


DNS Processes and Interactions

 [learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/dd197552\(v=ws.10\)](https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/dd197552(v=ws.10))

- Article
- 03/04/2013

Applies To: Windows Server 2008

DNS processes and interactions involve the communications between DNS clients and DNS servers during the resolution of DNS queries and dynamic update, and between DNS servers during name resolution and zone administration. Secondary processes and interactions depend on the support for technologies such as Unicode and WINS.

How DNS queries work

When a DNS client needs to look up a name used in a program, it queries DNS servers to resolve the name. Each query message the client sends contains three pieces of information, specifying a question for the server to answer:

1. A specified DNS domain name, stated as a fully qualified domain name (FQDN).
2. A specified query type, which can either specify a resource record (RR) by type or a specialized type of query operation.
3. A specified class for the DNS domain name. For DNS servers running the Windows operating system, this should always be specified as the Internet (IN) class.

For example, the name specified could be the FQDN for a computer, such as “host-a.example.microsoft.com.”, and the query type specified to look for an address (A) RR by that name. Think of a DNS query as a client asking a server a two-part question, such as “Do you have any A resource records for a computer named ‘hostname.example.microsoft.com.’?” When the client receives an answer from the server, it reads and interprets the answered A RR, learning the IP address for the computer it asked for by name.

DNS queries resolve in a number of different ways. A client can sometimes answer a query locally using cached information obtained from a previous query. The DNS server can use its own cache of resource record information to answer a query. A DNS server can also query or contact other DNS servers on behalf of the requesting client to fully resolve the name, and then send an answer back to the client. This process is known as recursion.

In addition, the client itself can attempt to contact additional DNS servers to resolve a name. When a client does so, it uses separate and additional queries based on referral answers from servers. This process is known as iteration.

In general, the DNS query process occurs in two parts:

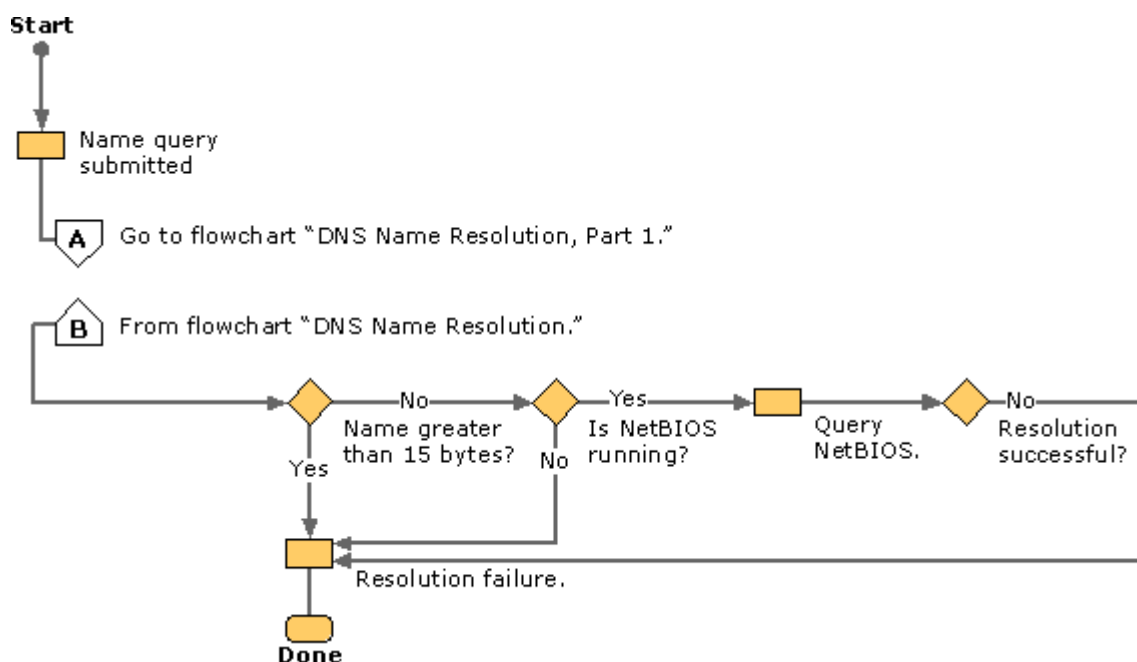
- A name query begins at a client computer and is passed to a resolver, the DNS Client service, for resolution.
- When the query cannot be resolved locally, DNS servers can be queried as needed to resolve the name.

Both of these processes are explained in more detail in the following sections.

Part 1: DNS Client service resolver

The following figure shows an overview of the complete DNS query process.

Overview of DNS Query Process



As shown in the initial steps of the query process, a DNS domain name is used in a program on the local computer. The request is then passed to the DNS Client service for resolution using locally cached information. If the queried name can be resolved, the query is answered and the process is completed.

The local resolver cache can include name information obtained from two possible sources:

- If a Hosts file is configured locally, any host name-to-address mappings from that file are loaded into the cache when the DNS Client service is started.
- Resource records obtained in answered responses from previous DNS queries are added to the cache and kept for a period of time.

If the query does not match an entry in the cache, the resolution process continues with the client querying a DNS server to resolve the name.

Part 2: Querying a DNS server

As indicated in the preceding figure, the client queries a preferred DNS server. The server used during the initial client/server query is selected from a global list.

When the DNS server receives a query, it first checks to see if it can answer the query authoritatively based on resource record information contained in a locally configured zone on the server. If the queried name matches a corresponding RR in local zone information, the server answers authoritatively, using this information to resolve the queried name.

If no zone information exists for the queried name, the server then checks to see if it can resolve the name using locally cached information from previous queries. If a match is found here, the server answers with this information. Again, if the preferred server can answer with a positive matched response from its cache to the requesting client, the query is completed.

If the queried name does not find a matched answer at its preferred server — either from its cache or zone information — the query process can continue, using recursion to fully resolve the name. This involves assistance from other DNS servers to help resolve the name. By default, the DNS Client service asks the server to use a process of recursion to fully resolve names on behalf of the client before returning an answer.

In order for the DNS server to do recursion properly, it first needs some helpful contact information about other DNS servers in the DNS domain namespace. This information is provided in the form of root hints, a list of preliminary RRs that can be used by the DNS service to locate other DNS servers that are authoritative for the root of the DNS domain namespace tree. Root servers are authoritative for the domain root and top-level domains in the DNS domain namespace tree.

By using root hints to find root servers, a DNS server is able to complete the use of recursion. In theory, this process enables any DNS server to locate the servers that are authoritative for any other DNS domain name used at any level in the namespace tree.

For example, consider the use of the recursion process to locate the name “host-b.example.microsoft.com.” when the client queries a single DNS server. The process occurs when a DNS server and client are first started and have no locally cached information available to help resolve a name query. It assumes that the name queried by the client is for a domain name of which the server has no local knowledge, based on its configured zones.

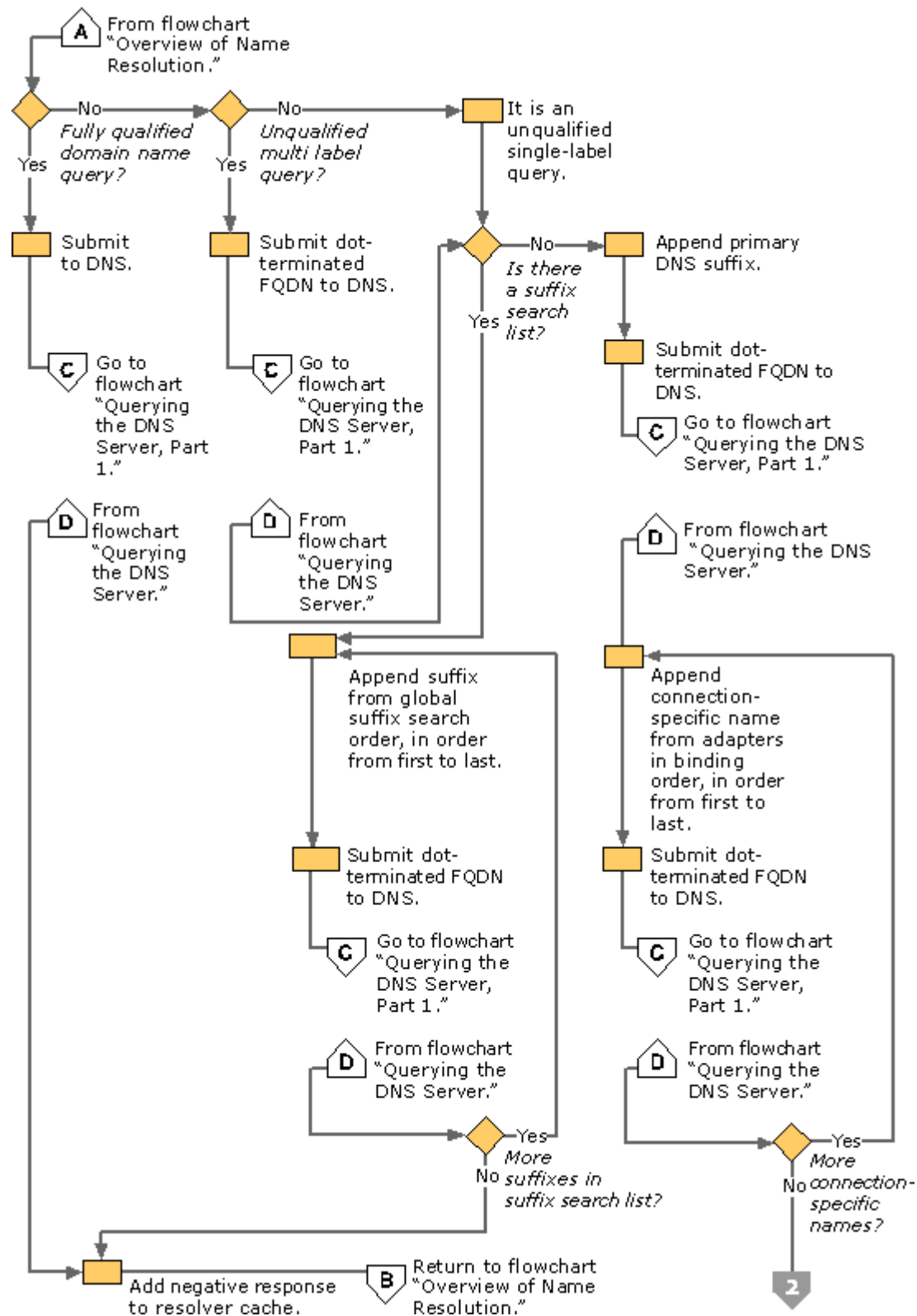
First, the preferred server parses the full name and determines that it needs the location of the server that is authoritative for the top-level domain, “com”. It then uses an iterative query to the “com” DNS server to obtain a referral to the “microsoft.com” server. Next, a referral answer comes from the “microsoft.com” server to the DNS server for “example.microsoft.com”.

Finally, the “example.microsoft.com.” server is contacted. Because this server contains the queried name as part of its configured zones, it responds authoritatively back to the original server that initiated recursion. When the original server receives the response indicating that an authoritative answer was obtained to the requested query, it forwards this answer back to the requesting client and the recursive query process is completed.

Although the recursive query process can be resource-intensive when performed as described above, it has some performance advantages for the DNS server. For example, during the recursion process, the DNS server performing the recursive lookup obtains information about the DNS domain namespace. This information is cached by the server and can be used again to help speed the answering of subsequent queries that use or match it. Over time, this cached information can grow to occupy a significant portion of server memory resources, although it is cleared whenever the DNS service is cycled on and off.

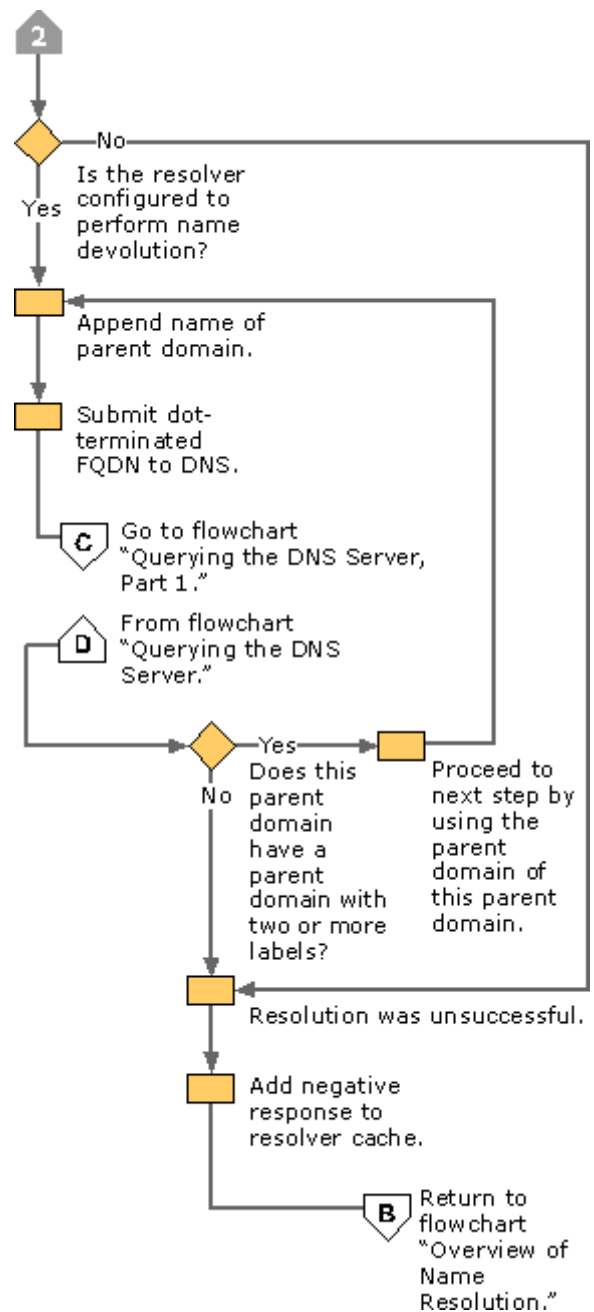
The following three figures illustrate the process by which the DNS client queries the servers on each adapter.

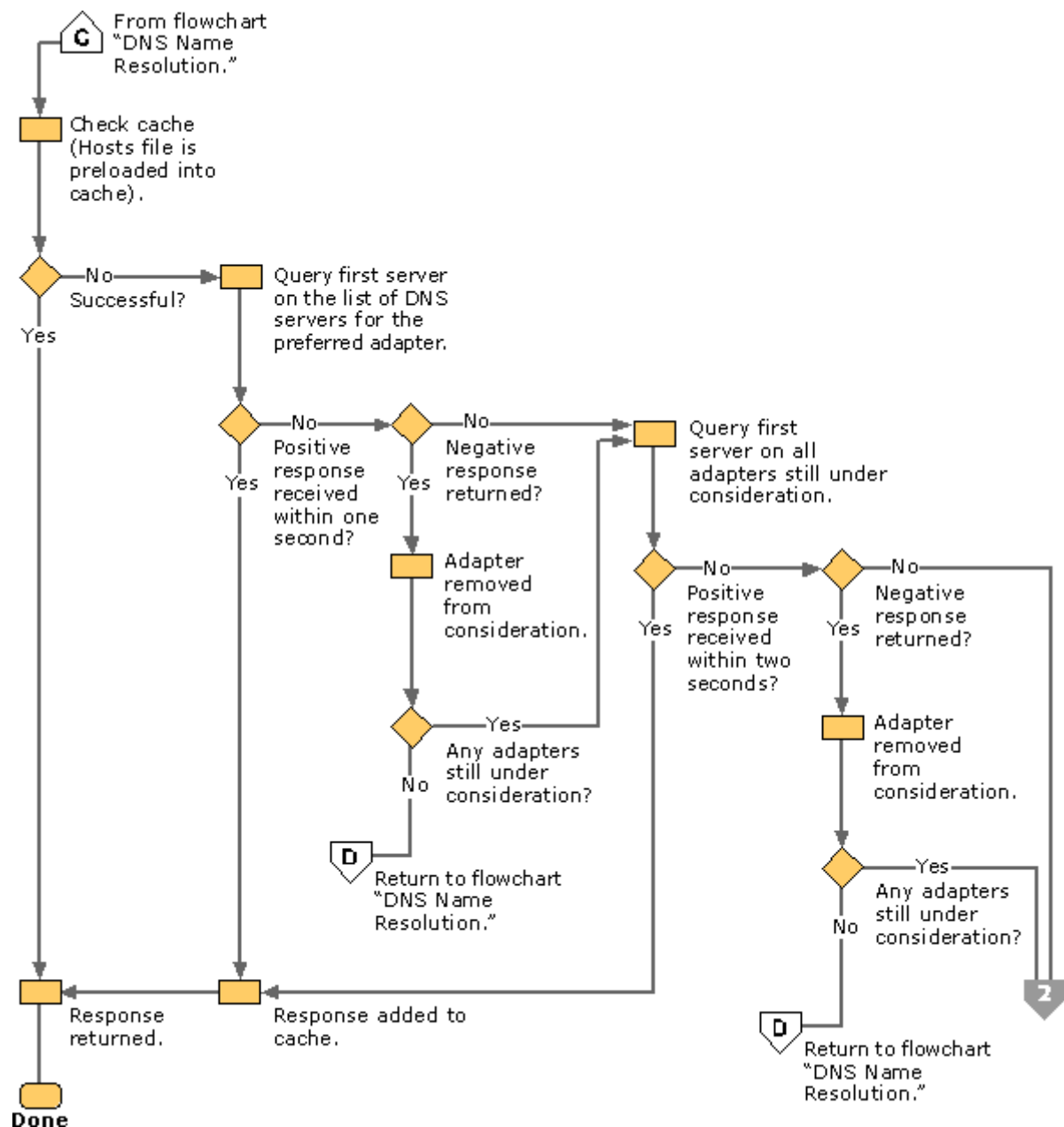
Querying the DNS Server, Part 1



Querying the DNS Server, Part 2

Querying the DNS Server, Part 3





The DNS Client service queries the DNS servers in the following order:

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

5. If the DNS Client service does not receive a response from any DNS server, the DNS client sends the query to all DNS servers on all adapters that are still under consideration and waits eight seconds for a response.

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

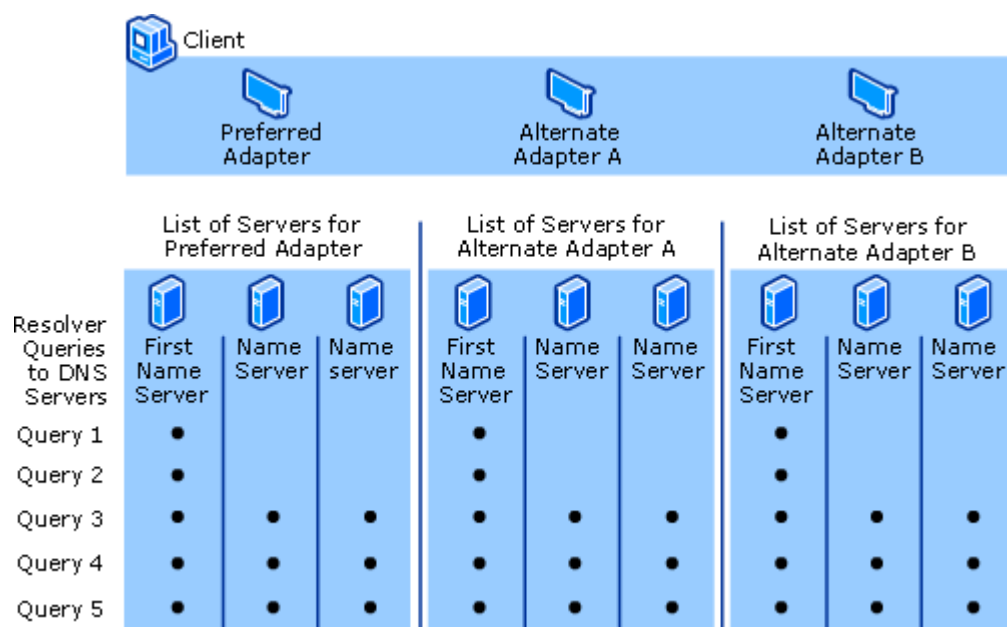
If the DNS Client service has not received a response from any server within eight seconds, the DNS Client service responds with a timeout. Also, if it has not received a response from any DNS server on a specified adapter, then for the next 30 seconds, the DNS Client service responds to all queries destined for servers on that adapter with a timeout and does not query those servers.

If at any point the DNS Client service 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 DNS Client service would not send the query to any other server on the list for Alternate Adapter A.

The DNS Client service keeps track of which servers answer name queries more quickly, and it moves servers up or down on the list based on how quickly they reply to name queries.

The following figure shows how the DNS client queries each server on each adapter.

Multihomed Name Resolution



Alternate query responses

The preceding description of DNS queries assumes that the process ends with a positive response returned to the client. However, queries can return other answers as well. These are the most common query answers:

- An authoritative answer
- A positive answer
- A referral answer
- A negative answer

An authoritative answer is a positive answer returned to the client and delivered with the authority bit set in the DNS message to indicate the answer was obtained from a server with direct authority for the queried name.

A positive response can consist of the queried RR or a list of RRs (also known as an RRset) that fits the queried DNS domain name and record type specified in the query message.

A referral answer contains additional RRs not specified by name or type in the query. This type of answer is returned to the client if the recursion process is not supported. The records are meant to act as helpful reference answers that the client can use to continue the query using iteration. A referral answer contains additional data such as RRs that are other than the type queried. For example, if the queried host name was “www” and no A RRs for this name were found in this zone but a CNAME RR for “www” was found instead, the DNS server can include that information when responding to the client. If the client is able to use iteration, it can make additional queries using the referral information in an attempt to fully resolve the name for itself.

A negative response from the server can indicate that one of two possible results was encountered while the server attempted to process and recursively resolve the query fully and authoritatively:

- An authoritative server reported that the queried name does not exist in the DNS namespace.
- An authoritative server reported that the queried name exists, but no records of the specified type exist for that name.

The resolver passes the results of the query, in the form of either a positive or negative response, back to the requesting program and caches the response.

If the resultant answer to a query is too long to be sent and resolved in a single UDP message packet, the DNS server can initiate a failover response over TCP port 53 to answer the client fully in a TCP connected session.

Disabling the use of recursion on a DNS server is generally done when DNS clients are being limited to resolving names to a specific DNS server, such as one located on your intranet. Recursion might also be disabled when the DNS server is incapable of resolving

external DNS names, and clients are expected to fail over to another DNS server for resolution of these names. If you disable recursion on the DNS server, you will not be able to use forwarders on the same server.

By default, DNS servers use several default timings when performing a recursive query and contacting other DNS servers. These defaults include:

- A recursion retry interval of 3 seconds. This is the length of time the DNS service waits before retrying a query made during a recursive lookup.
- A recursion timeout interval of 8 seconds. This is the length of time the DNS service waits before failing a recursive lookup that has been retried.

Under most circumstances, these parameters do not need adjustment. However, if you are using recursive lookups over a slow-speed wide area network (WAN) link, you might be able to improve server performance and query completion by making slight adjustments to the settings.

How iteration works

Iteration is the type of name resolution used between DNS clients and servers when the following conditions are in effect:

- The client requests the use of recursion, but recursion is disabled on the DNS server.
- The client does not request the use of recursion when querying the DNS server.

An iterative request from a client tells the DNS server that the client expects the best answer the DNS server can provide immediately, without contacting other DNS servers.

When iteration is used, a DNS server answers a client based on its own specific knowledge about the namespace with regard to the names data being queried. For example, if a DNS server on your intranet receives a query from a local client for “www.microsoft.com”, it might return an answer from its names cache. If the queried name is not currently stored in the names cache of the server, the server might respond by providing a referral — that is, a list of NS and A RRs for other DNS servers that are closer to the name queried by the client.

When iteration is used, a DNS server can further assist in a name query resolution beyond giving its own best answer back to the client. For most iterative queries, a client uses its locally configured list of DNS servers to contact other name servers throughout the DNS namespace if its primary DNS server cannot resolve the query.

The Windows Server 2008 DNS Client service does not perform recursion.

How caching works

As DNS servers process client queries using recursion or iteration, they discover and acquire a significant store of information about the DNS namespace. This information is then cached by the server.

Caching provides a way to speed the performance of DNS resolution for subsequent queries of popular names, while substantially reducing DNS-related query traffic on the network.

As DNS servers make recursive queries on behalf of clients, they temporarily cache resource records. Cached RRs contain information obtained from DNS servers that are authoritative for DNS domain names learned while making iterative queries to search and fully answer a recursive query performed on behalf of a client. Later, when other clients place new queries that request RR information matching cached RRs, the DNS server can use the cached RR information to answer them.

When information is cached, a Time-To-Live (TTL) value applies to all cached RRs. As long as the TTL for a cached RR does not expire, a DNS server can continue to cache and use the RR again when answering queries by its clients that match these RRs. Caching TTL values used by RRs in most zone configurations are assigned the minimum (default) TTL which is set in the zone's Start of Authority (SOA) RR. By default, the minimum TTL is 3,600 seconds (one hour) but can be adjusted or, if needed, individual caching TTLs can be set at each RR.

Note

By default, the DNS Server service uses a root hints file, `cache.dns`, that is stored in the `systemroot\System32\Dns` folder on the server computer. This file contains the NS and A RRs for the root servers of the DNS namespace (the Internet root servers or intranet root servers). When the DNS Server service is started, the root server list is queried for a current list of all the root servers. The results of the query are used to update the root hints file. This operation is also performed periodically while the service is running. When changes are made to the root hints by an administrator, these changes are written back to the root hints file.

Reverse lookup

In most DNS lookups, clients typically perform a forward lookup, which is a search based on the DNS name of another computer as stored in an address (A) RR. This type of query expects an IP address as the resource data for the answered response.

DNS also provides a reverse lookup process, enabling clients to use a known IP address during a name query and to look up a computer name based on its address. A reverse lookup takes the form of a question, such as "Can you tell me the DNS name of the computer that uses the IP address 192.168.1.20?"

DNS was not originally designed to support this type of query. One problem for supporting the reverse query process is the difference in how the DNS namespace organizes and indexes names and how IP addresses are assigned. If the only method available to answer the previous question was to search all domains in the DNS namespace, a reverse query would take too long and require too much processing to be useful.

To solve this problem, a special domain called the in-addr.arpa domain was defined in the DNS standards and reserved in the Internet DNS namespace to provide a practical and reliable way to perform reverse queries. To create the reverse namespace, subdomains within the in-addr.arpa domain are formed using the reverse ordering of the numbers in the dotted-decimal notation of IP addresses.

This reversed ordering of the domains for each octet value is needed because, unlike DNS names, when IP addresses are read from left to right, they are interpreted in the opposite manner. When an IP address is read from left to right, it is viewed from its most generalized information (an IP network address) in the first part of the address to the more specific information (an IP host address) contained in the last octets.

For this reason, the order of IP address octets must be reversed when building the in-addr.arpa domain tree. The IP addresses of the DNS in-addr.arpa tree can be delegated to companies as they are assigned a specific or limited set of IP addresses within the Internet-defined address classes.

Finally, the in-addr.arpa domain tree, as built into DNS, requires that an additional RR type — the pointer (PTR) RR — be defined. This RR is used to create a mapping in the reverse lookup zone that typically corresponds to a host (A) named RR for the DNS computer name of a host in its forward lookup zone.

The in-addr.arpa domain applies for use in all TCP/IP networks that are based on Internet Protocol version 4 (IPv4) addressing. The New Zone Wizard automatically assumes that you are using this domain when creating a new reverse lookup zone.

If you are installing DNS and configuring reverse lookup zones for an Internet Protocol version 6 (IPv6) network, you can specify an exact name in the New Zone Wizard. This will permit you to create reverse lookup zones in the DNS console that can be used to support IPv6 networks, which use a different special domain name, the ip6.arpa domain.

For information about IPv6 and DNS, including examples of how to create and use ip6.arpa domain names as described in RFC 1886 (“DNS Extensions to support IP version 6”), see [DNS Reference Information](#).

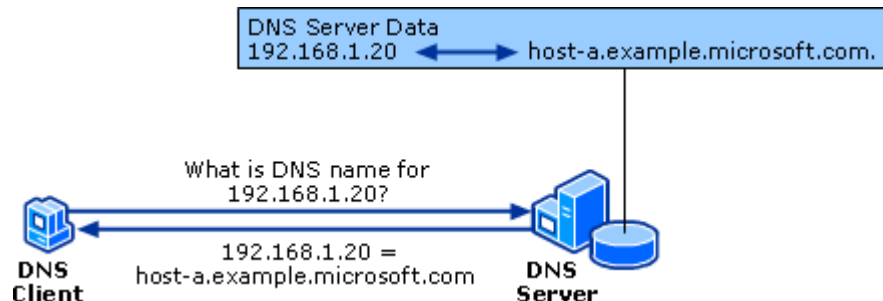
Note

The configuration of PTR RRs and reverse lookup zones for identifying hosts by reverse query is strictly an optional part of the DNS standard implementation. You are not required to use reverse lookup zones, although for some networked applications, they are used to perform security checks.

Example: Reverse query (for IPv4 networks)

The following figure shows an example of a reverse query initiated by a DNS client (host-b) to learn the name of another host (host-a) based on its IP address, 192.168.1.20.

Reverse Query



The reverse query process as shown in this figure occurs in the following steps:

1. The client, "host-b", queries the DNS server for a pointer (PTR) RR that maps to the IP address of 192.168.1.20 for "host-a".

Because the query is for PTR records, the resolver reverses the address and appends the in-addr.arpa domain to the end of the reverse address. This forms the fully qualified domain name ("20.1.168.192.in-addr.arpa.") to be searched in a reverse lookup zone.

2. After it has been located, the authoritative DNS server for "20.1.168.192.in-addr.arpa" can respond with the PTR record information. This includes the DNS domain name for "host-a", completing the reverse lookup.

Keep in mind that if the queried reverse name is not answerable from the DNS server, normal DNS resolution (either recursion or iteration) can be used to locate a DNS server that is authoritative for the reverse lookup zone and that contains the queried name. In this sense, the name resolution process used in a reverse lookup is identical to that of a forward lookup.

Note

The DNS console provides a means for you to configure a subnetted reverse lookup "classless" zone when the **Advanced** view is selected. This allows you to configure a zone in the in-addr.arpa domain for a limited set of assigned IP addresses where a nondefault IP subnet mask is used with those addresses.

Forwarding

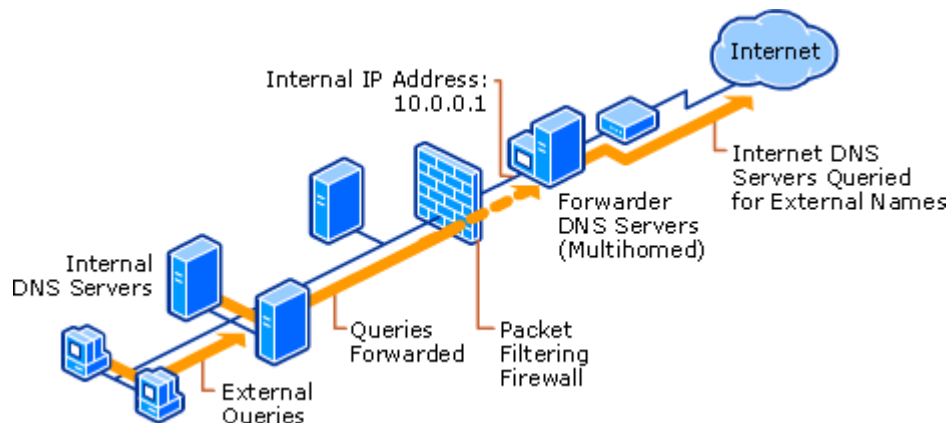
A forwarder is a DNS server on a network used to forward DNS queries for external DNS names to DNS servers outside of that network. You can also forward queries according to specific domain names using conditional forwarders.

A DNS server on a network is designated as a forwarder by having the other DNS servers in the network forward the queries they cannot resolve locally to that DNS server. By using a forwarder, you can manage name resolution for names outside of your network, such as names on the Internet, and improve the efficiency of name resolution for the computers in your network.

Directing name queries using forwarders

The following figure illustrates how external name queries are directed using forwarders.

External Name Queries Directed Using Forwarders



Without having a specific DNS server designated as a forwarder, all DNS servers can send queries outside of a network using their root hints. As a result, a lot of internal, and possibly critical, DNS information can be exposed on the Internet. In addition to this security and privacy issue, this method of resolution can result in a large volume of external traffic that is costly and inefficient for a network with a slow Internet connection or a company with high Internet service costs.

When you designate a DNS server as a forwarder, you make that forwarder responsible for handling external traffic, thereby limiting DNS server exposure to the Internet. A forwarder will build up a large cache of external DNS information because all of the external DNS queries in the network are resolved through it. In a short amount of time, a forwarder will resolve a good portion of external DNS queries using this cached data and thereby decrease the Internet traffic over the network and the response time for DNS clients.

Behavior of a DNS server configured to use forwarding

A DNS server configured to use a forwarder will behave differently from a DNS server that is not configured to use a forwarder. A DNS server configured to use a forwarder behaves as follows:

1. When the DNS server receives a query, it attempts to resolve this query using the primary and secondary zones that it hosts and its cache.

2. If the query cannot be resolved using this local data, then it will forward the query to the DNS server designated as a forwarder.
3. The DNS server will wait briefly for an answer from the forwarder before attempting to contact the DNS servers specified in its root hints.

When a DNS server forwards a query to a forwarder it sends a recursive query to the forwarder. This is different from the iterative query that a DNS server will send to another DNS server during standard name resolution (name resolution that does not involve a forwarder).

Forwarding sequence

The sequence in which the forwarders configured on a DNS server are used is determined by the order in which the IP addresses are listed on the DNS server. After the DNS server forwards the query to the forwarder with the first IP address, it waits a short period for an answer from that forwarder (according to the DNS server's timeout setting) before resuming the forwarding operation with the next IP address. It continues this process until it receives an affirmative answer from a forwarder.

Unlike conventional resolution, where a roundtrip time (RTT) is associated with each server, the IP addresses in the forwarders list are not ordered according to roundtrip time and must be reordered manually to change preference.

Forwarders and delegation

A DNS server configured with a forwarder and hosting a parent zone will use its delegation information before forwarding queries. If no delegation record exists for the DNS name in the query, then the DNS server will use its forwarders to resolve the query.

Forwarders and root servers

A common error when configuring forwarding is to attempt to configure forwarding on the root servers of a private DNS namespace. The goal of attempting to configure forwarding on root servers for a private DNS namespace is to forward all offsite queries to Internet DNS servers. Root servers cannot be configured with standard forwarding. If a root server is queried about any domain name, then it will refer to a DNS server that can answer the question (from its local zones, cache), or it will respond with a failure (NXDOMAIN), but it cannot be configured to forward to specific servers.

Note

A root server can be configured with a conditional forwarder. Conditional forwarding can be used to forward queries between root servers in separate DNS namespaces, although the DNS servers for the top-level domains in the namespace are better suited for this method of resolution.

Conditional forwarders

A conditional forwarder is a DNS server on a network that is used to forward DNS queries according to the DNS domain name in the query. For example, a DNS server can be configured to forward all the queries it receives for names ending with widgets.example.com to the IP address of a DNS server or to the IP addresses of multiple DNS servers.

Intranet name resolution

A conditional forwarder can be used to improve name resolution for domains within your intranet. Intranet name resolution can be improved by configuring DNS servers with forwarders for specific internal domain names. For example, all DNS servers in the domain widgets.example.com could be configured to forward queries for names that end with test.example.com to the authoritative DNS servers for merged.widgets.example.com, thereby removing the step of querying the root servers of example.com, or removing the step of configuring DNS servers in the widgets.example.com zone with secondary zones for test.example.com.

Internet name resolution

DNS servers can use conditional forwarders to resolve queries between the DNS domain names of companies that share information. For example, two companies, Widgets Toys and Tailspin Toys, want to improve how the DNS clients of Widgets Toys resolve the names of the DNS clients of Tailspin Toys. The administrators from Tailspin Toys inform the administrators of Widgets Toys about the set of DNS servers in the Tailspin Toys network where Widgets can send queries for the domain dolls.tailspintoys.com. The DNS servers within the Widgets Toys network are configured to forward all queries for names ending with dolls.tailspintoys.com to the designated DNS servers in the network for Tailspin Toys. Consequently, the DNS servers in the Widgets Toys network do not need to query their internal root servers, or the Internet root servers, to resolve queries for names ending with dolls.tailspintoys.com.

Dynamic update

Dynamic update enables DNS client computers to register and dynamically update their RRs with a DNS server whenever changes occur. This reduces the need for manual administration of zone records, especially for clients that frequently move or change locations and use DHCP to obtain an IP address.

The DNS Client and Server services support the use of dynamic updates, as described in RFC 2136, "Dynamic Updates in the Domain Name System." The DNS Server service allows dynamic update to be enabled or disabled on a per-zone basis at each server configured to load either a standard primary or directory-integrated zone. By default, the Windows Server 2008 DNS Client service will dynamically update host (A) RRs in DNS

when configured for TCP/IP. The Windows Server 2008 DNS Server service is configured, by default, to allow only secure dynamic update. You must change this configuration if you will be using dynamic update only.

Protocol description

RFC 2136 introduces a new opcode or message format called UPDATE. The update message can add and delete RRs from a specified zone as well as test for prerequisite conditions. Update is atomic, that is, all prerequisites must be satisfied or no update operation will take place.

As in any conventional DNS implementation, the zone update must be committed on a primary DNS server for that zone. If an update is received by a secondary DNS server, it will be forwarded up the replication topology until it reaches the primary DNS server. In the case of an Active Directory-integrated zone, an update for a resource record in a zone can be sent to any DNS server running on an Active Directory domain controller whose data store contains the zone.

A zone transfer process will always lock a zone so that a secondary DNS server receives a consistent zone view while transferring the zone data. When the zone is locked, it can no longer accept dynamic updates. If the zone is large and is locked very often for zone transfer purposes, it will starve dynamic update clients, and the system can become unstable. The Windows Server 2008 DNS Server service queues the update requests that arrived during the zone transfer and processes them after the zone transfer is completed.

How client and server computers update their DNS names

By default, computers that are statically configured for TCP/IP attempt to dynamically register host (A) and pointer (PTR) RRs for IP addresses configured and used by their installed network connections. All computers register records based on their FQDN.

The following defaults also apply to how computers update their DNS names:

- By default, the DNS client on Windows XP does not attempt dynamic update over a remote access or virtual private network (VPN) connection. To modify this configuration, you can modify the advanced TCP/IP settings of the particular network connection or modify the registry.
- By default, the DNS client does not attempt dynamic update of top-level domain (TLD) zones. Any zone named with a single-label name is considered a TLD zone, for example, com, edu, blank, my-company.

- By default, the primary DNS suffix portion of a computer's FQDN is the same as the name of the Active Directory domain to which the computer is joined. To allow different primary DNS suffixes, a domain administrator can create a restricted list of allowed suffixes by modifying the **msDS-AllowedDNSSuffixes** attribute in the domain object container. This attribute is managed by the domain administrator using Active Directory Service Interfaces (ADSI) or the Lightweight Directory Access Protocol (LDAP).

Dynamic updates can be sent for any of the following reasons or events:

- An IP address is added, removed, or modified in the TCP/IP properties configuration for any one of the installed network connections.
- At startup time, when the computer is turned on.
- A member server is promoted to a domain controller.
- An IP address lease changes or renews with the DHCP server any one of the installed network connections, for example, when the computer is started or if the **ipconfig /renew** command is used.
- The **ipconfig /registerdns** command is used to manually force a refresh of the client name registration in DNS.

Important

On DHCP client computers running Windows Vista® or later, if you type **ipconfig /registerdns** at a command prompt, the DNS client service will attempt to directly register its DNS record., bypassing the DHCP server. This occurs even if the DHCP server is configured to **Always dynamically update DNS A and PTR records**.

If the client does not have permission to update its resource record, the registration will silently fail. If the DNS client has this permission, the resource record will be updated and permissions can be reset such that the DHCP server is no longer able to perform future updates on the resource record.

The recommended method to update DNS registration for DHCP clients running Windows Vista or later is to use **ipconfig /renew**. Do not use **ipconfig /registerdns**.

When one of the previous events triggers a dynamic update, the DNS Client service (not the DHCP Client service as occurred in previous operating systems) sends updates. This is designed so that if a change to the IP address information occurs, corresponding updates in DNS are performed to synchronize name-to-address mappings for the computer. The DNS Client service performs this function for all network connections used on the system, including connections not configured to use DHCP.

This update process assumes that installation defaults are in effect for servers running Windows Server 2008. Specific names and update behavior is tunable where advanced TCP/IP properties are configured to use non-default DNS settings.

In addition to the full computer name (or primary name) of the computer, additional connection-specific DNS names can be configured and optionally registered or updated in DNS.

Example: How dynamic update works

Dynamic updates are typically requested when either a DNS name or IP address changes on the computer. For example, suppose a client named “oldhost” is first configured in System properties with the following names:

Computer name	oldhost
DNS domain name of computer	example.microsoft.com
Full computer name	oldhost.example.microsoft.com

In this example, no connection-specific DNS domain names are configured for the computer. Later, the computer is renamed from “oldhost” to “newhost”, resulting in the following name changes on the system:

Computer name	newhost
DNS domain name of computer	example.microsoft.com
Full computer name	newhost.example.microsoft.com

After the name change is applied in **System properties**, you are prompted to restart the computer. When the computer restarts Windows, the DNS Client service performs the following sequence to update DNS:

1. The DNS Client service sends an SOA type query using the DNS domain name of the computer.

The client computer uses the currently configured FQDN of the computer (such as “newhost.example.microsoft.com”) as the name specified in this query.

2. The authoritative DNS server for the zone containing the client FQDN responds to the SOA-type query.

For standard primary zones, the primary server (owner) returned in the SOA query response is fixed and static. It always matches the exact DNS name as it appears in the SOA RR stored with the zone. If, however, the zone being updated is directory-integrated, any DNS server that is running on a domain controller for the Active Directory domain in the FQDN can respond and dynamically insert its own name as the primary server (owner) of the zone in the SOA query response.

3. The DNS Client service then attempts to contact the primary DNS server.

The client processes the SOA query response for its name to determine the IP address of the DNS server authorized as the primary server for accepting its name. It then proceeds to perform the following sequence of steps as needed to contact and dynamically update its primary server:

- It sends a dynamic update request to the primary server determined in the SOA query response.
- If the update succeeds, no further action is taken.
- If this update fails, the client next sends an NS-type query for the zone name specified in the SOA record.
- When it receives a response to this query, it sends an SOA query to the first DNS server listed in the response.
- After the SOA query is resolved, the client sends a dynamic update to the server specified in the returned SOA record.
- If the update succeeds, no further action is taken.
- If this update fails, then the client repeats the SOA query process by sending to the next DNS server listed in the response.

4. After the primary DNS server that can perform the update is contacted, the client sends the update request and the DNS server processes it.

The contents of the update request include instructions to add A (and possibly PTR) RRs for “newhost.example.microsoft.com” and remove these same record types for “oldhost.example.microsoft.com”, the name that was previously registered.

The DNS server also checks to ensure that updates are permitted for the client request. For standard primary zones, dynamic updates are not secured, so any client attempt to update succeeds. For Active Directory–integrated zones, updates are secured and performed using directory-based security settings. For more information, see “Secure dynamic update” later in this topic.

Dynamic updates are sent or refreshed periodically. By default, computers send a refresh once every seven days. If the update results in no changes to zone data, the zone remains at its current version and no changes are written. Updates result in actual zone changes or increased zone transfer only if names or addresses actually change.

Note that names are not removed from DNS zones if they become inactive or are not updated within the refresh interval (seven days). DNS does not use a mechanism to release or tombstone names, although DNS clients do attempt to delete or update old name records when a new name or address change is applied.

When the DNS Client service registers A and PTR RRs for a computer, it uses a default caching Time To Live (TTL) of 15 minutes for host records. This determines how long other DNS servers and clients cache a computer's records when they are included in a query response.

DNS and DHCP clients and servers

Windows DNS clients are dynamic update-aware and can initiate the dynamic update process. A DNS 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 updates the A and PTR RRs of the client. Depending on the negotiation process, the DNS client, the DHCP server, or both, update the records by sending the dynamic update requests to the primary DNS servers that are authoritative for the names that are to be updated.

Clients and servers that are running versions of Windows earlier than Windows 2000 do not support dynamic update. The Windows Server 2008 DHCP Server service can perform dynamic updates on behalf of clients that do not support the DHCP Client service FQDN option (which is described in the following section). For example, clients that are running Microsoft Windows® 95, Windows 98, and Windows NT do not support the FQDN option. However, this functionality can be enabled in the **DNS** tab of the server properties for the DHCP console. 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 RRs.

In some cases, stale PTR or A RRs can appear on DNS servers when the lease of a DHCP client expires. For example, when a DNS client running Windows Vista® or Windows Server 2008 tries to negotiate a dynamic update procedure with a DHCP server running Windows NT 4.0, the DNS client must register both A and PTR RRs itself. Later, if the client running Windows 2000 is improperly removed from the network, the client cannot deregister its A and PTR RRs and they become stale.

If a stale A RR appears in a zone that allows only secure dynamic updates, no computer is able to register any other RR for the name in that A RR. To prevent problems with stale PTR and A RRs, you can enable the aging and scavenging feature. For more information

about the aging and scavenging feature, see “Understanding aging and scavenging” in this topic.

To provide fault tolerance for dynamic updates, consider Active Directory integration for those zones that accept dynamic updates from Windows Server 2008 network-based clients. To speed up the discovery of authoritative DNS servers, you can configure each client with a list of preferred and alternate DNS servers that are primary for that directory-integrated zone. If a client fails to update the zone with its preferred DNS server because the DNS server is unavailable, the client can try an alternate server. When the preferred DNS server becomes available, it loads the updated, directory-integrated zone that includes the update from the client.

Dynamic update process for network connections configured by DHCP

To initiate the dynamic update process, the DHCP client sends its FQDN to the DHCP server in the DHCPREQUEST packet by using the DHCP Client service FQDN option. The DHCP server then replies to the DHCP client by sending a DHCP acknowledgment (DHCPACK) message that includes the FQDN option (option code 81).

The following table lists the fields of the FQDN option of the DHCPREQUEST packet.

Fields in the FQDN option of the DHCPREQUEST packet

Field	Explanation
Code	Specifies the code for this option (81).
Len	Specifies the length, in octets, of this option (minimum of 4).
Flags	Can be one of the following values: 0. Client wants to register the A RR and requests that the server update the PTR RR. 1. Client wants server to register the A and PTR RRs. 3. DHCP server registers the A and PTR RRs regardless of the request of the client.
RCODE1 and RCODE2	The DHCP server uses these fields to specify the response code from the A and PTR RRs registrations 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.

The conditions under which DHCP clients send the FQDN option and the actions taken by DHCP servers depend on the operating system that the client and server are running and how the client and server are configured.

The client requests a dynamic update depending on whether it is running Windows Server 2008 or earlier. It also depends on the client configuration. Clients can take any of the following actions:

- By default, the Windows Server 2008 DHCP Client service sends the FQDN option with the Flags field set to 0 to request that the client update the A RR, and the DHCP Server service updates the PTR RR. After the client sends the FQDN option, it waits for a response from the DHCP server. Unless the DHCP server sets the Flags field to 3, the DNS client then initiates an update for the A RR. If the DHCP server does not support or is not configured to perform registration of the DNS record, then no FQDN is included in the DHCP server's response and the DNS client attempts registration of the A and PTR RRs.
- If the DHCP client is running a Windows operating system earlier than Windows 2000, or if the client is Windows 2000 and it is configured not to register DNS resource records, then the client does not send the FQDN option. In this case, the client does not update either record.

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

In the following cases, the DHCP server does not perform any action:

- The DHCP server (for example, a server running Windows NT 4.0) does not support dynamic update.
- The DHCP server is running Windows Server 2008 and is configured not to do dynamic updates for clients that do not support the FQDN option.
- The DHCP server is running Windows Server 2008 and is configured not to register DNS resource records.

If the Windows network-based DHCP client requests that the server updates the PTR RR but not the A RR, behavior depends on the type of DHCP server and how it is configured. The server can perform any of the following actions:

- If the DHCP server is running Windows Server 2008 and is configured not to perform dynamic updates, its response does not contain the FQDN option and does not update either RR. In this case, the DNS client attempts to update both the A and PTR RRs, if it capable.

- If the DHCP server is running Windows Server 2008 and is configured to update according to the request of the DHCP client, the server attempts to update the PTR RR. The DHCP server DHCPACK message to the DHCP client contains the FQDN option with the Flags field set to 0, confirming that the DHCP server updates the PTR record. The DNS client then attempts to update the A RR, if it is capable.

If the DHCP server is running Windows Server 2008 and is configured to always update A and PTR both records, the DHCP server attempts to update both RRs. The DHCP server DHCPACK message to the DHCP client contains the FQDN option with the Flags field set to 3, notifying the DHCP client that the DHCP server updates A and PTR records. In this case, the DNS client does not attempt to update either RR.

Dynamic update process for 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 RRs every time they start and then every 24 hours in case the records become corrupted or need to be refreshed in the DNS database.

Remote access clients can dynamically update A and PTR RRs when a dial-up connection is made. They can also attempt to withdraw, or deregister, the A and PTR RRs when the user closes down the connection explicitly. Computers running Windows Server 2008 with a remote access network connection attempt the dynamic registration of the A and PTR records corresponding to the IP address of this connection. By default, the DNS Client service on Windows XP does not attempt dynamic update over a remote access or VPN connection. To modify this configuration, you can modify the advanced TCP/IP settings of the particular network connection or modify the registry.

In all operating systems, if a remote access client does not receive a successful response from the attempt to deregister a DNS resource record, or if for any other reason fails to deregister a resource record within four seconds, the DNS client closes the connection. In such cases, the DNS database might contain a stale record.

If the remote access client fails to deregister a DNS 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, but the remote access server attempts to deregister the PTR RR when the client is disconnected.

By default, Windows Server 2008 DNS Client service dial-up networking clients do not attempt to update A and PTR records automatically. Due to the nature of their business, it is common that ISPs do not enable dynamic updating of DNS information by their customers. If you use an ISP that does not support dynamic update, configure the connection properties to prevent the computer from performing dynamic updates.

Dynamic update process for multihomed clients

If a dynamic update client is multihomed (has more than one network connection and associated IP address), it registers all IP addresses for each network connection. If you do not want it to register these IP addresses, you can configure the network connection to not register IP addresses.

Important

This behavior was changed in Windows Server 2008. Previously, a multihomed client would register only the first IP address for each network connection by default. For more information, see Microsoft Knowledge Base article 975808.

The dynamic update client does not register all IP addresses with the DNS servers in all namespaces that the computer is connected to. For example, a multihomed computer, client1.noam.example.com, 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.

Time to Live

Whenever a dynamic update client registers in DNS, the associated A and PTR RRs include the Time to Live (TTL), which by default is set to 10 minutes for records registered by the Net Logon service, and 15 minutes for records registered by the DHCP Client service. If the DNS Server service dynamically registers records for its own zones, the default TTL is 20 minutes. You can change the default setting in the registry. A small value causes cached entries to expire sooner, which increases DNS traffic but decreases the risk of cached records becoming outdated. Expiring entries quickly is useful for computers that frequently renew their DHCP leases. Long retention times are useful for computers that renew their DHCP leases infrequently.

Resolving name conflicts

When the DNS Client service attempts to register an A record and it discovers that the authoritative DNS zone already contains an A record for the same name but with a different IP address, by default, the DNS Client service attempts to replace the existing A record (or records) with the new A record containing the IP address of the DNS client. As a result, any computer on the network can modify the existing A record unless secure dynamic update is used. Zones that are configured for secure dynamic update allow only authorized users to modify the resource record.

You can change the default setting so that the DNS Client service ends the registration process and logs the error in Event Viewer, instead of replacing the existing A record.

Secure dynamic update

DNS update security is available only for zones that are integrated into Active Directory. When you integrate a zone into Active Directory, access control list (ACL) editing features are available in the DNS console so you can add or remove users or groups from the ACL for a specified zone or resource record. ACLs are for DNS administration access control only, and do not influence DNS query resolution.

By default, dynamic update security for DNS servers and clients are handled as follows:

- DNS clients attempt to use unsecured dynamic update first. If an unsecured update is refused, clients try to use secure update.

Also, clients use a default update policy that permits them to attempt to overwrite a previously registered resource record, unless they are specifically blocked by update security.

- After a zone becomes Active Directory–integrated, DNS servers running Windows Server 2008 default to allowing only secure dynamic updates.

When using standard zone storage, the default for the DNS Server service is to not allow dynamic updates on its zones. For zones that are either directory-integrated or use standard file-based storage, you can change the zone to allow all dynamic updates, which permits all updates to be accepted.

Dynamic update is a recent additional DNS standard specification, defined in RFC 2136. For more information about RFCs, see [DNS Reference Information](#).

The dynamic registration of DNS resource records can be restricted with the use of registry entries.

How secure dynamic update works

The secure dynamic update process is described as follows:

- To initiate a secure dynamic update, the DNS client first initiates the security context negotiation process, during which the tokens are passed between client and server using TKEY RRs. At the end of the negotiation process, the security context is established.
- Next, the DNS client sends the dynamic update request (containing resource records for the purpose of adding, deleting, or modifying data) to the DNS server, signed using the previously established security context and passing the signature in the TSIG RR, included in the dynamic update packet.
- The server attempts to update Active Directory using the client's credentials and sends the result of the update to the client. These results are signed using the security context and pass the signature in the TSIG RR included in the response.

Secure dynamic update process

The secure dynamic update process is described as follows:

1. The DNS client queries the preferred DNS server to determine which DNS server is authoritative for the domain name it is attempting to update. The preferred DNS server responds with the name of the zone and the primary DNS server that is authoritative for the zone.
2. The DNS client attempts a standard dynamic update, and if the zone is configured to allow only secure dynamic updates (the default configuration for Active Directory-integrated zones), the DNS server refuses the non-secure update. Had the zone been configured for standard dynamic update rather than secure dynamic update, the DNS server would have accepted the DNS client's attempt to add, delete, or modify resource records in that zone.
3. The DNS client and DNS server begin TKEY negotiation.
4. First, the DNS client and DNS server negotiate an underlying security mechanism. Windows dynamic update clients and DNS servers can only use the Kerberos protocol.
5. Next, by using the security mechanism, the DNS client and DNS server verify their respective identities and establish the security context.
6. The DNS client sends the dynamic update request to the DNS server, signed using the established security context. The signature is included in the signature field of the TSIG RR that is included in the dynamic update request packet. The DNS server verifies the origin of the dynamic update packet by using the security context and the TSIG signature.
7. The DNS server attempts to add, delete, or modify resource records in Active Directory. Whether or not it can make the update depends on whether the DNS client has the proper permissions to make the update and whether the prerequisites have been satisfied.
8. The DNS server sends a reply to the DNS client stating whether it was able to make the update, signed using the established security context. The signature is included in the signature field of the TSIG RR that is included in the dynamic update response packet. If the DNS client receives a spoofed reply, it ignores it and waits for a signed response.

Security for DHCP clients that do not support the FQDN option

Windows DHCP clients that do not support the FQDN option (option 81) are not capable of dynamic updates. If you want the A and PTR RRs for these clients dynamically registered in DNS, you must configure the DHCP server to perform dynamic updates on their behalf.

However, having the DHCP server to perform secure dynamic updates on behalf of DHCP clients that do not support the FQDN option is undesirable because when a DHCP server performs a secure dynamic update on a name, that DHCP server becomes the owner of that name, and only that DHCP server can update any record for that name. This can cause problems in some circumstances.

For example, suppose that the DHCP server DHCP1 created an object for the name nt4host1.example.com and then stopped responding, and that later the backup DHCP server, DHCP2, tried to update a record for the same name, nt4host1.example.com. In this situation, 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.example.com, and then the administrator upgraded nt4host1.example.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 DNS records for the name.

To solve this problem, the built-in security group called DnsUpdateProxy is provided. If all DHCP servers are added as members of the DnsUpdateProxy group, one server's records can be updated by another server if the first server fails. Also, because all objects created by the members of the DnsUpdateProxy group are not secured, the first user (that is not a member of the DnsUpdateProxy group) to modify the set of records associated with a DNS name becomes its owner. When legacy clients are upgraded, they can therefore take ownership of their name records at the DNS server. If every DHCP server registering resource records for older clients is a member of the DnsUpdateProxy group, the problems discussed earlier are eliminated.

Securing records when using the DnsUpdateProxy group

DNS domain names that are registered by the DHCP server are not secure when the DHCP server is a member of the DnsUpdateProxy group. As a result, do not use this group in an Active Directory integrated-zone that allows only secure dynamic updates without taking additional steps to allow records created by members of the group to be secured.

To protect against unsecured records, or to allow members of the DnsUpdateProxy group to register records in zones that allow only secured dynamic updates, Windows Server 2008 DHCP and DNS allow you to create a dedicated user account and configure DHCP servers to perform DNS dynamic updates with the user account credentials (user name, password, and domain). The credentials of one dedicated user account can be used by multiple DHCP servers.

The dedicated user account is a standard user account used only to supply DHCP servers with credentials for DNS dynamic update registration. Each DHCP server supplies these credentials when registering names on behalf of DHCP clients using DNS dynamic update. The dedicated user account is created in the same forest where the

primary DNS server for the zone to be updated resides. The dedicated user account can also be located in another forest as long as the forest it resides in has a forest trust established with the forest containing the primary DNS server for the zone to be updated.

When installed on a domain controller, the DHCP Server service inherits the security permissions of the domain controller and has the authority to update or delete any DNS record that is registered in a secure Active Directory-integrated zone (this includes records that were securely registered by other computers running Windows Server 2008, including domain controllers). When installed on a domain controller, configure the DHCP server with the credentials of the dedicated user account to prevent the server from inheriting, and possibly misusing, the power of the domain controller.

Configure a dedicated user account and configure the DHCP Server service with the account credentials under the following circumstances:

- A domain controller is configured to function as a DHCP server.
- The DHCP server is configured to perform DNS dynamic updates on behalf of DHCP clients.
- The DNS zones to be updated by the DHCP server are configured to allow only secure dynamic updates.

After you have created a dedicated user account, you can configure DHCP servers with the user account credentials by using the DHCP console or by using the Netsh command (**netsh dhcp server set dnscredentials**).

Note

-

If the supplied credentials belong to an object (such as a computer) that is a member of the DnsUpdateProxy security group, the next object to register the same name record in DNS will become the record owner.

-

If you have specified credentials (user name, domain, and password) that the DHCP server uses when registering DHCP client computers in DNS, these credentials are not backed up with either synchronous or asynchronous backup. After a DHCP database is restored, new credentials must be configured.

Controlling update access to zones and names

Access to the DNS zones and resource records stored in Active Directory is controlled ACLs. ACLs can be specified for the DNS Server service, an entire zone or for specific DNS names. By default, any authenticated Active Directory user can create the A or PTR RRs in any zone. After an owner name has been created for a zone (regardless of the type of resource record), only the users or groups specified in the ACL for that name that have write permission are enabled to modify records corresponding to that name. While this approach is desirable in most scenarios, some special situations need to be considered separately.

DNSAdmins group

By default, the DNSAdmins group has full control of all zones and records in the Windows Server 2008 domain in which it is specified. In order for a user to be able to enumerate zones in a specific Windows Server 2008 domain, the user (or a group the user belongs to) must be enlisted in the DNSAdmin group.

It is possible that a domain administrator might not want to grant full control to all users listed in the DNSAdmins group. Typically, this would be the result if a domain administrator wanted to grant full control for a specific zone and read-only control for other zones in the domain to a set of users. To accomplish this, the domain administrator can create a separate group for each of the zones, and add specific users to each group. Then the ACL for each zone will contain a group with full control for that zone only. At the same time, all of the groups will be included in the DNSAdmins group, which can be configured to have read permissions only. As a result of the fact that a zone's ACL always contains the DNSAdmins group, all users enlisted in the zone-specific groups will have read permission for all the zones in the domain.

Reserving names

The default DNS Server service configuration of allowing any authenticated user to create a new name in a zone might not be sufficient for environments that require a high level of security. In such cases, the default ACL can be changed to allow for the creation of objects in a zone by certain groups or users only. Per-name administration of ACLs provides another solution to this problem. An administrator can reserve a name in a zone leaving the rest of the zone open for the creation of any new objects by all authenticated users. To accomplish this, an administrator creates a record for the reserved name and sets the appropriate list of groups or users in the ACL. As a result, only the users listed in the ACL will be able to register another record under the reserved name.

Understanding aging and scavenging

DNS servers running Windows Server 2008 support aging and scavenging features. These features are provided as a mechanism for performing cleanup and removal of stale resource records, which can accumulate in zone data over time.

With dynamic update, RRs are automatically added to zones when computers start on the network. However, in some cases, they are not automatically removed when computers leave the network. For example, if a computer registers its own host (A) RR at startup and is later improperly disconnected from the network, its host (A) RR might not be deleted. If your network has mobile users and computers, this situation can occur frequently.

If left unmanaged, the presence of stale RRs in zone data might cause some problems. The following are examples:

- If a large number of stale RRs remain in server zones, they can eventually take up server disk space and cause unnecessarily long zone transfers.
- DNS servers loading zones with stale RRs might use outdated information to answer client queries, potentially causing the clients to experience name resolution problems on the network.
- The accumulation of stale RRs at the DNS server can degrade its performance and responsiveness.
- In some cases, the presence of a stale RR in a zone could prevent a DNS domain name from being used by another computer or host device.

To solve these problems, the DNS Server service has the following features:

- Time stamping, based on the current date and time set at the server computer, for any RRs added dynamically to primary-type zones. In addition, time stamps are recorded in standard primary zones where aging/scavenging is enabled.
- For RRs that you add manually, a time stamp value of zero is used, indicating that they are not affected by the aging process and can remain without limitation in zone data unless you otherwise change their time stamp or delete them.
- Aging of RRs in local data, based on a specified refresh time period, for any eligible zones. Only primary type zones that are loaded by the DNS Server service are eligible to participate in this process.
- Scavenging for any RRs that persist beyond the specified refresh period. When a DNS server performs a scavenging operation, it can determine that RRs have aged to the point of becoming stale and remove them from zone data. Servers can be configured to perform recurring scavenging operations automatically, or you can initiate an immediate scavenging operation at the server.

Note

By default, the aging and scavenging mechanism for the DNS Server service is disabled. It should only be enabled when all parameters are fully understood. Otherwise, the server could be accidentally configured to delete records that should not be deleted. If a record

is accidentally deleted, not only will users fail to resolve queries for that record, but any user can create the record and take ownership of it, even on zones configured for secure dynamic update.

The server uses the contents of each RR-specific time stamp, along with other aging and scavenging properties that you can adjust or configure, to determine when it scavenges records.

Prerequisites for aging and scavenging

Before the aging and scavenging features of DNS can be used, several conditions must be met:

1. Scavenging and aging must be enabled both at the DNS server and on the zone.

By default, aging and scavenging of resource records is disabled.

2. Resource records must either be dynamically added to zones or manually modified to be used in aging and scavenging operations.

Typically, only those resource records added dynamically using the DNS dynamic update protocol are subject to aging and scavenging.

You can, however, enable scavenging for other resource records added through non-dynamic means. For records added to zones in this way, either by loading a text-based zone file from another DNS server or by manually adding them to a zone, a time stamp of zero is set. This makes these records ineligible for use in aging and scavenging operations.

In order to change this default, you can administer these records individually, to reset and permit them to use a current (non-zero) time stamp value. This enables these records to become aged and scavenged.

Note

In the case of changing a zone from standard primary to Active Directory–integrated, you might want to enable scavenging of all existing resource records in the zone. To enable aging for all existing resource records in a zone, you can use the **AgeAllRecords command**, which is available through the dnscmd command-line tool.

Aging and scavenging terminology

The following list indicates new or revised terms that have been introduced to help specifically when discussing aging and scavenging.

Current server time The current date and time on the DNS server. This number can be expressed as an exact numeric value at any point in time.

No-refresh interval An interval of time, determined for each zone, as bounded by the following two events:

- The date and time when the record was last refreshed and its time stamp set.
- The date and time when the record next becomes eligible to be refreshed and have its time stamp reset.

This value is needed to decrease the number of write operations to the Active Directory database. By default, this interval is set to seven days. It should not be increased to an unreasonably high level, because the benefits of the aging and scavenging feature might either be lost or diminished.

Record refresh When a DNS dynamic update is processed for a resource record when only the resource record time stamp, and no other characteristics of the record, are revised. Refreshes generally occur for the following reasons:

- When a computer is restarted on the network and, if at startup, its name and IP address information are consistent with the same name and address information it used prior to being shut down, it sends a refresh to renew its associated resource records for this information.
- A periodic refresh is sent by the computer while it is running.
- The Windows XP and Windows Server 2008 DNS Client service renews DNS registration of client resource records every 24 hours. When this dynamic update occurs, if the dynamic update request does not cause modification to the DNS database, then it is considered to be a refresh and not a resource record update.
- Other network services make refresh attempts, such as DHCP servers, which renew client address leases; cluster servers, which register and update records for a cluster; and the Net Logon service, which can register and update resource records used by Active Directory domain controllers.

Record update When a DNS dynamic update is processed for a resource record where other characteristics of the record in addition to its time stamp are revised. Updates generally occur for the following reasons:

- When a new computer is added to the network and, at startup, it sends an update to register its resource records for the first time with its configured zone.
- When a computer with existing records in the zone has a change in IP address, causing updates to be sent for its revised name-to-address mappings in DNS zone data.
- When the Net Logon service registers a new Active Directory domain controller.

Refresh interval An interval of time, determined for each zone, as bounded by the following two distinct events:

- The earliest date and time when the record becomes eligible to be refreshed and have its time stamp reset.
- The earliest date and time when the record becomes eligible to be scavenged and removed from the zone database.

This value should be large enough to allow all clients to refresh their records. By default, this interval is set to seven days. It should not be increased to an unreasonably high level, because the benefits of the aging and scavenging feature might either be lost or diminished.

Resource record time stamp A date and time value used by the DNS server to determine removal of the resource record when it performs aging and scavenging operations.

Scavenging period When automatic scavenging is enabled at the server, this period represents the time between repetitions of the automated scavenging process. The default value for this is seven days. To prevent deterioration of DNS server performance, the minimum allowed value for this is one hour.

Scavenging servers An optional advanced zone parameter that enables you to specify a restricted list of IP addresses for DNS servers that are enabled to perform scavenging of the zone. By default, if this parameter is not specified, all DNS servers that load a directory-integrated zone (also enabled for scavenging) attempt to perform scavenging of the zone. In some cases, this parameter can be useful if it is preferable that scavenging only be performed at some servers loading the directory-integrated zone. To set this parameter, you must specify the list of IP addresses for the servers enabled to scavenge the zone in the ScavengingServers parameter for the zone. This can be done using the dnscmd command, a command-line based tool for administering Windows DNS servers.

Start scavenging time A specific time, expressed as a number. This time is used by the server to determine when a zone becomes available for scavenging.

When scavenging can start

After all prerequisites for enabling the use of scavenging are met, scavenging can start for a server zone when the current server time is greater than the value of the start scavenging time for the zone.

The server sets the time value to start scavenging on a per-zone basis whenever any one of the following events occurs:

- Dynamic updates are enabled for the zone.

- A change in the state of the **Scavenge stale resource records** check box is applied. You can use the DNS console to modify this setting at either an applicable DNS server or one of its primary zones.
- The DNS server loads a primary zone enabled to use scavenging. This can occur when the server computer is started or when the DNS Server service is started.
- When a zone resumes service after having been paused.

When any of the previous events occur, the DNS server sets the value of start scavenging time by calculating the following sum:

Current server time + Refresh interval = Start scavenging time

This value is used as a basis of comparison during scavenging operations.

Example of the aging and scavenging process for a sample record

To understand the process of aging and scavenging at the server, consider the life span and successive stages of a single resource record, as it is added to a server and zone where this process is in effect and then aged and removed from the database.

1. A sample DNS host, “host-a.example.microsoft.com”, registers its host (A) RR at the DNS server for a zone where aging/scavenging is enabled for use.
2. When registering the record, the DNS server places a time stamp on this record based on current server time.

After the record time stamp is written, the DNS server does not accept refreshes for this record for the duration of the zone no-refresh interval. It can, however, accept updates prior to that time. For example, if the IP address for “host-a.example.microsoft.com” changes, the DNS server can accept the update. In this case, the server also updates (resets) the record time stamp.

3. Upon expiration of the no-refresh period, the server begins to accept attempts to refresh this record.

After the initial no-refresh period ends, the refresh period immediately begins for the record. During this time, the server does not suppress attempts to refresh the record for its remaining life span.

4. During and after the refresh period, if the server receives a refresh for the record, it processes it.

This resets the time stamp for the record based on the method described in step 2.

5. When subsequent scavenging is performed by the server for the “example.microsoft.com” zone, the record (and all other zone records) are examined by the server.

Each record is compared to current server time on the basis of the following sum to determine whether the record should be removed:

Record time stamp + **No-refresh interval** for zone + **Refresh interval** for zone

- If the value of this sum is greater than current server time, no action is taken and the record continues to age in the zone.
- If the value of this sum is less than current server time, the record is deleted both from any zone data currently loaded in server memory and also from the applicable DnsZone object store in Active Directory for the directory-integrated “example.microsoft.com” zone.

Unicode character support

Originally, Internet host names were restricted to the character set specified in RFCs 952 and 1123. These restrictions include limiting names to using uppercase and lowercase letters (A-“Z”, a-z), numbers (0-9) and hyphens (-). In addition, the first character of the DNS name can be a number and names must be encoded and represented using US-ASCII-based characters.

These requirements were maintained when DNS was introduced as part of RFC 1035, one of the core DNS standards specifications. For use of DNS in international settings, this requirement has significant limitations where extended character sets are used for local naming standards.

To remove these limitations, Microsoft expands DNS character support beyond the RFC 1035 specification. The DNS service now provides enhanced default support for UTF-8, a Unicode transformation format.

What is UTF-8?

UTF-8 is the recommended character set for protocols evolving beyond the use of ASCII. The UTF-8 protocol provides for support of extended ASCII characters and translation of UCS-2, a 16-bit Unicode character set that encompasses most of the world’s writing systems. UTF-8 enables a far greater range of names than can be achieved using ASCII or extended ASCII encoding for character data.

Computers running Windows Server 2008 are UTF-8 aware. This means that when UTF-8-encoded characters are received or used as data by the server, the server can load and store this data in its zones. Although Windows-based DNS servers are UTF-8 aware, they remain compatible with other DNS servers that use traditional US-ASCII data encoding and current DNS standards.

How the DNS service implements UTF-8

To provide standards compatibility and interoperability with other DNS implementations, the DNS service uses uniform downcasing of any received character data. In this process, the DNS service converts all uppercase characters used in standard US-ASCII data to lowercase equivalent data for the following reasons:

- To maintain compatibility with current and existing DNS standards.
- To provide interoperability with DNS server implementations that do not recognize or support UTF-8 encoding.

To understand why uniform downcasing was chosen, several related points must first be considered from the current revised Internet standards for DNS. Several key points in the standards pertain directly to how character data is to be handled between DNS servers and other servers and clients. These include the following:

- Any binary string can be used in a DNS name. (RFC 2181)
- DNS servers must be able to compare names in a case-insensitive way. (RFC 1035)
- The original case for character data should be preserved whenever possible as the data is entered into the system. (RFC 1035)

Because case insensitivity is a required part of the core DNS standard and case preservation is an optional recommendation, uniform downcasing was chosen to provide an effective standards-compliant solution. By downcasing UTF-8 encoded names before transmission, other DNS servers (which are not UTF-8 aware) are able to receive and perform successful binary comparisons of the data and obtain the desired results.

Considerations for interoperability with UTF-8

The DNS Server service can be configured to allow or disallow the use of UTF-8 characters on a per-server basis. Although other DNS server implementations that are not UTF-8 aware might be able to accept the transfer of a zone containing UTF-8 encoded names, these servers might not be able to write back those names to a zone file or reload those names from a zone file. Administrators should exercise caution when transferring a zone containing UTF-8 names to a DNS server that is not UTF-8-aware.

Some protocols place restrictions on the characters allowed in a name. In addition, names that are intended to be globally visible (RFC 1958) should contain ASCII-only characters, as recommended in RFC 1123.

The use of UTF-8 for transformation of Unicode characters is not noticeable for general users, but UTF-8-encoded characters can be observed when Network Monitor or another similar tool is used to analyze DNS-related traffic over the physical network.

In addition to DNS server support for the UTF-8 encoding format, the client resolver defaults to using the UTF-8 character encoding format.

Names encoded in UTF-8 format must not exceed the size limits clarified in RFC 2181, which specifies a maximum of 63 octets per label and 255 octets per name. Character count is insufficient to determine size because some UTF-8 characters exceed one octet in length.

The UTF-8 encoding protocol adapts to use with existing DNS protocol implementations that expect US-ASCII characters because representation of US-ASCII characters in UTF-8 is identical, byte for byte, to the US-ASCII representation. DNS client or server implementations that do not recognize UTF-8 characters always encode names in the US-ASCII format. Those names are correctly interpreted by the DNS Server service.

The DNS service provides the ability to configure name checking to allow or restrict the use of UTF-8 characters in DNS data.

By default, multibyte UTF-8 name checking is used, allowing the greatest tolerance when the DNS service processes characters. This is the preferred name-checking method for most privately operated DNS servers that are not providing name service for Internet hosts.

WINS lookup integration

Support for using Windows Internet Name Service (WINS) is provided to look up DNS names that cannot be resolved by querying the DNS domain namespace. To accomplish WINS lookup, two specific resource record types are used and can be enabled for any zones loaded by the DNS service:

- The WINS RR, which can be enabled to integrate WINS lookup into forward lookup zones
- The WINS-R RR, which can be enabled to integrate node adapter status request for reverse lookup zones

WINS resource record

The WINS and DNS services are used to provide name resolution for the NetBIOS namespace and the DNS domain namespace, respectively. Although both DNS and WINS can provide a separate and useful name service to clients, WINS is mainly needed to provide support for older clients and programs that require support for NetBIOS naming.

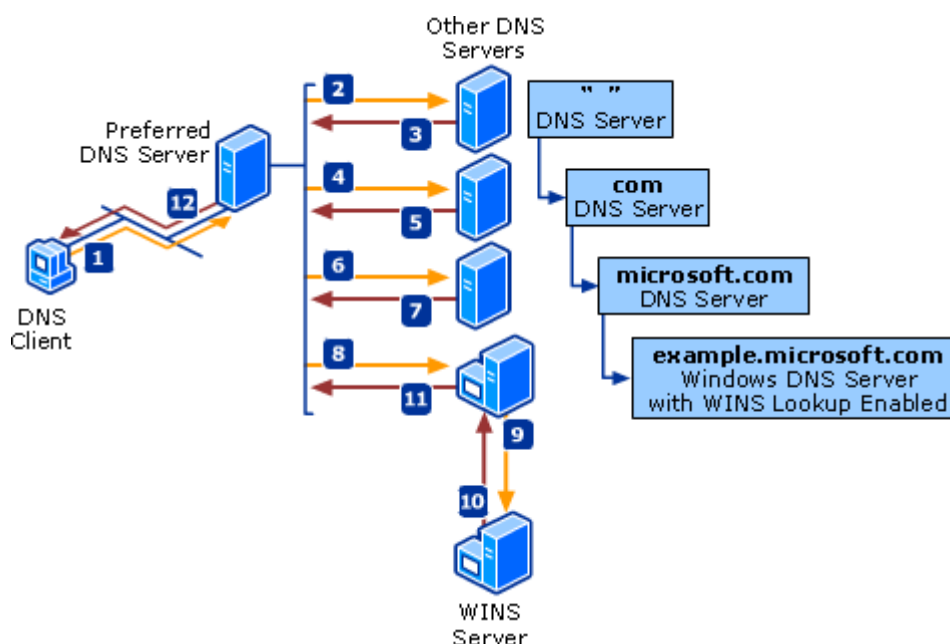
However, the DNS service can work with WINS to provide combined name searches in both namespaces when resolving a DNS domain name not found in zone information. To provide this interoperability, a new record (the WINS record) was defined as part of the zone database file.

The presence of a WINS RR can instruct the DNS service to use WINS to look up any forward queries for host names or names that are not found in the zone database. This functionality is particularly useful for name resolution required by clients that are not WINS-aware (for example, UNIX) for the names of computers not registered with DNS, such as Windows 95 or Windows 98 computers.

How WINS lookup works

The following is an example of a DNS client (host-b) querying its DNS server in an attempt to look up the address for another computer named “host-a.example.microsoft.com.”

WINS Lookup



In step 1, the client queries its preferred DNS server. In steps 2 through 8, the normal process of recursion proceeds 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 example.microsoft.com zone is located through the previous chain of referral answers.

When the DNS server for the example.microsoft.com zone receives the query for host-a, it looks in its configured zone to see if a matching address (A) RR can be found. If no A record is found and the zone is enabled to use WINS lookup, the server does the following:

- The DNS server separates the host part of the name (host-a) from the fully qualified domain name contained in the DNS query.

The host part of the name is the first label in the queried DNS domain name before a period is used in the name.

- The server then sends a NetBIOS name request to the WINS server using the host name, host-a.
- If the WINS server can resolve the name, it returns the IP address to the DNS server.
- The DNS server then compiles an A RR using the IP address resolved through the WINS server and returns this record to the original preferred DNS server that was queried by the requesting client, host-b.
- The preferred DNS server then passes the query answer back to the requesting client.

How WINS reverse lookup works

There is also a WINS-R record or WINS reverse lookup entry that can be enabled and added to reverse lookup zones. However, because the WINS database is not indexed by IP address, the DNS service cannot send a reverse name lookup to WINS to get the name of a computer given its IP address.

Because WINS does not provide reverse lookup capability, the DNS service 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 back onto the NetBIOS name provided in the node status response and forwards the result to the requesting client.

Note

WINS and WINS-R RRs are proprietary to the DNS Server service provided by Windows. You can prevent these resource records from being included in zone transfers to other DNS server implementations.