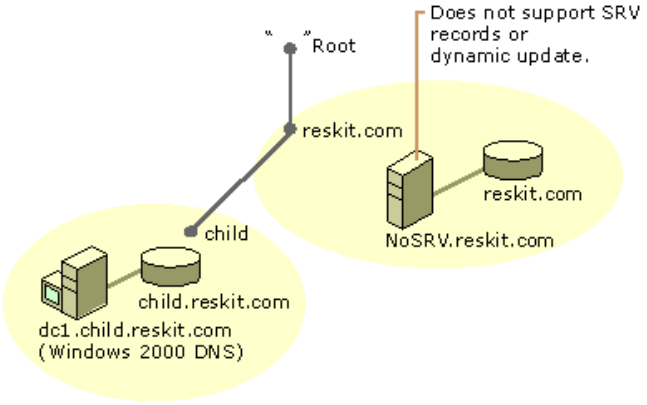


Figure 6.25 Implementing a Windows 2000 DNS Server to Support a Delegated Subdomain

In this example, the primary name server for reskit.com, NoSRV.reskit.com, does not support SRV records and, therefore, cannot be used to support Active Directory. Because of this, the administrator of NoSRV.reskit.com delegated the subdomain **child.reskit.com** to a Windows 2000 DNS server. The Windows 2000 DNS server provides the same capabilities for this zone as for any other zone. For example, it can be stored in Active Directory, as described in "Active Directory Integration and Multimaster Replication" earlier in this chapter.

Using the Existing Zone Name as the Active Directory Domain Name

If your organization already has a DNS domain (for example, reskit.com), and the DNS server that is authoritative for that domain does not support RFC 2136 and the IETF Internet-Draft "A DNS RR for specifying the location of services (DNS SRV)," and you cannot upgrade the server, you can still implement Active Directory with the name of the existing DNS zone. To implement Active Directory, add another DNS server that does support those standards and delegate certain zones to this server.

On the DNS server that does not support SRV records and dynamic update, delegate the following zones to the DNS server that does:

- _tcp.<*Active Directory domain name*>
- _udp.<*Active Directory domain name*>
- _msdcs.<*Active Directory domain name*>
- _sites. <*Active Directory domain name*>

On the DNS server that does support these features, create and then enable dynamic update on each of the zones in the preceding list. The domain controllers dynamically update the appropriate records in these zones.

Figure 6.26 illustrates this configuration for the example domain reskit.com:

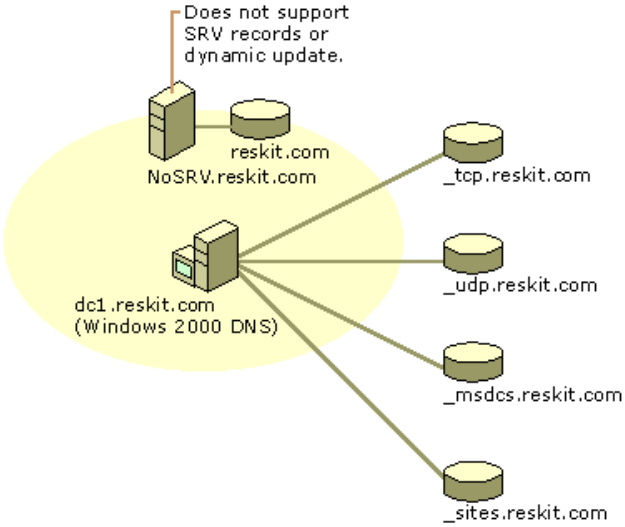


Figure 6.26 Delegating Zones to a DNS Server That Can Support Active Directory

The Netlogon service sends dynamic updates to the delegated zones. By default, the Netlogon service attempts a dynamic update of an A resource record that contains an *owner name* that is the same name as the Active Directory domain name. An owner name is the name of the node to which the resource record pertains. In this example, Netlogon fails the dynamic update; this causes an error message to appear in Event Viewer that says a dynamic update failed. To prevent Netlogon from registering A resource records, add the entry **DnsRegisterARecords** to the registry in the following registry subkey:

HKEY_LOCAL_MACHINE \SYSTEM \CurrentControlSet \Services \NetLogon \Parameters

Set the value of **DnsRegisterARecords** to 0x0 (DWORD).

Caution Do not use a registry editor to edit the registry directly unless you have no alternative. The registry editors bypass the standard safeguards provided by administrative tools. These safeguards prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your system. Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. To configure or customize Windows 2000, use the programs in Control Panel or Microsoft Management Console (MMC) whenever possible.

Internet Access Considerations

For your organization to be visible on the Internet, you must have an external namespace, a public namespace that anyone on the Internet can access. An Internet name authority must assign to you a DNS domain and make sure that a parent DNS zone includes a delegation to the DNS server that is authoritative for the DNS domain. However, to help prevent malicious access of your network, you can use an internal namespace, a private namespace that only users within your organization can see, preventing unauthorized people from learning the names and IP addresses of the computers on your network.

If you plan to have both an internal and external namespace, you must configure your DNS servers to enable internal clients to resolve names in both namespaces. How you plan your namespace depends on the type of clients that you have.

Planning Your Namespace

When planning your namespace, you must decide whether to use a private root and whether you want your internal and external namespaces to have the same domain name.

Whether you can use a private root depends on the type of clients you have. You can use a private root only if each of your clients has one of the following:

- *name exclusion list*. A list of DNS suffixes that are internal.
- *proxy autoconfiguration (PAC) file*. A list of DNS suffixes and exact names that are internal or external.

If you have clients lacking both of these, the DNS server hosting your organization's top-level internal domain must forward queries to the Internet.

Table 6.12 shows, based on the proxy capability of your client, whether you can use a private root. (Note that a *local address table* is a list of IP addresses that are internal and external.)

Table 6.12 Configuring Internal and External Namespaces Based on Proxy Capability

		Local Address Table		**Proxy Auto-configuration (PAC)**

	No Proxy	(LAT)	Name Exclusion List	File
Microsoft software with corresponding proxy capability	Generic Telnet	Windows Sockets Proxy (WSP) 1.*x*, WSP 2.*x*	WSP 1.*x*, WSP 2.*x*, and all versions of Microsoft® Internet Explorer.	WSP 2.*x*, Internet Explorer 3.01 and later.
Can you forward queries?	Must forward queries.	Must forward queries.	Possible.	Possible.
Can you use a private root?	Not possible.	Not possible.	Possible.	Possible.

To simplify name resolution for internal clients, use a different domain name for your internal and external namespaces. For example, you can use the name reskit01-ext.com for your external namespace and reskit.com for your internal namespace. You can also use the name reskit.com for your external namespace and noam.reskit.com for your internal namespace. However, do not make your external domain a subdomain of your internal domain; that is, in the context of this example, do not use reskit.com for your internal namespace and noam.reskit.com for your external namespace.

You can use the same name internally and externally, but doing so causes configuration problems and generally increases administrative overhead. If you want to use the same domain name internally and externally, you need to perform one of the following actions:

- Duplicate internally the public DNS zone of your organization.
- Duplicate internally the public DNS zone and all public servers (such as Web servers) that belong to your organization.
- In the PAC file on each of your clients, maintain a list of the public servers that belong to your organization.

Caution Make sure that the domain name for your internal namespace is not used anywhere on the Internet. Otherwise, you might have problems with ambiguity in the name resolution process.

Which action you need to perform to use the same domain name internally and externally varies. Table 6.13 shows whether you can use the same domain name for your internal and external namespaces, and if so, which method you must use, based on your client software proxy capability.

Table 6.13 Using the Same Name for Internal and External Namespaces Based on Proxy Capability

	No Proxy	Local Address Table (LAT)	Name Exclusion List	Proxy Auto-configuration (PAC) File
Use different domain names.	Possible.	Possible.	Possible.	Possible (using simple exclusion)
Use the same domain name; internally duplicating organization's public DNS namespace (records).	Possible.	Possible (by populating LAT).	Not possible.	Possible. When a PAC file is used, duplicated external records are not used.
Use the same domain name; internally duplicating organization's public DNS namespace and public servers.	Possible.	Possible.	Possible.	Possible.
Use the same domain name; maintaining list of public servers in the PAC files.	Not possible.	Not possible.	Not possible.	Possible.

Namespace Planning Example

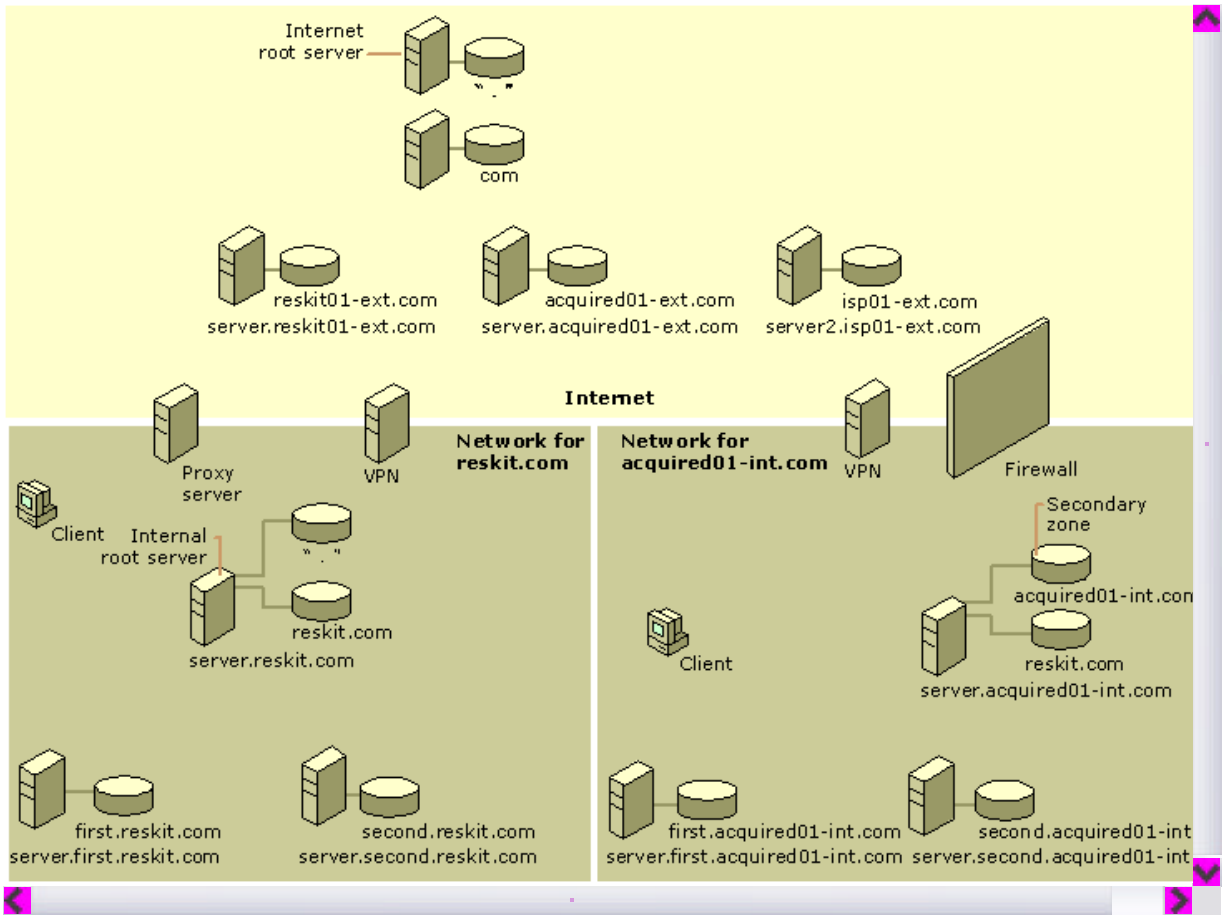
The following sections explain some of the issues you must consider when planning your namespace by describing the configuration of two fictitious organizations. The first organization, which has reserved the DNS domain names reskit.com and reskit01-ext.com, has only proxy clients that support either exclusion lists or PACs. In contrast, the second organization, which has reserved the DNS domain names acquired01-int.com and acquired01-ext.com, has no such proxy clients. Both organizations use a different domain name for their internal and external namespaces.

Reskit.com and acquired01-int.com both need a configuration that does the following:

- Exposes only the public part of the organization's namespace to the Internet.
- Enables any computer within the organization to resolve any internal or external name.
- Enables any computer within the organization to resolve any name from the Internet.

Moreover, both organizations have merged, and every computer from within each private namespaces must be able to resolve any name from the other namespace.

The following sections describe how both organizations have configured their external and internal namespaces to satisfy these requirements. Figure 6.27 shows this configuration.



If your browser does not support inline frames, click here to view on a separate page.

Figure 6.27 Example Configuration of the DNS Domains Reskit.com and Acquired01-int.com

Configuring the External Namespace

In the external namespace, two zones exist: reskit01-ext.com and acquired01-ext.com. The zones contain only the records (the names and delegations) that the companies want to expose to the outside world. The server server.reskit01-ext.com. hosts the zone reskit01-ext.com, and the server server.acquired01-ext.com hosts the zone acquired01-ext.com. The names reskit01-ext.com and acquired01-ext.com must be registered with an Internet name authority.

Configuring the Internal Namespace

The internal namespace for the organization that hosts reskit01-ext.com externally is reskit.com. Similarly, the internal namespace for the organization that hosts acquired01-ext.com externally is acquired01-int.com. The server server.reskit.com hosts the zone reskit.com, and the server server.acquired01-int.com hosts the zone acquired01-int.com. The names reskit.com and acquired01-int.com must be registered with an Internet name authority.

All the computers in reskit.com support either exclusion lists or PACs, and none of the computers in acquired01-int.com support either exclusion lists or PACs.

Namespace Without Proxy Clients That Support Exclusion Lists or PACs

For a namespace in which none of the computers are proxy clients that support either exclusion lists or PACs (in this example, the namespace of acquired01-int.com), an organization must devote one or more DNS servers to maintain zones that contain all names from the internal namespace. Every DNS client must send DNS queries to one or more of these DNS servers. If a DNS server contains the zone for the top level of the organization's namespace (for example, acquired01-int.com), then it must forward those queries through a firewall to one or more DNS servers in the Internet namespace. All other DNS servers must forward queries to one or more DNS servers that contain the zone for the top level of the organization's namespace.

To make sure that any client within the organization can resolve any name from the merged organization, every DNS server containing the zone for the top level of the organization's namespace must also contain the zones that include all the internal and external names of the merged organization.

This solution places a significant load on the internal DNS servers that contain the organization's internal top-level zones. Most of the queries generated within the organization are forwarded to these servers, including queries for computers in the external namespace and in the merged organization's private namespace. Also, the servers must contain secondary copies of the merged organization's zones.

Namespace with Proxy Clients That Support Exclusion Lists or PACs

For a namespace in which all of the computers are proxy clients that support either exclusion lists or PACs (for example, the namespace of reskit.com), the private namespace can include a private root. In the internal namespace, there can be one or more root servers, and all other DNS servers must include the name and IP address of a root server in their root hints files.

To resolve internal and external names, every DNS client must submit all queries to either the internal DNS servers or to a proxy server, based on an exclusion list or PAC file.

To make sure that every client within the organization can resolve every name from the merged organization, the private root zone must contain a delegation to the zone for the top level of the merged organization.

Using proxy clients and a private root simplifies DNS configuration because none of the DNS servers need to include a secondary copy of the zone. However, this configuration requires you to create and manage exclusion lists or PAC files, which must be added to every proxy client in the network.

Examples of Queries

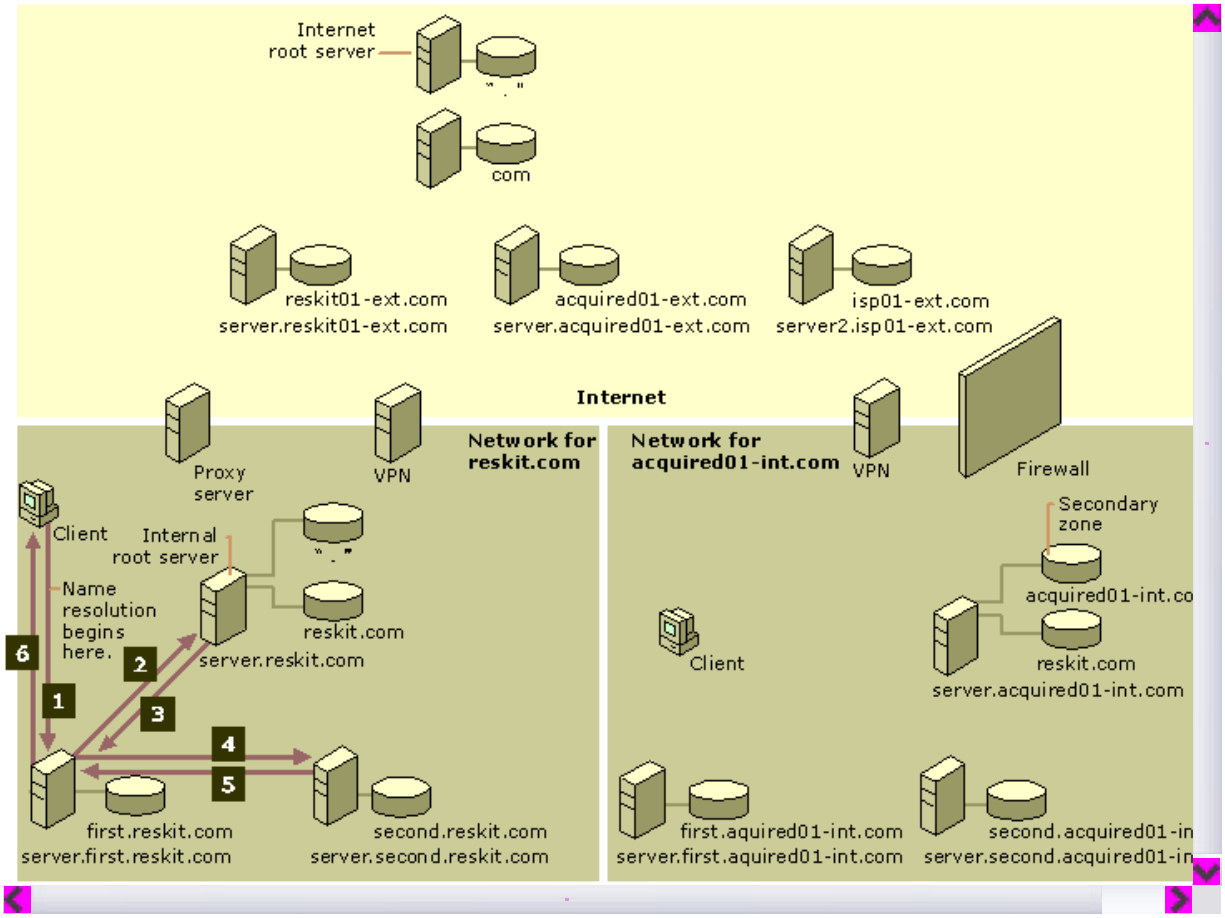
The following examples show how queries for the following names are resolved:

- Internal name
- Name on the Internet
- Name in the external namespace of an organization
- Name in the internal namespace of a merged organization

Note In all of these examples, no DNS server has cached the name for which the client is querying. An actual query might progress differently, because the name might be cached.

Query for a Name in the Internal Namespace

Suppose that a computer in reskit.com needs to resolve a DNS query for host.second.reskit.com. First, the computer consults its exclusion list or its PAC file and discovers that host.second.reskit.com is in the internal namespace. Therefore, the computer submits the query to a local DNS server. Figure 6.28 shows how the query proceeds.



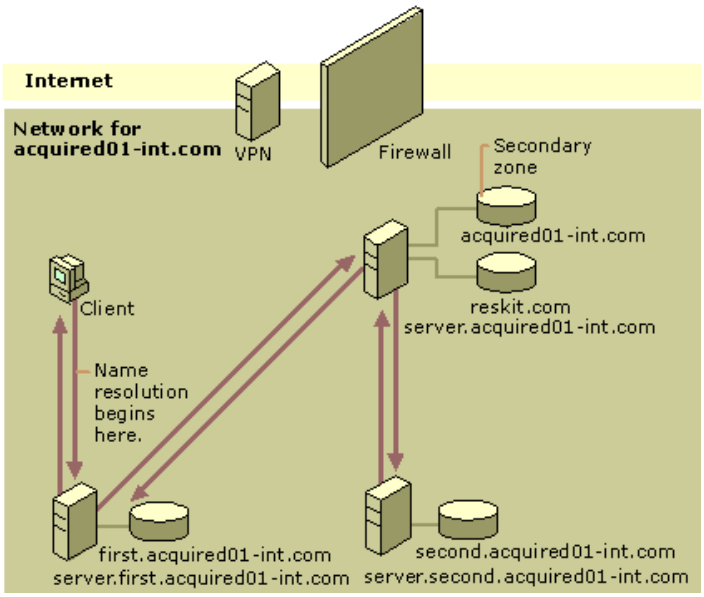
If your browser does not support inline frames, click here to view on a separate page.

Figure 6.28 Query for an Internal Name in the Domain Reskit.com

The query proceeds as follows:

1. The computer submits a query to its local DNS server, server.first.reskit.com.
2. If the local server is not authoritative for host.second.reskit.com, the local DNS server queries a root server.
3. The root server returns a reference to the authoritative server, server.second.reskit.com.
4. The local server, server first.reskit.com, queries server.second.reskit.com.
5. Server.second.reskit.com resolves the query and returns the response to the local server.
6. Server.first.reskit.com passes the response to the client.

Now suppose that a computer in acquired01-int.com needs to resolve a DNS query for host.second.acquired01-int.com. Figure 6.29 shows how the query proceeds.



If your browser does not support inline frames, click here to view on a separate page.

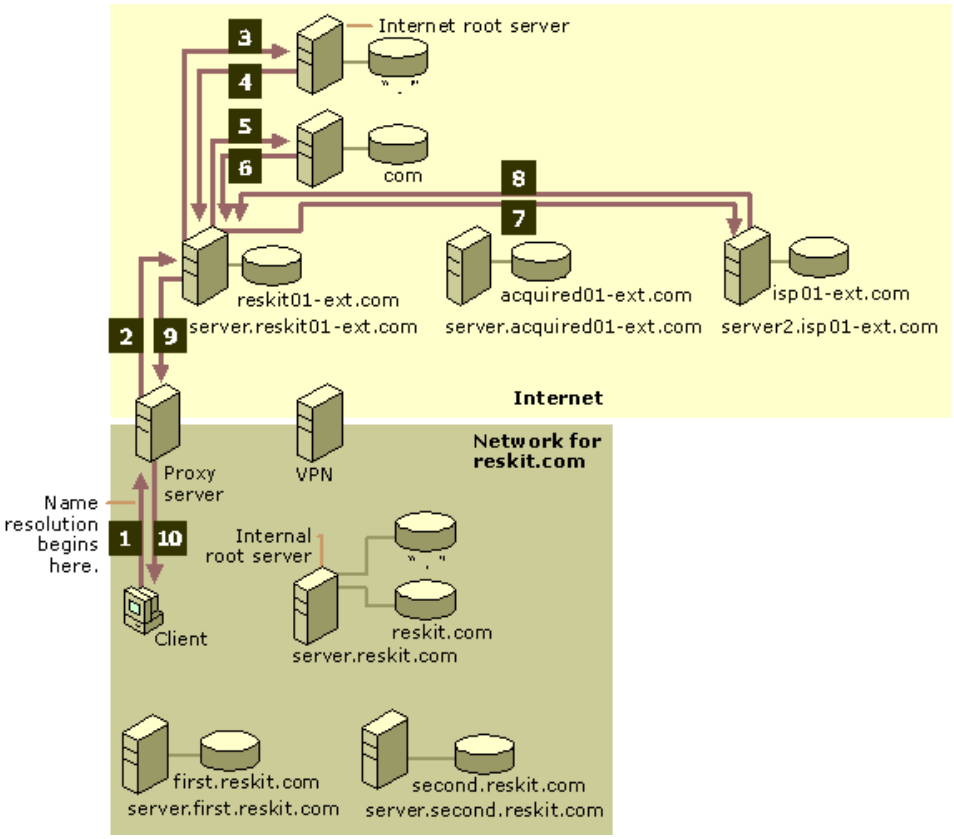
Figure 6.29 Query for an Internal Name in the Domain Acquired01-int.com

The query proceeds as follows:

1. The computer submits the query to its local DNS server, server.first.acquired01-int.com.
2. If the local server is not authoritative for host.second.acquired01-int.com, the local DNS server forwards the query to the DNS server that is authoritative for the acquired01-int.com zone.
3. The DNS server that is authoritative for the acquired01-int.com zone finds a delegation to the server server.second.acquired01-int.com and queries that server.
4. Server.second.acquired01-int.com resolves the query and returns the name to the DNS server authoritative for the acquired01-int.com zone.
5. The DNS server that is authoritative for the acquired01-int.com zone returns the name to the local DNS server.
6. Server.first.acquired01-int.com returns the name to the client.

Query for a Name in the External Namespace

Suppose that a computer in reskit.com needs to access a Web page on the computer host.isp01-ext.com. Figure 6.30 shows how the query proceeds.



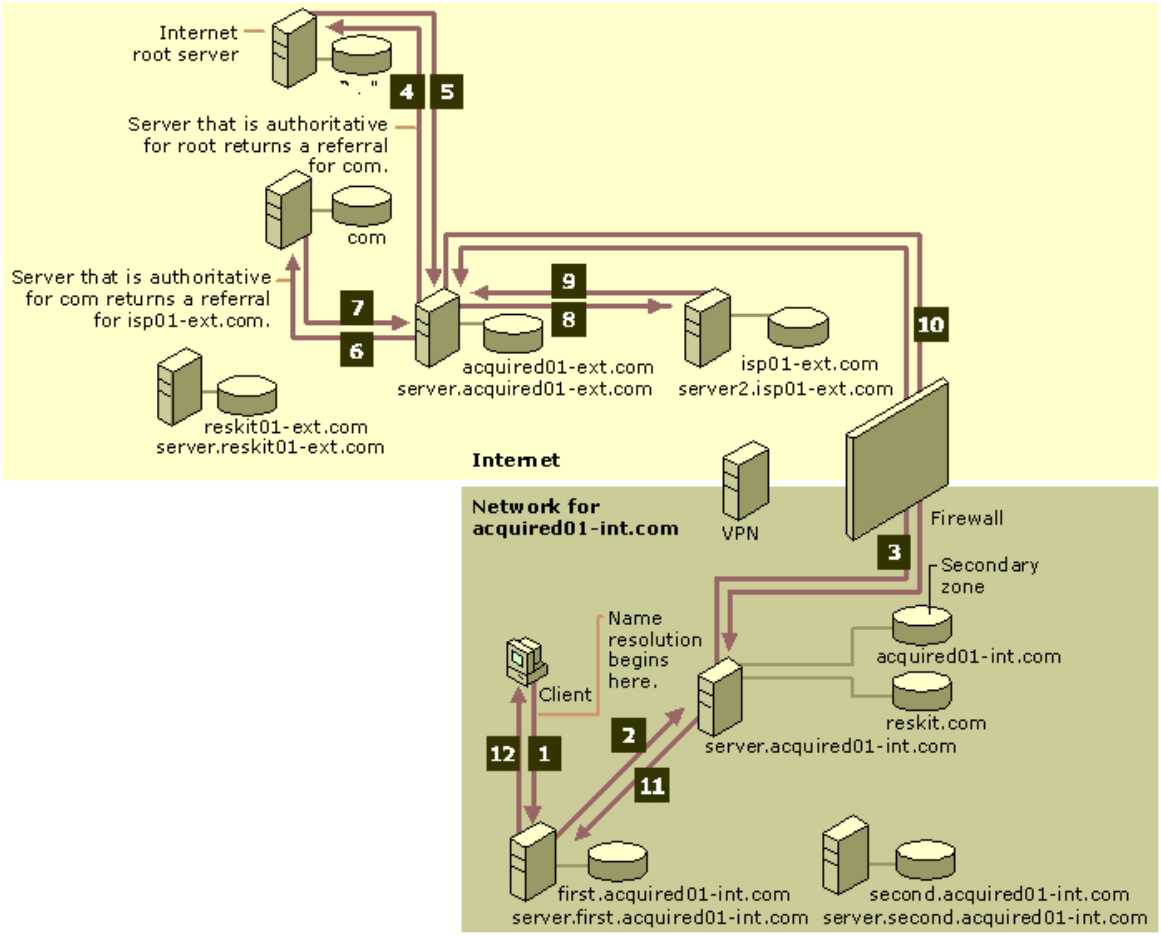
If your browser does not support inline frames, click here to view on a separate page.

Figure 6.30 Query in the Domain Reskit.com for a Name on the Internet

The query proceeds as follows:

1. Because the client is a proxy client, it consults its exclusion list or its PAC file and determines that the name is not in the internal namespace. Therefore, the client sends the request to the proxy server.
2. The proxy server sends a query to the DNS server to which it is configured to send queries. In this case, the server is server.reskit01-ext.com.
3. The server server.reskit01-ext.com sends a query to the Internet root server.
4. The Internet root server returns a referral to a server that is authoritative for the Internet zone com.
5. The server server.reskit01-ext.com queries the server that is authoritative for the com zone.
6. The server that is authoritative for the zone com returns a referral to the server that is authoritative for the zone isp01-ext.com.
7. The server server.reskit01-ext.com queries the server that is authoritative for the zone isp01-ext.com.
8. The server that is authoritative for the zone isp01-ext.com returns the IP address that corresponds to the name host.isp01-ext.com.
9. The server server.reskit01-ext.com returns the response to the proxy server.
10. The proxy server uses the IP address to contact host.isp01-ext.com and provides necessary information to the client.

Now suppose that a computer in acquired01-int.com needs to resolve a DNS query for host.isp01-ext.com. Figure 6.31 shows how the query proceeds.



If your browser does not support inline frames, click here to view on a separate page.

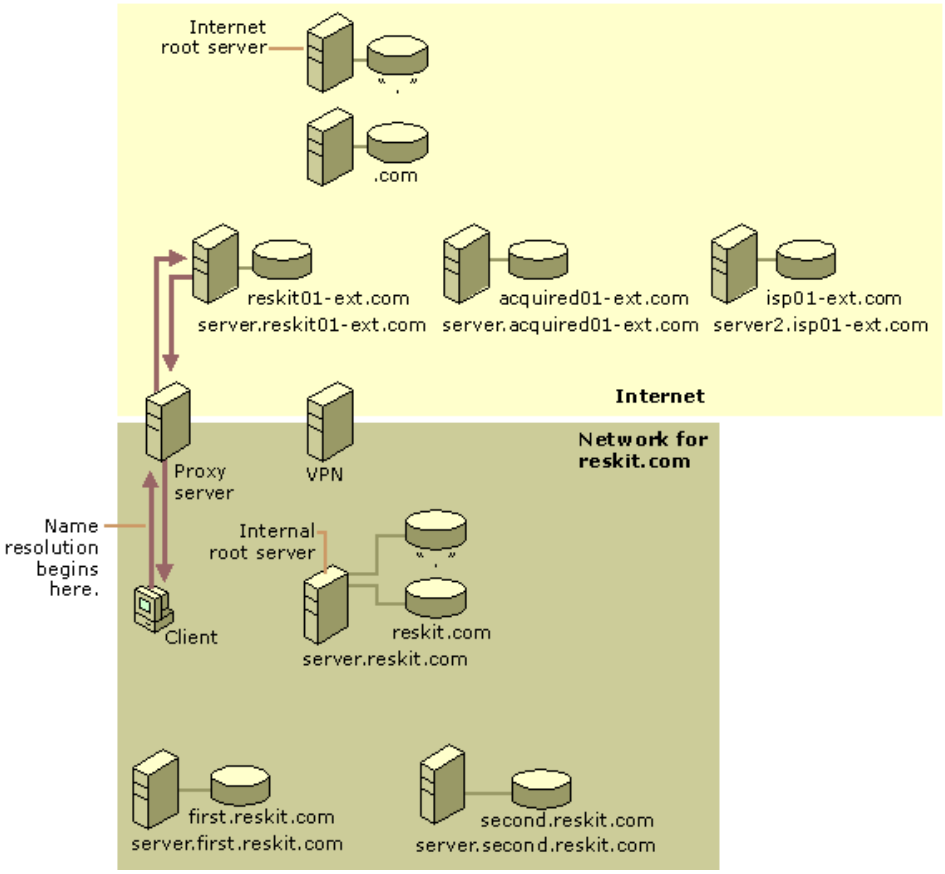
Figure 6.31 Query in the Domain Acquired01-int.com for a Name on the Internet

The query proceeds as follows:

1. The computer queries its local DNS server, server.first.acquired01-int.com.
2. If the server cache does not contain the requested data, the local DNS server forwards the query to the DNS server that is authoritative for the zone acquired01-int.com, server.acquired01-int.com.
3. The server server.acquired01-int.com forwards the query to the external server, server.acquired01-ext.com, through the firewall.
4. The server server.acquired01-ext.com sends a query to the Internet root server.
5. The Internet root server returns a referral to a server that is authoritative for the Internet zone com.
6. The server server.acquired01-ext.com queries the server that is authoritative for the zone com.
7. The server that is authoritative for the zone com returns a referral to the server that is authoritative for the zone isp01-ext.com.
8. The server server.acquired01-ext.com queries the server that is authoritative for the zone isp01-ext.com.
9. The server that is authoritative for the zone isp01-ext.com returns the IP address that corresponds to the name host.isp01-ext.com.
10. The server server.acquired01-ext.com returns the IP address to server.acquired01-int.com through the firewall.
11. Server.acquired01-int.com returns the IP address to the local DNS server, server.first.acquired01-int.com.
12. Server.first.acquired01-int.com returns the IP address to the client. The client can then contact the host through the firewall and download the desired Web page.

Query for a Name in the External Namespace of an Organization

Suppose that a computer in reskit.com needs to access a Web page in the external zone www.reskit01-ext.com. Figure 6.32 shows how the query proceeds.



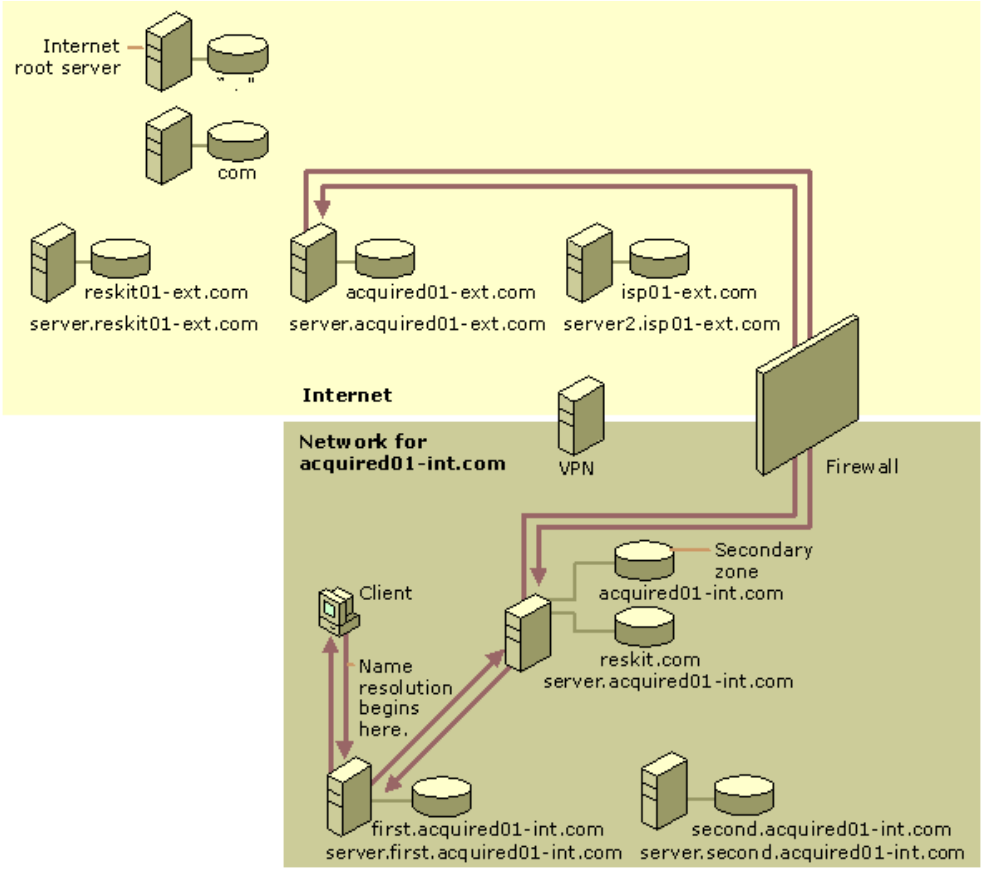
If your browser does not support inline frames, click here to view on a separate page.

Figure 6.32 Query for a Name in the External Zone Reskit01-ext.com

The query proceeds as follows:

1. Because the computer is a proxy client, it consults its exclusion list or its PAC file. After finding that the name is not in the exclusion list, it sends a request to the proxy server.
2. The proxy server submits the query to the DNS server that the proxy server is configured to use, server.reskit01-ext.com. In this example, server.reskit01-ext.com also happens to be authoritative for www.reskit.com.
3. The server server.reskit01-ext.com resolves the query and returns the response to the proxy server.
4. The proxy server uses the resulting IP address to contact server.reskit.com and provides the necessary information to the client.

Now suppose that a computer in the zone acquired01-int.com needs to open a Web page in the external zone www.acquired01-ext.com. Figure 6.33 shows how the query proceeds.



If your browser does not support inline frames, click here to view on a separate page.

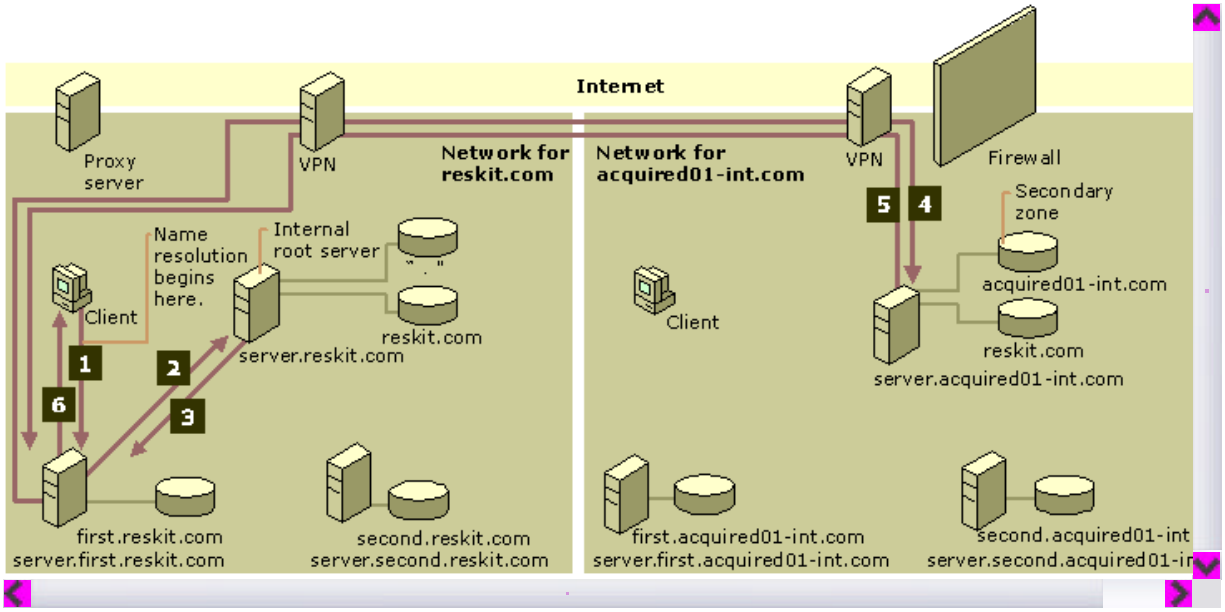
Figure 6.33 Query for a Name in the External Zone Acquired01-ext.com

The query proceeds as follows:

1. The computer submits the query to its local DNS server, server.first.acquired01-int.com.
2. If the cache does not contain the necessary data, server.first.acquired01-int.com forwards the query to the DNS server that is authoritative for the zone acquired01-int.com.
3. The server that is authoritative for the zone acquired01-int.com forwards the request through the firewall to server.acquired01-ext.com.
4. Server.acquired01-ext.com resolves the name and returns the response through the firewall to server.acquired01-int.com.
5. Server.acquired01-int.com returns the response to server.first.acquired01-int.com.
6. Server.first.acquired01-int.com returns the response to the client, and the client then uses the IP address to connect through the firewall to the Web server, which is located on the Internet.

Query for a Name in the Namespace of the Merged Organization

Suppose that a computer in reskit.com needs to contact the computer host.acquired01-int.com. Figure 6.34 shows how the query proceeds.



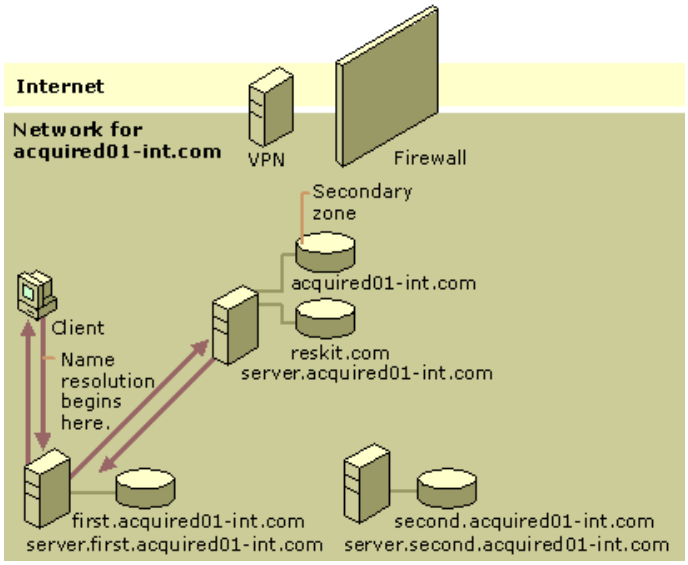
If your browser does not support inline frames, click here to view on a separate page.

Figure 6.34 Query for a Name in the Acquired01-int.com Namespace

The query proceeds as follows:

1. Because the computer is a proxy client, it consults its exclusion list or its PAC file and submits a query for the name host.acquired01-int.com to the local DNS server, server.first.reskit.com.
2. If the cache does not contain the necessary data, the server queries the internal root server.
3. The root server finds a delegation to the zone acquired01-int.com and returns the IP address of the server that is authoritative for acquired01-int.com to the local DNS server.
4. The local DNS server submits the query to the server that is authoritative for acquired01-int.com.
5. Because that server is authoritative for host.acquired01-int.com, the server resolves the query and returns the answer to the local DNS server.
6. Server.first.reskit.com returns the response to the client.

Now suppose that a computer in acquired01-int.com needs to contact the computer host.reskit.com. Figure 6.35 shows how the query proceeds.



If your browser does not support inline frames, click here to view on a separate page.

Figure 6.35 Query for a Name in the Reskit.com Namespace

The query proceeds as follows:

1. The computer submits a query to its local DNS server, server.first.acquired01-int.com.
2. If the cache does not contain the necessary data, the server forwards the query to the DNS server that is authoritative for the zone acquired01-int.com.
3. Because the DNS server that is authoritative for the zone acquired01-int.com contains a secondary copy of the zone reskit.com, it resolves the query and returns the response to server.first.acquired01-int.com.
4. Server.first.acquired01-int.com returns the response to the client.

Troubleshooting

The following sections describe useful troubleshooting tools, provides best practices to help you avoid common errors, lists procedures to help you verify that you have correctly configured your name servers, and explains how to diagnose and solve common DNS problems.

Troubleshooting Tools

Windows 2000 provides many tools that can help you diagnose and solve problems with DNS. This section discusses the following tools:

Nslookup You can use Nslookup to perform DNS queries and to examine the contents of zone files on local and remote servers.

Ipconfig You can use Ipconfig to view DNS client settings, display and flush the resolver cache, and force a dynamic update client to register its DNS records.

Event Viewer You can use Event Viewer to view DNS client and server error messages.

DNS Log You can configure the DNS server to monitor certain events and log them in the DNS log for your examination.

Network Redirector Command You can stop DNS client caching and flush the cache by using the network redirector commands **net start** and **net stop**.

Monitoring in the DNS Console You can perform test queries by using options on the **Monitoring** tab in the DNS console.

You can examine the packets that the DNS servers on your network send and receive by using Network Monitor. For more information about Network Monitor, see "Monitoring Network Performance" in the *Microsoft® Windows® 2000 Server Resource Kit Server Operations Guide*.

You can also use the Netdiag tool to quickly identify problems with your DNS configuration. For more information about Netdiag, see "TCP/IP Troubleshooting" in this book.

Nslookup

Nslookup is a standard command-line tool provided in most DNS server implementations, including Windows 2000. Nslookup offers the ability to perform query testing of DNS servers and obtain detailed responses at the command prompt. This information can be useful for diagnosing and solving name resolution problems, for verifying that resource records are added or updated correctly in a zone, and for debugging other server-related problems. This section describes how to perform troubleshooting tasks and lists and explains Nslookup error messages.

For information about the exact syntax of Nslookup, see Windows 2000 Server Help, or in Nslookup, type **help** at the command prompt.

Performing Simple Tasks with Nslookup

This section describes how to perform the following simple troubleshooting tasks:

- Use Nslookup in non-interactive mode to look up a single piece of data
- Enter interactive mode and use the debug feature
- Perform the following tasks from within interactive mode:
  - Set options for your query
  - Look up a name
  - Look up records in a zone
  - Perform zone transfers
  - Exit Nslookup

Note When you are entering queries, it is generally a good idea to enter FQDNs, so you can control what name is submitted to the server. However, if you want to know which suffixes are added to unqualified names before they are submitted to the server, you can enter Nslookup in debug mode and then enter an unqualified name.

To use Nslookup in non-interactive mode

- Type the following and then press ENTER:

  nslookup <*name*> <*server*>

  where *name* is the owner of the record you are looking for, and *server* is the server you want to query.

With interactive mode, you can look up more than one piece of data. Starting Nslookup with the command-line parameter **-d2** puts Nslookup in interactive mode with verbose debugging enabled. Verbose debugging enables you to examine the query and response packets between the resolver and the server.

To start Nslookup in interactive mode

- Type the following and then press ENTER:

  nslookup [-d2]

To exit interactive mode

- At the Nslookup prompt, type:

  exit

In interactive mode, you can use the **set** command to configure how the resolver will carry out queries. Table 6.14 shows a few of the options available with **set**:

Table 6.14 Command-Line Options Available With Set

Option	Purpose
set all	Shows all the options available with the **set** option.
set d2	Puts Nslookup in debug mode, so you can examine the query and response packets between the resolver and the sever.
set domain=<*domain name*>	Tells the resolver what domain name to append for unqualified queries.
set timeout=<*time-out*>	Tells the resolver what time-out to use. This option is useful for slow links where queries frequently time-out and the wait time must be lengthened.
set type=<*record type*> – Or – **set querytype=**<*record type*> – Or – **set q=**<*record type*>	Tells the resolver what type of resource records to search for (for example, A, PTR, or SRV). If you want the resolver to query for all types of resource records, type **set type=all**.

You can look up a single name.

To look up names from interactive mode

- Type the following:

  <*name*> [*server*]

  where *name* is the owner name for the record you are looking for, and *server* is the server that you want to query.

  You can use the wildcard character (*) in your query. For example, if you want to look for all resource records that have "K" as the first letter, you can type the following:

  K*

You can view the contents of a domain.

To view the contents of a domain

- Type the following:

  set type=<*record type*>

  ls -t <*domain name*>

  where *record type* is the type of record (use **any** to view all resource records) and *domain name* is the name of the domain you want to view.

By adding the **-d** switch, you can simulate and test a zone transfer. This can help you determine whether or not the server you are querying allows zone transfers to your computer.

To simulate a zone transfer

- Type the following:

  ls -d <*domain name*>

Nslookup provides help from the Nslookup prompt.

To get help from interactive mode

- At the Nslookup command prompt, type **help** or **?**.

Nslookup Errors

A successful Nslookup response looks like this:

Server: <*Name of DNS server*>

Address: <*IP address of DNS server*>

<*Response data*>

Nslookup might also return one of several errors. The following message means that the resolver did not locate a PTR resource record (containing the host name) for the server IP address. Nslookup can still query the DNS server, and the DNS server can still answer queries. For more information about using Nslookup to verify your DNS configuration, see "Verifying Your Basic DNS Configuration" later in this chapter.

DNS request timed out.

Timeout was <*x*> seconds.

*** Can't find server name for address <*IP Address*>: Timed out

*** Default servers are not available

Default Server: Unknown

Address: <*IP address of DNS server*>

The following message means that a request timed out. This might happen, for example, if the DNS service was not running on the DNS server that is authoritative for the name.

*** Request to <*Server*> timed-out

The following message means that the server is not receiving requests on UDP port 53. For more information about troubleshooting server problems, see "Checking the DNS Server for Problems" later in this chapter.

*** <*Server*> can't find <*Name or IP address queried for*>: No response from server

The following message means that this DNS server was not able to find the name or IP address in the authoritative domain. The authoritative domain might be on that DNS server or on another DNS server that this DNS server is able to reach.

*** <*Server*> can't find <*Name or IP address queried for*>: Non-existent domain

The following message generally means that the DNS server is running, but is not working properly. For example, it might include a corrupted packet, or the zone in which you are querying for a record might be paused. However, this message can also be returned if the client queries for a host in a domain for which the DNS server is not authoritative and the DNS server cannot contact its root servers, or is not connected to the Internet, or has no root hints.

*** <*Server*> can't find <*Name or IP address queried for*>: Server failed.

Using IPConfig

You can use the command-line tool Ipconfig to view your DNS client settings, to view and reset cached information used locally for resolving DNS name queries, and to register the resource records for a dynamic update client.

If you use Ipconfig with no parameters, it displays DNS information for each adapter, including the domain name and DNS servers used for that adapter.

Table 6.15 shows some command-line options available with Ipconfig.

Table 6.15 Ipconfig Command-Line Examples

Command	Action
ipconfig /all	Displays additional information about DNS, including the FQDN and the DNS suffix search list.
ipconfig /flushdns	Flushes and resets the DNS resolver cache. For more information about this option, see "Viewing and Displaying the Cache" earlier in this chapter.
ipconfig /displaydns	Displays the contents of the DNS resolver cache. For more information about this option, see "Viewing and Displaying the Cache" earlier in this chapter.
ipconfig /registerdns	Refreshes all DHCP leases and registers any related DNS names. This option is available only on Windows 2000–based computers that run the DHCP Client service. For more information about this option, see "Dynamic Update and Secure Dynamic Update" earlier in this chapter.
ipconfig /release [*adapter*]	Releases all DHCP leases.

ipconfig /renew [*adapter*]	Refreshes all DHCP leases and dynamically updates DNS names. This option is available only on systems that are running the DHCP Client service.

Event Viewer

The Event Viewer logs errors with the Windows 2000 operating system and services such as the DNS server. If you are having problems with DNS, you can check Event Viewer for DNS-related events.

To open the event viewer

- Click Start, point to **Programs**, point to **Administrative Tools**, and then click **Event Viewer.**

  To view messages about the DNS server, click **DNS Server**.

  – Or –

  To view messages about the DNS client, click **System Log**.

For more information about Event Viewer, see Windows 2000 Help.

DNS Log

You can configure the DNS server to create a log file that records the following types of events:

- Queries
- Notification messages from other servers
- Dynamic updates
- Content of the question section for DNS query message
- Content of the answer section for DNS query messages
- Number of queries this server sends
- Number of queries this server has received
- Number of DNS requests received over a UDP port
- Number of DNS requests received over a TCP port
- Number of full packets sent by the server
- Number of packets written through by the server and back to the zone

The DNS log appears in % *Systemroot*%\System32\dns\Dns.log. Because the log is in RTF format, you must use WordPad to view it.

You can change the directory and file name in which the DNS log appears by adding the following entry to the registry with the REG_SZ data type:

HKEY_LOCAL_MACHINE \SYSTEM \CurrentControlSet \Services \DNS \Parameters**LogFilePath**

Set the value of **LogFilePath** equal to the file path and file name where you want to locate the DNS log.

By default, the maximum file size of Dns.log is 4 MB. If you want to change the size, add the following entry to the registry with the REG_DWORD data type:

HKEY_LOCAL_MACHINE \SYSTEM \CurrentControlSet \Services \DNS \Parameters**LogFileMaxSize**

Set the value of **LogFileMaxSize** equal to the desired file size in bytes. The minimum size is 64 Kb.

Once the log file reaches the maximum size, Windows 2000 writes over the beginning of the file. If you make the value higher, data persists for a longer time, but the log file consumes more disk space. If you make the value smaller, the log file uses less disk space, but the data persists for a shorter time.

Caution Do not leave DNS logging during normal operation because it consumes both processing and hard disk resources. Enable it only when diagnosing and solving DNS problems.

To configure the server to log DNS events

1. In the DNS console, click the box next to the server, right-click the server, and then click Properties.
2. Click the **Logging** tab, and then select the options you want to log.

Stopping and Flushing the Cache

In addition to flushing the cache by using Ipconfig, you can stop and flush the cache by stopping and starting the client.

To stop the client

- At the command prompt, type the following:

  net stop "**dns client**"

To start the client

- At the command prompt, type the following:

  net start "**dns client**"

Monitoring in the DNS Console

You can use the DNS console to perform a test query to determine whether or not your server is working properly.

To perform test queries from within the DNS console

1. In the DNS console, double-click the server name to expand the server information.
2. Right-click the server, and then click **Properties**.
3. Click the **Monitoring** tab.
4. Select the tests you want to perform, and then click **Test Now**.

If the simple query fails, check whether the local server contains the zone 1.0.0.127.in-addr.arpa. If the recursive query fails, check whether your root hints are correct and whether your root servers are running. For more information about simple queries and recursive queries, see "Introduction to DNS" in this book.

For more information about troubleshooting recursion problems, see "Checking for Recursion Problems" later in this chapter.

Best Practices for Configuring and Administering DNS

Observe the following suggestions to prevent common configuration errors:

- Enter the correct e-mail address of the responsible person for each zone you add to or manage on a DNS server.

  This field is used by applications to notify DNS administrators for a variety of reasons. For example, this field can be used to report query errors, incorrect data returned in a query, and security problems. Although most Internet e-mail addresses contain the at sign (@) when used in e-mail applications, you must replace this symbol with a period (.) when typing an e-mail address for this field. For example, instead of administrator@reskit.com, you would use administrator.reskit.com.

- When designing your DNS network, use standard guidelines and wherever possible, follow preferred practices for managing your DNS infrastructure.

- Make sure that you have at least two servers hosting each zone. They can host either primary and secondary copies of the zone, or two directory-integrated copies of each zone.

- If you are using Active Directory, use directory-integrated storage for your zones.

  In an integrated zone, domain controllers for each of your Active Directory domains correspond in a direct one-to-one mapping to DNS servers. When you troubleshoot DNS and Active Directory replication problems, the same server computers are used in both topologies, which simplifies planning, deployment, and troubleshooting.

  Using directory-integrated storage also simplifies dynamic updates for DNS clients that are running Windows 2000. When you configure a list of preferred and alternate DNS servers for each client, you can specify servers corresponding to domain controllers located near each client. If a client fails to update with its preferred server because the server is unavailable, the client can try an alternate server. When the preferred server becomes available, it loads the updated, directory-integrated zone that includes the updates that the client made.

- If you are not using Active Directory integration, correctly configure your clients and understand that a standard primary zone becomes a single point of failure for dynamic updates and for zone replication.

  Standard primary zones are required to create and manage zones in your DNS namespace if you are not using Active Directory. In this case, a single-master update model applies, with one DNS server designated as the primary server for a zone. Only the primary server, as determined in the SOA record properties for the zone, can process an update to the zone.

  For this reason, make sure that this DNS server is reliable and available. Otherwise, clients cannot update their A or PTR resource records.

- Consider using secondary or caching-only servers for your zones to offload DNS query traffic.

  Secondary servers can be used as backups for DNS clients, but they can also be used as the preferred DNS servers for legacy DNS clients. For mixed-mode environments, this enables you to balance the load of DNS query traffic on your network and, thus, reserve your DNS-enabled primary servers for Windows 2000–based clients that need primary servers to perform dynamic registration and updates of their A and PTR resource records.

The IETF has published several Requests for Comment (RFCs) that cover best practices for DNS, as recommended by DNS architects and planners for the Internet. You might find the following RFCs useful, especially if you are planning a large DNS design:

- RFC 1912, "Common DNS Operational and Configuration Errors"
- RFC 2182, "Selection and Operation of Secondary DNS Servers"
- RFC 2219, "Use of DNS Aliases for Network Services"

Verifying Your Basic DNS Configuration

If you use a third-party DNS server to support Active Directory, you must perform configuration tasks manually, and doing so, you might cause common configuration errors that prevent DNS and Active Directory from working properly. The following sections describe tests that you can perform to verify that your DNS server is working properly, that the forward and reverse lookup zones are properly configured, and that DNS can support Active Directory.

If you use either the Configure DNS Server wizard or the Active Directory Installation wizard to install your Windows 2000 DNS server, most configuration tasks are performed automatically and you can avoid many common configuration errors, but you might still want to perform the tests in this section.

Before checking anything else, check the event log for errors. For more information about Event Viewer, see "Troubleshooting Tools" earlier in this chapter.

Verifying That Your DNS Server Can Answer Queries

Use the following process to verify that your DNS server is started and can answer queries.

- Make sure that your server has basic network connectivity. For more information about verifying basic network connectivity, see "Checking the DNS Server for Problems" later in this chapter.
- Make sure that the server can answer both simple and recursive queries from the Monitoring tab in the DNS console. For more information about the Monitoring tab, see "Troubleshooting Tools" earlier in this chapter.
- From a client, use Nslookup to look up a domain name and the name of a host in the domain. For more information about using Nslookup, see "Troubleshooting Tools" earlier in this chapter.
- On the server, run **netdiag** to make sure the server is working properly and that the resource records Netlogon needs are registered on a DNS server. For more information about Netdiag, see "Troubleshooting Tools" earlier in this chapter.
- Make sure that the server can reach a root server by typing the following:

  nslookup

  server <*IP address of server***>**

  set querytype=NS

  .

- Make sure that there is an A and PTR resource record configured for the server. For information about PTR resource records, see "Testing for Reverse Lookup Zones and PTR Records" later in this chapter.

Verifying That the Forward Lookup Zone Is Properly Configured

After you create a forward lookup zone, you can use Nslookup to make sure it is properly configured and to test its integrity to host Active Directory. To start Nslookup, type the following

Nslookup
server <*IP address of server on which you created zone*>
set querytype=any

Nslookup starts. If the resolver cannot locate a PTR resource record for the server, you see an error message, but you are still able to perform the tests in this section.

To verify the zone is responding correctly, simulate a zone transfer by typing the following:

ls -d <*domain name*>

If the server is configured to restrict zone transfers, you might see an error message in Event Viewer. (For more information about Event Viewer, see "Troubleshooting Tools" earlier in this chapter.) Otherwise, you see a list of all the records in the domain.

Next, query for the SOA record by typing the following and pressing ENTER:

<*domain name*>

If your server is configured correctly, you see an SOA record. The SOA record includes a "primary name server" field. To verify that the primary name server has registered an NS record, type the following:

set type=ns
<*domain name*>

If your server is configured correctly, you see an NS record for the name server.

Make sure that the authoritative name server listed in the NS record can be contacted to request queries by typing the following:

server <*server name or IP address*>

Next, query the server for any name for which it is authoritative.

If these tests are successful, the NS record points to the correct hostname, and the hostname has the correct IP address associated with it.

Testing for Reverse Lookup Zones and PTR Resource Records

You do not need reverse lookup zones and PTR resource records for Active Directory to function. However, you need them if you want clients to be able to resolve FQDNs from IP addresses. Also, PTR resource records are commonly used by some applications for security purposes, to verify the identity of the client.

You do not need to have the reverse lookup zones and PTR resource records on your own servers; instead, another DNS server can contain these zones.

After you have configured your reverse lookup zones and PTR resource records, manually examine them in the DNS console. A reverse lookup zone must exist for each subnet, and the parent reverse lookup zone must have a delegation to your reverse lookup zone. For example, if you have a private root and the subnets 172.32.16.x and 172.32.17.x, the private root can host all reverse lookup zones, or it can contain the reverse lookup zone 172.32.x and delegate the reverse lookup zones 172.32.16.x and 172.32.17.x to other servers. Also, PTR resource records must exist for all the computers in your network. For more information about adding a reverse lookup zone, see "Adding a Reverse Lookup Zone" earlier in this chapter.

You can also use Nslookup to verify that the reverse lookup zones and PTR resource records are configured correctly.

To make sure your reverse lookup zones and PTR resource records are configured correctly

1. Start Nslookup by typing **Nslookup** at the command prompt and then pressing ENTER.
2. Switch to the server you want to query by typing the following:

   server <*Server IP Address*>
3. Enter the IP address of the computer whose PTR resource record you want to verify, and then press ENTER.

   If the reverse lookup zone and PTR resource record are configured correctly, Nslookup returns the name of the computer.
4. To quit Nslookup, type **exit** and then press ENTER.

Verifying Your DNS Configuration After Installing Active Directory

When you use third-party DNS servers to support Active Directory, you can verify the registration of domain controller locator resource records. If the server does not support dynamic update, you need to add these records manually.

The Netlogon service creates a log file that contains all the locator resource records and places the log file in the following location:

%SystemRoot%\System32\Config\Netlogon.dns

You can check this file to find out which locator resource records are created for the domain controller.

The locator resource records are stored in a text file, compliant with RFC specifications. If your server is configured correctly, you see the LDAP SRV record for the domain controller:

```
_ldap._tcp.<Active Directory domain name> IN SRV <priority> <weight> 389 <domain controller name>
```

For example:

```
_ldap._tcp.reskit.com. IN SRV 0 0 389 dc1.reskit.com
```

Next, use the Nslookup command-line tool to verify that the domain controller registered the SRV resource records that were listed in Netlogon.dns.

Note During the following test, if you have not configured a reverse lookup zone and PTR resource record for the DNS server you are querying, you might see several time-outs. This is not a problem.

To verify that SRV resource records are registered for the domain controller

1. At the command prompt, type **nslookup** and then press ENTER.
2. To set the DNS query type to filter for SRV records only, type **set type=SRV** and then press ENTER.

3. To send a query for the registered SRV record for a domain controller in your Active Directory domain, type **_ldap._tcp.**<*Active Directory domain name*> and then press ENTER.
4. You should see the SRV records listed in Netlogon.dns. If you do not, SRV resource records might not be registered for the domain controller.

The following example shows a full Nslookup session, used to verify SRV resource records that are registered for locating two domain controllers on a network. In this example, the two domain controllers (DC1 and DC2) are registered for the domain noam.reskit.com.

```
C:\>nslookup
Default Server: dc1.noam.reskit.com
Address: 10.0.0.14
> set type=SRV
> _ldap._tcp.noam.reskit.com
Server: dc1.noam.reskit.com
Address: 10.0.0.14
_ldap._tcp.noam.reskit.com SRV service location:
priority = 0
weight = 0
port = 389
svr hostname = dc1.noam.reskit.com
_ldap._tcp.noam.reskit.com SRV service location:
priority = 0
weight = 0
port = 389
svr hostname = dc2.noam.reskit.com
dc1.noam.reskit.com internet address = 10.0.0.14
dc2.noam.reskit.com internet address = 10.0.0.15
```

Diagnosing Name Resolution Problems

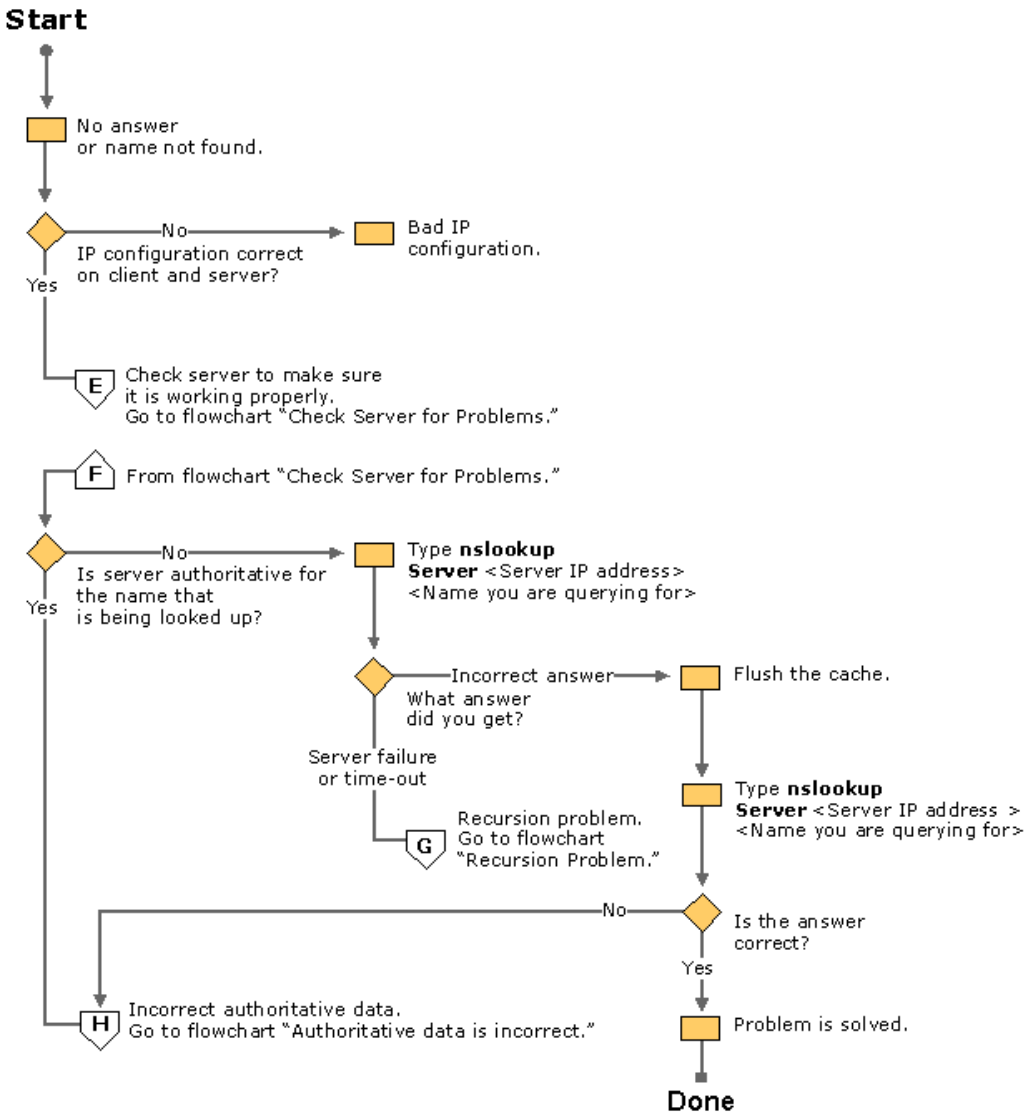
Most failed name resolution attempts fail in one of two general ways:

- A user receives a negative response when attempting to resolve a name, such as an error message indicating "name not found."
- A user receives a positive response when attempting to resolve a name, but the information returned is incorrect.

Important Whenever you are trying to troubleshoot problems with name resolution, always submit an FQDN. In that way, you can make sure that your problem is not caused by an incorrect domain suffix appended to the queried name.

The following flowcharts and associated text in Figures 6.36–6.41 explain how to diagnose each of these problems. For another good source of information for diagnosing common problems, see RFC 1912, "Common DNS Operational and Configuration Errors."

Note The flowcharts in Figures 6.36–6.41 direct you to other flowcharts in other figures. To locate the correct flow chart, see the figure captions.



If your browser does not support inline frames, click here to view on a separate page.

Figure 6.36 No Answer or Name Not Found

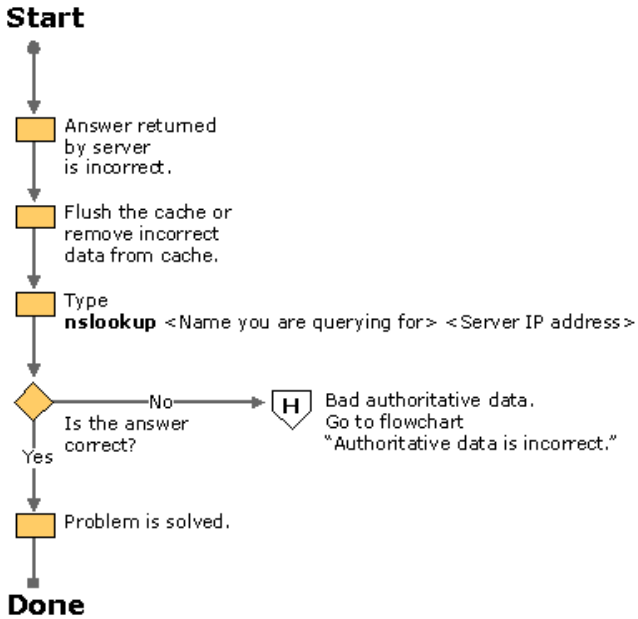
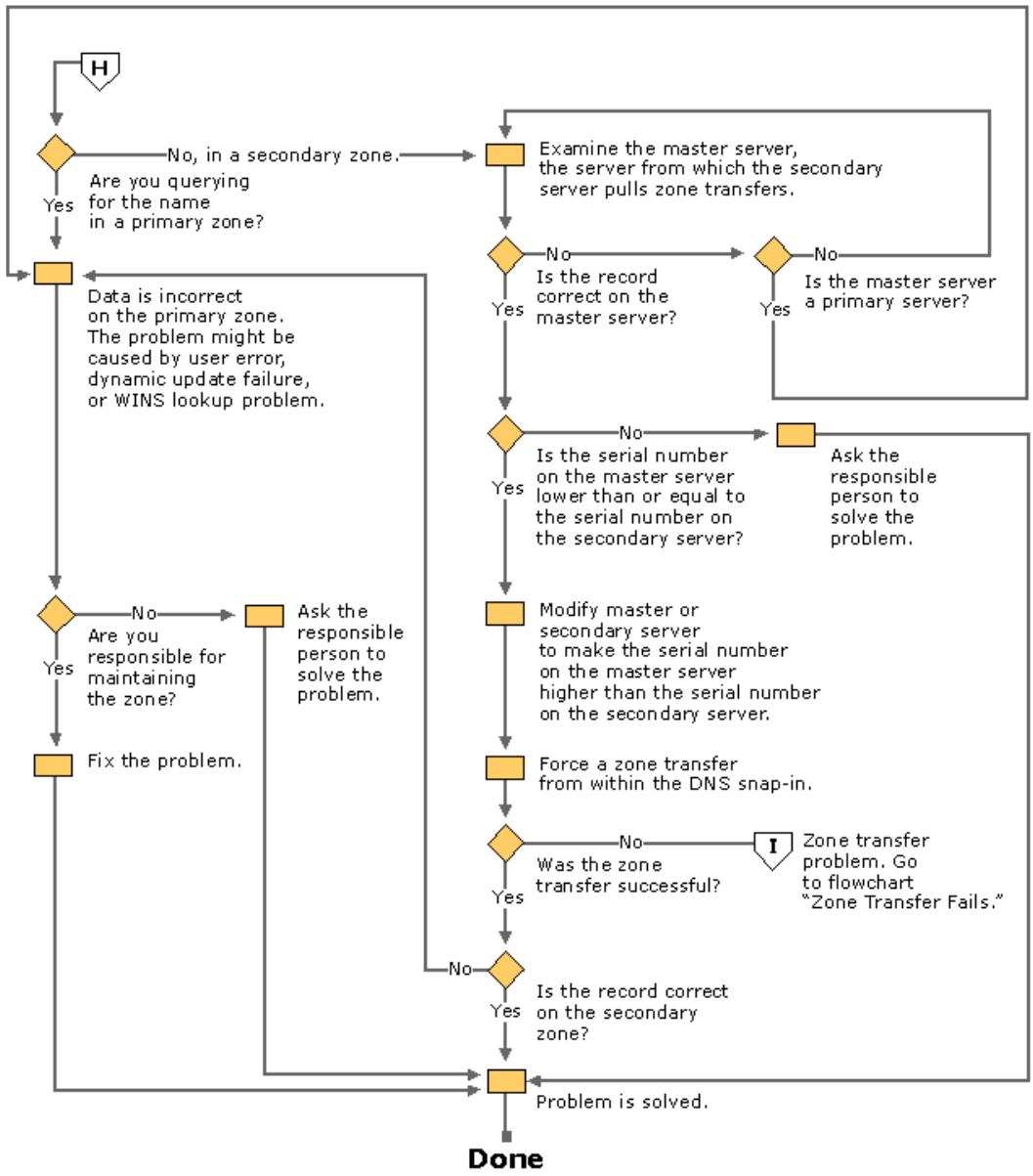
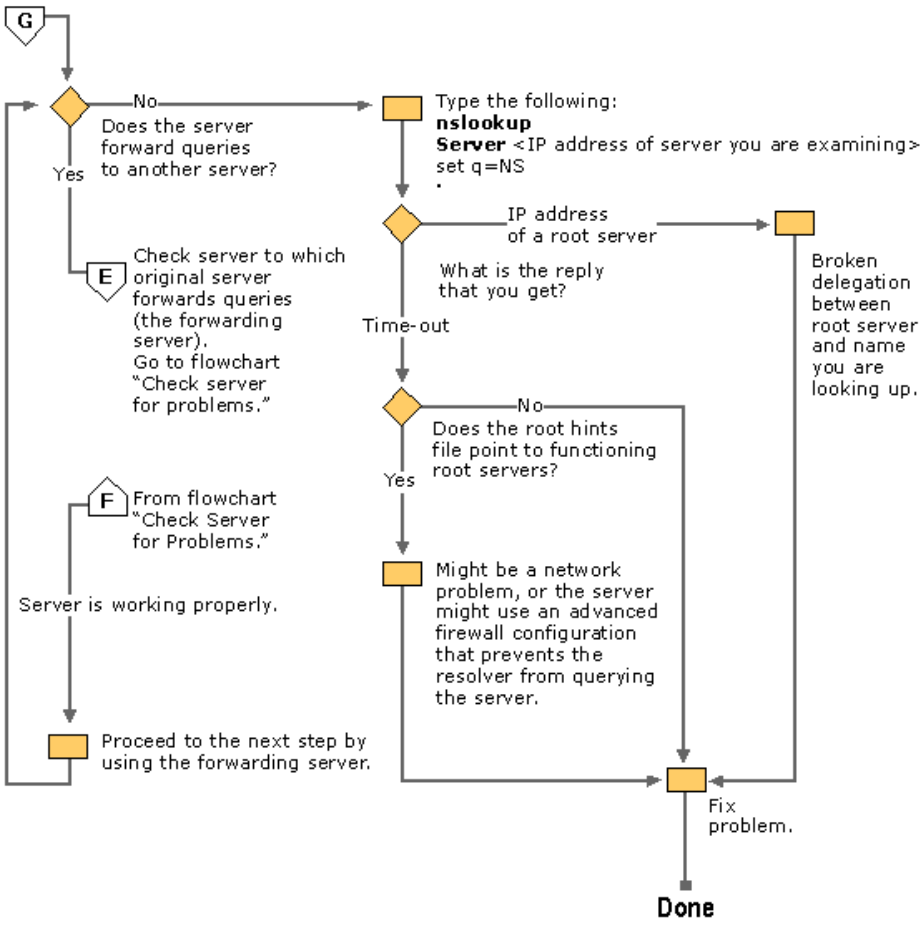


Figure 6.37 Answer Is Incorrect



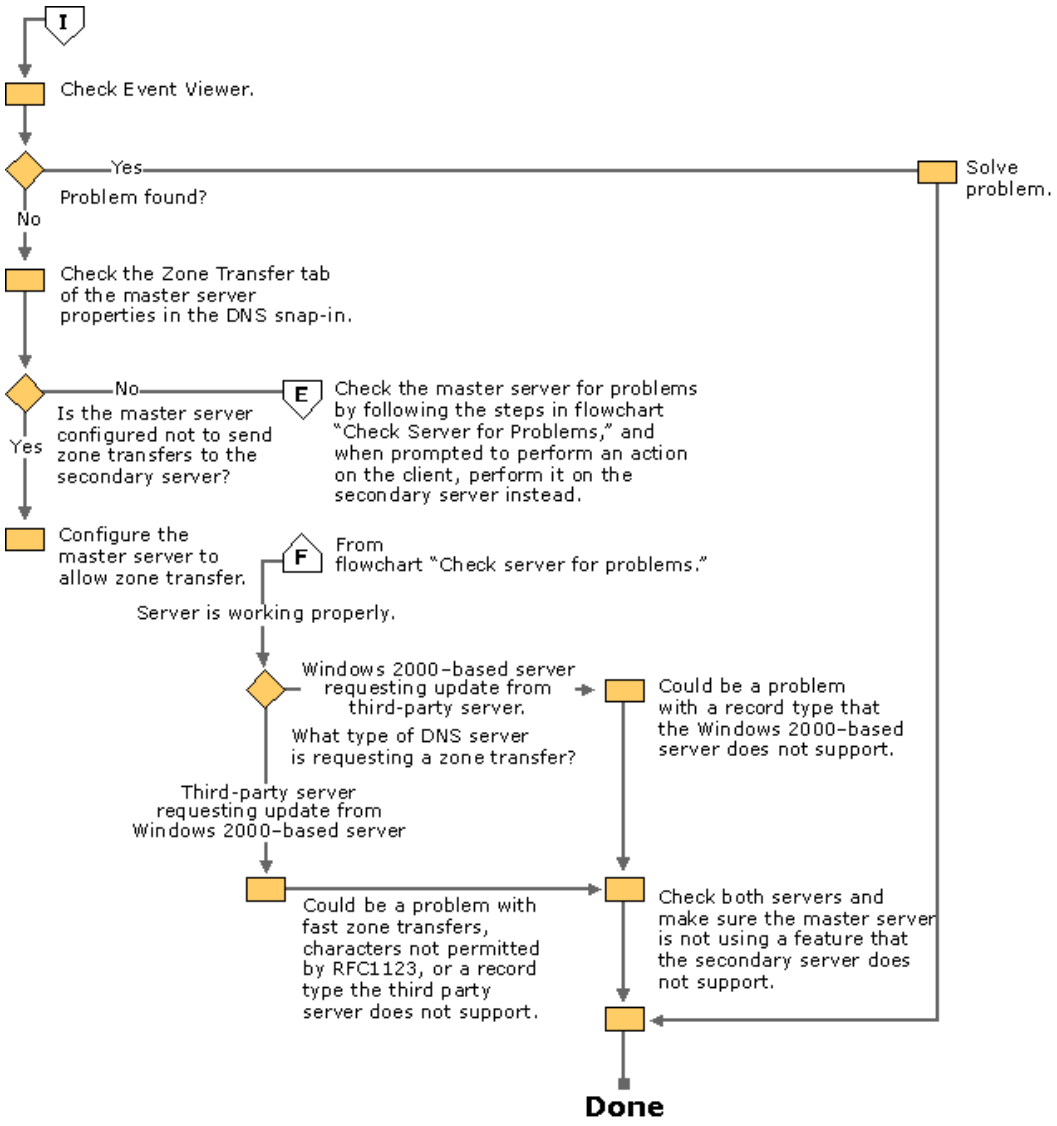
If your browser does not support inline frames, click here to view on a separate page.

Figure 6.38 Authoritative Data Is Incorrect



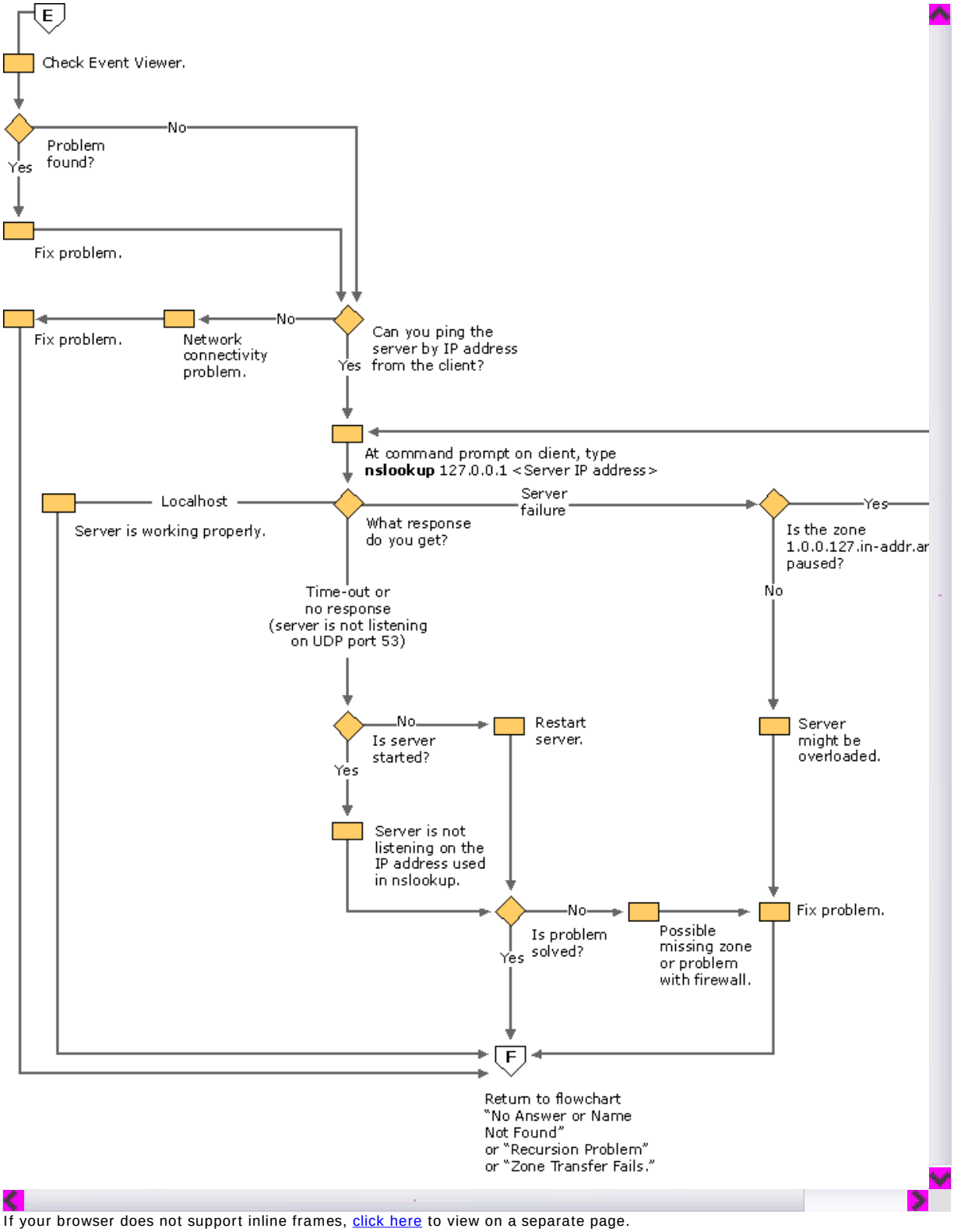
If your browser does not support inline frames, click here to view on a separate page.

Figure 6.39 Recursion Problem



If your browser does not support inline frames, click here to view on a separate page.

Figure 6.40 Zone Transfer Fails



If your browser does not support inline frames, click here to view on a separate page.

Figure 6.41 Check Server for Problems

Cannot Find Name or IP Address

If a query fails because you get the response **Non-existent domain** from Nslookup or the response **Unknown host** from Ping, the DNS server did not find the name or IP address that you are looking up. Use the following process, shown in Figure 6.36, to help troubleshoot the problem:

1. Check that the client and server computers have a valid IP configuration.

   To check IP configuration, type **ipconfig /all** at the command prompt. In the command-line output, verify the IP address, subnet mask, and default gateway.

2. Check that the server is working properly. For more information about verifying that the server is working properly, see "Checking the DNS Server for Problems" later in this chapter.

3. Check whether the DNS server is authoritative for the name that is being looked up.

   If the DNS server is authoritative for the name that is being looked up, you probably have a problem with authoritative data. For more information about checking for problems with authoritative data, see "Checking for Problems with Authoritative Data" later in this chapter.

– Or –

If the DNS server is not authoritative for the name that is being looked up, proceed to the next step.

4. Query for the name by using Nslookup. At the command prompt, type the following:

   Nslookup <*query address*> <*IP address of server*>

   where *IP address of server* is the IP address of the server that you queried originally, and *query address* is the name or IP address you are attempting to resolve. If you get the message "Server failed" or "Request to *server* timed out," you probably have a problem involving a broken delegation. For more information about problems with broken delegations, see "Checking for Recursion Problems" later in this chapter.

   – Or –

   If you get an incorrect answer or the message "Non-existent domain," proceed to the next step.

5. Flush the resolver cache. At the command prompt, type the following:

   Nslookup <*query address*> <*IP address of server*>

   where *IP address of server* is the IP address of the server that you queried originally, and *query address* is the name or IP address you are attempting to resolve. If the answer is correct, the problem was a stale cache entry, and your problem is solved.

   – Or –

   If the answer is still not correct, you probably have a problem with authoritative data. For more information about problems with authoritative data, see "Checking for Problems with Authoritative Data" later in this chapter.

Incorrect Answer

If you query a DNS server and it responds with incorrect information, use the following process, shown in Figure 6.37, to solve the problem.

1. Flush the resolver cache.
2. At the command prompt, type the following:

   Nslookup <*query address*> <*IP address of server*>

   where *IP address of server* is the IP address of the server that you queried originally, and *query address* is the name or IP address you are attempting to resolve. If the answer is correct, the problem was a stale cache entry, and your problem is solved.

   – Or –

   If the answer is still not correct, you probably have a problem with authoritative data. For more information about how to diagnose problems with authoritative data, see "Checking for Problems with Authoritative Data" later in this chapter.

Checking the DNS Server for Problems

Use the following process, shown in Figure 6.41, to check the DNS server for problems.

1. Check Event Viewer for error messages. For information about Event Viewer, see "Troubleshooting Tools" earlier in this chapter.
2. Check for basic connectivity between the client computer and the DNS server that you used for your original query by pinging the DNS server by its IP address.

   If the DNS server does not respond to a direct ping of its IP address, you probably have a network connectivity problem between the client and the DNS server.

3. At the command prompt on the client computer, type the following:

   nslookup 127.0.0.1 <*IP address of server*>

   If the resolver returns the name of the local host, the server does not have any problems.

   – Or –

   If the resolver returns the response "Server failure," proceed to step 4.

   – Or –

   If the resolver returns the response "Request to *server* timed out" or "No response from server," proceed to step 5.

4. If the resolver returns the response "Server failure," the zone 1.0.0.127.in-addr.arpa is probably paused, or the server is possibly overloaded. You can find out whether it is paused by checking the **General** tab of the zone properties, from within the DNS console.
5. If the resolver returns the response "Request to *server* timed out" or "No response from server," the DNS server probably is not running. Try to restart the server by typing the following at the command prompt on the server:

   net start DNS

   – Or –

   If it is running, the server might not be listening on the IP address that you used in your Nslookup query. From the **Interfaces** tab of the server properties page in the DNS console, administrators can restrict a DNS server to listen only on selected addresses. If the DNS server has been configured to limit service to a specific list of its configured IP addresses, it is possible that the IP address used to contact the DNS server is not in the list. You can try a different IP address in the list or add the IP address to the list. For more information about restricting a DNS server to listen only on selected addresses, see Windows 2000 Help.

   – Or –

   In rare cases, the DNS server might be configured to disable the use of its automatically created default zones. By default, the DNS service automatically creates the following standard reverse lookup zones based on RFC recommendations:

   - 0.in-addr.arpa
   - 127.in-addr.arpa
   - 255.in-addr.arpa

   The automatic creation of these zones can only be disabled through the registry, so it is unlikely that this has happened. However, if you think automatic creation has been disabled, you can use the DNS console to make sure that the zones exist.

   – Or –

   In rare cases, the DNS server might have an advanced security or firewall configuration. If the server is located on another network that is reachable only through an intermediate host (such as a packet filtering router or proxy server), the DNS server might use a non-standard port to listen for and receive client requests. By default, Nslookup sends queries to DNS servers on UDP port 53, so if the DNS server uses any other port, Nslookup queries fail. If you think this might be the problem, check whether an intermediate filter is intentionally used to block traffic on well-known DNS ports. If not, try to modify the packet filters or port rules on the

firewall to allow traffic on UDP/TCP port 53.

Diagnosing Problems with Incorrect Authoritative Data

If you have determined that the server contains incorrect authoritative (non-cached) data, use the following process to help troubleshoot the problem:

1. Determine whether the server that is returning the incorrect response is either a primary or secondary server for the zone.

   If the server is a primary server for the zone — either the standard primary server for the zone or a server that uses Active Directory integration to load the zone — the data is incorrect on the primary zone. Go to step 5.

   – Or –

   If the server is hosting a secondary copy of the zone, proceed to the next step.

2. Examine the zone on the master server (the server from which this server pulls zone transfers). You can determine which server is the master server by examining the properties of the secondary zone in the DNS console. Is the name correct on the master server?

   If the name is not correct on the master server, go to step 1. When prompted to examine a server, examine the server from which this server pulls zone transfers.

   – Or –

   If the name was correct on the master server, proceed to the next step.

3. Check whether the serial number on the master server is lower than or equal to the serial number on the secondary server. If not, proceed to the next step.

   – Or –

   If the serial number on the master server is lower than or equal to the serial number on the secondary server, modify either the master server or the secondary server so that the serial number on the master server is higher than the serial number on the secondary server. Then, proceed to the next step.

4. Force a zone transfer from within the DNS console. (For information about how to force a zone transfer, see Windows 2000 Server Help.) Next, examine the secondary server again to see whether the zone was transferred correctly. If not, you probably have a zone transfer problem; see "Zone Transfer Problems" later in this chapter.

   – Or –

   If the zone was transferred correctly, check whether the data is now correct. If not, the data is incorrect on the primary zone. Proceed to the next step.

5. If the data is incorrect on the primary zone, the problem might be caused by user error when entering data into the zone, a problem with Active Directory replication, a problem with dynamic update, or a WINS lookup problem. For information about problems with user error and Active Directory replication, see "Troubleshooting DNS Problems" later in this chapter. For information about problems with dynamic update, see "Troubleshooting Dynamic Update." For information about WINS lookup problems, see "Solving Common DNS Problems" later in this chapter.

   If you are responsible for maintaining the zone, you can solve the problem. Otherwise, ask the person who is responsible for maintaining the zone to solve the problem.

Diagnosing Problems with Recursion

For recursion to work successfully, all DNS servers that are used in the path of a recursive query must be able to respond and forward correct data. If they cannot, a recursive query can fail for any of the following reasons:

- The query times out before it can be completed.
- A server used during the query fails to respond.
- A server used during the query provides incorrect data.

If you have determined that you have a problem with recursion, use the following process, shown in Figure 6.39, to help troubleshoot the problem. Start with the server used in your original query:

1. Check whether this server forwards queries to another server by examining the **Forwarders** tab in the server properties in the DNS console. If the check box **Enable forwarders** is selected and one or more servers are listed, this server forwards queries.

   If this server does forward queries to another server, check for problems with the server to which this server forwards queries. To check for problems, follow the troubleshooting steps in "Checking the DNS Server for Problems." When that section instructs you to perform a task on the client, perform it on the server instead.

   If the server is healthy and can forward queries, repeat this step, examining the server to which this server forwards queries.

   – Or –

   If this server does not forward queries to another server, proceed to the next step.

2. Test whether this server can query a root server by typing the following:

   nslookup
   server <*IP address of the server you are examining*>
   set querytype=NS
   .

   If the resolver returns the IP address of a root server, you probably have a broken delegation between the root server and the name or IP address that you are attempting to resolve. Follow the procedure "To test for a broken delegation" to determine where you have a broken delegation.

   –Or –

   If the resolver returns the response "Request to *server* timed out," check whether the root hints points to functioning root servers by following the procedure "To view the current root hints." If the root hints does point to functioning root servers, you might have a network problem, or the server might use an advanced firewall configuration that prevents the resolver from querying the server, as described in "Checking the Server for Problems," earlier in this chapter. It is also possible that the recursive time-out default (15 seconds) is too short. For information about how to change this time-out, see the Windows 2000 Server Help. Search for "tuning advanced parameters."

Note Begin the tests in the following procedure by querying a valid root server. The test takes you through a process of querying all the DNS servers from the root down to the server that you are testing for a broken delegation.

To test for a broken delegation

1. At the command prompt on the server that you are testing, type the following:

nslookup

server <*server IP address*>
set norecursion
set querytype=<*resource record type*>

<*FQDN* >

where *resource record type* is the type of resource record that you were querying for in your original query, and *FQDN* is the FQDN for which you were querying (terminated by a period).

2. If the response includes a list of NS and A resource records for delegated servers, repeat step 1 for each server and use the IP address from the A resource records as the server IP address.

   – Or –

   If the response does not contain an NS resource record, you have a broken delegation.

   –Or –

   If the response contains NS resource records, but no A resource records, type **set recursion** and query individually for A resource records of servers listed in the NS records. If for each NS resource record in a zone, you do not find at least one valid IP address of an A resource record for each NS resource record, you have a broken delegation.

If you determine that you have a broken delegation, fix it by adding or updating an A resource record in the parent zone with a valid IP address for a correct DNS server for the delegated zone.

To view the current root hints

1. Start the DNS console.
2. Add or connect to the DNS server that failed a recursive query.
3. Right-click the server and select **Properties**.
4. Click **Root Hints**.
5. Check for basic connectivity to the root servers.
6. If root hints appear to be configured correctly, verify that the DNS server used in a failed name resolution can ping the root servers by IP address.

   If the root servers do not respond to pinging by IP address, the IP addresses for the root servers might have changed. However, reconfiguration of root servers, is uncommon.

Diagnosing Zone Transfer Problems

If you have determined that a secondary server cannot pull a zone transfer from a master server, use the following process, shown in Figure 6.40, to diagnose and solve your zone transfer problems.

1. Check Event Viewer for both the primary and secondary DNS server. For information about Event Viewer, see "Troubleshooting Tools" earlier in this chapter.
2. Check the master server to see whether it is refusing to send the transfer for security reasons. Check the **Zone Transfers** tab of the zone properties in the DNS console. If the server restricts zone transfers to a list of servers, such as those listed on the **Name Servers** tab of the zone properties, make sure that the secondary server is on that list. Make sure that the server is configured to send zone transfers.
3. Check the master server for problems by following the steps in "Checking the DNS Server for Problems" earlier in this chapter. When prompted to perform a task on the client, perform the task on the secondary server instead.
4. Check whether the secondary server is running another DNS server implementation, such as BIND. If so, the problem might have one of several causes:
   - The Windows 2000 master server might be configured to send fast zone transfers, but the third-party secondary server might not support fast zone transfers. If so, disable fast zone transfers on the master server by selecting the check box **Bind secondaries** on the **Advanced** tab of the properties for your server, from within the DNS console.
   - If a forward lookup zone on the Windows 2000 server contains a WINS lookup record or the reverse lookup zone contains a WINS-R record, the BIND server might not be able to transfer the zone. For information about diagnosing problems in which a BIND server cannot transfer a zone, see "Solving Common DNS Problems" later in this chapter.
   - If a forward lookup zone on the Windows 2000 server contains a record type (for example, an SRV record) the secondary server does not support, the secondary server might have problems pulling the zone.
5. Check whether the master server is running another DNS server implementation, such as BIND.

   If so, it is possible that the zone on the master server includes incompatible resource records that Windows 2000 does not recognize. For a complete list of all RFC-compliant resource record types that are supported by DNS servers that are running under Windows 2000 Server, see Windows 2000 Server Help.
6. If either the master or secondary server is running another DNS server implementation, check both servers to make sure that they support the same features. You can check the Windows 2000 server from the **Advanced** tab of the properties page for the server from within the DNS console. In addition to the **Bind secondaries** box, this page includes the **Name checking** drop down list, which enables you to select enforcement of strict RFC compliance for characters in DNS names.

Solving Other Common DNS Problems

This section lists several common DNS problems and explains how to solve them.

Event ID 7062 appears in the event log.

If you see event ID 7062 in the event log, the DNS server has sent a packet to itself. This is usually caused by a configuration error. Check the following:

- Make sure that there is no lame delegation for this server. A *lame delegation* occurs when one server delegates a zone to a server that is not authoritative for the zone.
- Check the forwarders list to make sure that it does not list itself as a forwarder.
- If this server includes secondary zones, make sure that it does not list itself as a master server for those zones.
- If this server includes primary zones, make sure that it does not list itself in the notify list.

Zone transfers to secondary servers that are running BIND are slow.

By default, the Windows 2000 DNS server always uses a fast method of zone transfer. This method uses compression and includes multiple resource records in each message, substantially increasing the speed of zone transfers. Most DNS servers support fast zone

transfer. However, BIND 4.9.4 and earlier does not support fast zone transfer. This is unlikely to be a problem, because when the Windows 2000 DNS Server service is installed, fast zone transfer is disabled by default. However, if you are using BIND 4.9.4 or earlier, and you have enabled fast zone transfer, you need to disable fast zone transfer.

To disable fast zone transfer

1. In the DNS console, right-click the DNS server, and then click **Properties**.
2. Click the **Advanced** tab.
3. In the **Server options** list, select the **Bind secondaries** check box, and then click **OK**.

You see the error message "Default servers are not available."

When you start Nslookup, you might see the following error message:

```
*** Can't find server name for address <address>: Non-existent domain

*** Default servers are not available

Default Server: Unknown

Address: 127.0.0.1
```

If you see this message, your DNS server is still able to answer queries and host Active Directory. The resolver cannot locate the PTR resource record for the name server that it is configured to use. The properties for your network connection must specify the IP address of at least one name server, and when you start Nslookup, the resolver uses that IP address to look up the name of the server. If the resolver cannot find the name of the server, it displays that error message. However, you can still use Nslookup to query the server.

To solve this problem, check the following:

- Make sure that a reverse lookup zone that is authoritative for the PTR resource record exists. For more information about adding a reverse lookup zone, see "Adding a Reverse Lookup Zone" earlier in this chapter.
- Make sure that the reverse lookup zone includes a PTR resource record for the name server.
- Make sure that the name server you are using for your lookup can query the server that contains the PTR resource record and the reverse lookup zone either iteratively or recursively.

User entered incorrect data in zone.

For information about how to add or update records by using the DNS console, see Windows 2000 Server Help. For more information about using resource records in zones, search for the keywords "managing" and "resource records" in Windows 2000 Server Help.

Active Directory-integrated zones contain inconsistent data.

For Active Directory–integrated zones, it is also possible that the affected records for the query have been updated in Active Directory but not replicated to all DNS servers that are loading the zone. By default, all DNS servers that load zones from Active Directory poll Active Directory at a set interval — typically, every 15 minutes — and update the zone for any incremental changes to the zone. In most cases, a DNS update takes no more than 20 minutes to replicate to all DNS servers that are used in an Active Directory domain environment that uses default replication settings and reliable high-speed links.

User cannot resolve name that exists on a correctly configured DNS server.

First, confirm that the name was not entered in error by the user. Confirm the exact set of characters entered by the user when the original DNS query was made. Also, if the name used in the initial query was unqualified and was not the FQDN, try the FQDN instead in the client application and repeat the query. Be sure to include the period at the end of the name to indicate the name entered is an exact FQDN.

If the FQDN query succeeds and returns correct data in the response, the most likely cause of the problem is a misconfigured domain suffix search list that is used in the client resolver settings.

Name resolution to Internet is slow, intermittent, or fails.

If queries destined for the Internet are slow or intermittent, or you cannot resolve names on the Internet, but local Intranet name resolution operates successfully, the cache file on your Windows 2000–based server might be corrupt, missing, or out of date. You can either replace the cache file with an original version of the cache file or manually enter the correct root hints into the cache file from the DNS console. If the DNS server is configured to load data on startup from Active Directory and the registry, you must use the DNS console to enter the root hints.

To enter root hints in the DNS console

1. In the DNS console, double-click the server to expand it.
2. Right-click the server, and then click **Properties**.
3. Click the **Root Hints** tab.
4. Enter your root hints, and then click **OK**.

To replace your cache file

1. Stop the DNS service by typing the following at the command prompt:

   net stop dns
2. Type the following:

   cd %Systemroot%\System32\DNS
3. Rename your cache file by typing the following:

   ren cache.dns cache.old
4. Copy the original version of the cache file, which might be found in one of two places, by typing either of the following:

   copy backup\cache.dns

   – Or –

   copy samples\cache.dns
5. Start the DNS service by typing the following:

   net start dns

If name resolution to the Internet still fails, repeat the procedure, copying the cache file from your Windows 2000 source media.

To copy the cache file from your Windows 2000 source media

- At the command prompt, type the following:

  expand <*drive*>**:\i386\cache.dn_ %***Systemroot***%\system32\dns\cache.dns**

  where *drive* is the drive that contains your Windows 2000 source media.

Resolver does not take advantage of round robin feature.

Windows 2000 includes subnet prioritization, a new feature, which reduces network traffic across subnets. However, it prevents the resolver from using the round robin feature as defined in RFC 1794. By using the round robin feature, the server rotates the order of A resource record data returned in a query answer in which multiple resource records of the same type exist for a queried DNS domain name. However, if the resolver is configured for subnet prioritization, the resolver reorders the list to favor IP addresses from networks to which they are directly connected.

If you would prefer to use the round robin feature rather than the subnet prioritization feature, you can do so by changing the value of a registry entry. For more information about configuring the subnet prioritization feature, see "Configuring Subnet Prioritization" earlier in this chapter.

WINS Lookup record causes zone transfer to a third-party DNS server to fail.

If a zone transfer from a Windows 2000 server to a third-party DNS server fails, check whether the zone includes any WINS or WINS-R records. If it does, you can prevent these records from being propagated to a secondary DNS server.

To prevent propagation of WINS lookup records to a secondary DNS server

1. In the DNS console, double-click your DNS server, right-click the zone name that contains the WINS record, and then click **Properties**.
2. In the **Properties** dialog box for the zone, click the **WINS** tab and select the check box **Do not replicate this record.**

To prevent propagation of WINS-R records to a secondary DNS server

1. In the DNS console, double-click your DNS server, right-click the reverse lookup zone that contains the WINS-R record, and then click **Properties**.
2. In the properties page for the zone, click the **WINS-R** tab and select the check box **Do not replicate this record**.

WINS lookup record causes a problem with authoritative data.

If you have a problem with incorrect authoritative data in a zone for which WINS lookup integration is enabled, the erroneous data might be caused by WINS returning incorrect data. You can tell whether WINS is the source of the incorrect data by checking the TTL of the data in an Nslookup query. Normally, the DNS service answers with names stored in authoritative zone data by using the set zone or resource record TTL value. It generally answers only with decreased TTLs when providing answers based on non-authoritative, cached data obtained from other DNS servers during recursive lookups.

However, WINS lookups are an exception. The DNS server represents data from a WINS server as authoritative but stores the data in the server cache only, rather than in zones, and decreases the TTL of the data.

To determine whether data comes from a WINS server

1. At the command prompt, type the following:

   nslookup -d2

   server <*server*>

   where <*server*> is a server that is authoritative for the name that you want to test.

   This starts nslookup in user-interactive, debug mode and makes sure that you are querying the correct server. If you query a server that is not authoritative for the name that you test, you are not able to tell whether the data comes from a WINS server.
2. To test for a WINS forward lookup, type the following:

   set querytype=a

   – Or –

   To test for a WINS reverse lookup, type the following:

   set querytype=ptr
3. Enter the forward or reverse DNS domain name that you want to test.
4. In the response, note whether the server answered authoritatively or non-authoritatively, and note the TTL value.
5. If the server does not answer authoritatively, the source of the data is not a WINS server. However, if the server answered authoritatively, repeat a second query for the name.
6. In the response, note whether the TTL value decreased. If it did, the source of the data is a WINS server.

If you have determined that the data comes from a WINS server, check the WINS server for problems. For more information about checking the WINS server for problems, see "Windows Internet Name Service" in this book.

A zone reappears after you delete it.

In some cases, when you delete a secondary copy of the zone, it might reappear. If you delete a secondary copy of the zone when an Active Directory-integrated copy of the zone exists in Active Directory, and the DNS server from which you delete the secondary copy is configured to load data on startup from Active Directory and the registry, the zone reappears.

If you want to delete a secondary copy of a zone that exists in Active Directory, configure the DNS server to load data on startup from the registry, and then delete the zone from the DNS server that is hosting the secondary copy of the zone. Alternatively, you can completely delete the zone from Active Directory when you are logged into a domain controller that has a copy of the zone.

You see error messages stating that PTR records could not be registered

When the DNS server that is authoritative for the reverse lookup zone cannot or is configured not to perform dynamic updates, the system records errors in the event log stating that PTR records could not be registered. You can eliminate the event log errors by disabling dynamic update registration of PTR records on the DNS client. To disable dynamic update registration, add the **DisableReverseAddressRegistrations** entry, with a value of 1 and a data type of REG_DWORD, to the following registry subkey:

HKEY_LOCAL_MACHINE \SYSTEM \CurrentControlSet \Services \Tcpip \Parameters \Interfaces<*name of the interface*>

where *name of the interface* is the GUID of a network adapter.

Solving Dynamic Update and Secure Dynamic Update Problems

If you have problems with dynamic update, use the following steps to diagnose and solve your problem.

Troubleshooting Dynamic Update

If dynamic update does not register a name or IP address properly, use the following process to diagnose and solve your problem.

- Force the client to renew its registration by typing **ipconfig /registerdns**.
- Check whether dynamic update is enabled for the zone that is authoritative for the name that the client is trying to update.

  For more information about dynamic update and secure dynamic update, see "Dynamic Update and Secure Dynamic Update" earlier in this chapter.
- To rule out other problems, check whether the dynamic update client lists the primary DNS server for the zone as its preferred DNS server.

  This is not necessary for dynamic update to work; however, if the client lists a preferred server other than the primary DNS server for the zone, many other problems might cause the failure, such as a network connectivity problem between the two servers or a prolonged recursive lookup for the primary server of the zone. To ascertain the preferred DNS server for the client, check the IP address configured in the TCP/IP properties of the network connection for the client, or at the command prompt type **ipconfig /all**.

  If the zone is Active Directory-integrated, any DNS server that hosts an Active Directory-integrated copy of the zone can process the updates.
- Check whether the zone is configured for secure dynamic update.

  If the zone is configured for secure dynamic update, the update can fail if zone or record security does not permit this client to make changes to the zone or record, or the update can fail if this client does not have ownership of the name that it is trying to update. To see whether the update failed for one of these reasons, check Event Viewer on the client. For more information about Event Viewer, see "Troubleshooting Tools" earlier in this chapter.

  For information about what to do if the update failed because the zone is configured for secure dynamic update, see "Troubleshooting Secure Dynamic Update" later in this chapter.

Troubleshooting Secure Dynamic Update

Secure dynamic update can prevent a client from creating, modifying, or deleting records, depending on the ACL for the zone and the name. By default, secure dynamic update prevents a client from creating, deleting, or modifying a record if the client is not the original creator of the record. For example, if two computers have the same name and both try to register their names in DNS, dynamic update fails for the client that registers second.

If a client failed to update a name in a zone that is configured for secure dynamic update, the failure could be caused by one of the following conditions:

- *The system time on the client and the system time on the DNS server are not in sync*.
- *You have modified the* **UpdateSecurityLevel** *registry entry to disallow the use of secure dynamic update on the client*. For more information about dynamic update and secure dynamic update, see "Dynamic Update and Secure Dynamic Update" earlier in this chapter.
- *The client does not have the appropriate rights to update the resource record*. You can confirm this by checking the ACL associated with the name to be updated.

  If the client does not have the appropriate rights to update the resource record, check whether the DHCP server registered the name of the client and that the DHCP server is the owner of the corresponding dnsNode object. If so, you might consider placing the DHCP server in the DNSUpdateProxy security group. Any object created by a member of the DNSUpdateProxy security group has no security.

  For more information about the DNSUpdateProxy security group, see "Dynamic Update and Secure Dynamic Update Interoperability Considerations" earlier in this chapter.

Additional Resources

- For more information about DNS, see *DNS and BIND*, 3d ed., by Paul Albitz and Cricket Liu, 1998, Sebastopol, CA: O'Reilly & Associates.
- For more information about Request for Comments (RFC) documents and IETF Internet-Drafts, see the Internet Engineering Task Force (IETF) link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.
