

Chapter 2

Configure Network Services

THE FOLLOWING 70-410 EXAM OBJECTIVES ARE COVERED IN THIS CHAPTER:

✓ **Deploy and configure DNS service**

- Configure Active Directory integration of primary zones
- Configure forwarders
- Configure Root Hints
- Manage DNS cache
- Create A and PTR resource records

✓ **Deploy and configure Dynamic Host Configuration Protocol (DHCP) service**

- Create and configure scopes
- Configure a DHCP reservation
- Configure DHCP options
- Configure client and server for PXE boot
- Configure DHCP relay agent
- Authorize DHCP server



The Domain Name System (DNS) is one of the key topics that you'll need to understand if you plan to take any of the Microsoft Windows Server 2012 R2 administration exams (70-410,

70-411, 70-412, and so forth).

It's also imperative that you understand DNS to work with Active Directory because it requires DNS to function properly, and many important system functions (including Kerberos authentication and finding domain controllers) are handled through DNS lookups. Windows 2000, Windows XP, Windows Vista, Windows 7, and Windows 8 clients use DNS for name resolution and to find Kerberos key distribution centers (KDCs), global catalog servers, and other services that may be registered in DNS.

By the time you complete this chapter, you will have a deeper understanding of how DNS works and how to set up, configure, manage, and troubleshoot DNS in Microsoft Windows Server 2012 R2.

In this chapter, you'll also learn how to install and manage DHCP, including how to set up plain DHCP scopes, superscopes, and multicast scopes. You'll also learn how to set up integration between Dynamic DNS and DHCP and how to authorize a DHCP server to integrate with Active Directory.



There are two versions of DHCP: DHCP v4 and DHCP v6. In this chapter, I will just say "DHCP server" when referring to the physical DHCP server. If I am referring to a specific version of DHCP, I will specify the version.

Introducing DNS

The *Domain Name System (DNS)* is a service that allows you to resolve a hostname to an Internet Protocol (IP) address. One of the inherent complexities of operating in networked environments is working with multiple protocols and network addresses. Owing largely to the tremendous rise in the popularity of the Internet, however, most environments have transitioned to use *Transmission Control Protocol/Internet Protocol (TCP/IP)* as their primary networking protocol. Microsoft is no exception when it comes to supporting TCP/IP in its workstation and server products. All current versions of Microsoft's operating systems support TCP/IP, as do most other modern operating systems.

An easy way to understand DNS is to think about making a telephone call. If you wanted to call Microsoft and did not know the phone number, you could call information, tell

them the name (Microsoft), and get the telephone number. You would then make the call. Now think about trying to connect to Server1. You don't know the TCP/IP number (the computer's telephone number), so your computer asks DNS (information) for the number of Server1. DNS returns the number, and your system makes the connection (call). DNS is your network's 411, or information, and it returns the TCP/IP data for your network.

TCP/IP is actually a collection of different technologies (protocols and services) that allow computers to function together on a single, large, and heterogeneous network. Some of the major advantages of this protocol include widespread support for hardware, software, and network devices; reliance on a system of standards; and scalability. TCP handles tasks such as sequenced acknowledgments. IP involves many jobs, such as logical subnet assignment and routing.

The Form of an IP Address

To understand DNS, you must first understand how TCP/IP addresses are formed. Because DNS is strictly on a network to support TCP/IP, understanding the basics of TCP/IP is extremely important.



Microsoft exams cover TCP/IP. The TCP/IP material will be covered in Chapter 8, "Configure TCP/IP."

An *IP address* is a logical number that uniquely identifies a computer on a TCP/IP network. TCP/IP allows a computer packet to reach the correct host. Windows Server 2012 R2 works with two versions of TCP/IP: IPv4 and IPv6. An IPv4 address takes the form of four octets (eight binary bits), each of which is represented by a decimal number between 0 and 255. The four numbers are separated by decimal points. For example, all of the following are valid IP addresses:

- 128.45.23.17
- 230.212.43.100
- 10.1.1.1

The dotted decimal notation was created to make it easier for users to deal with IP addresses, but this idea did not go far enough. As a result, another abstraction layer was developed, which used names to represent the dotted decimal notation—the domain name. For example, the IP address 11000000 10101000 00000001 00010101 maps to 192.168.1.21, which in turn might map to server1.company.org, which is how the computer's address is usually presented to the user or application.

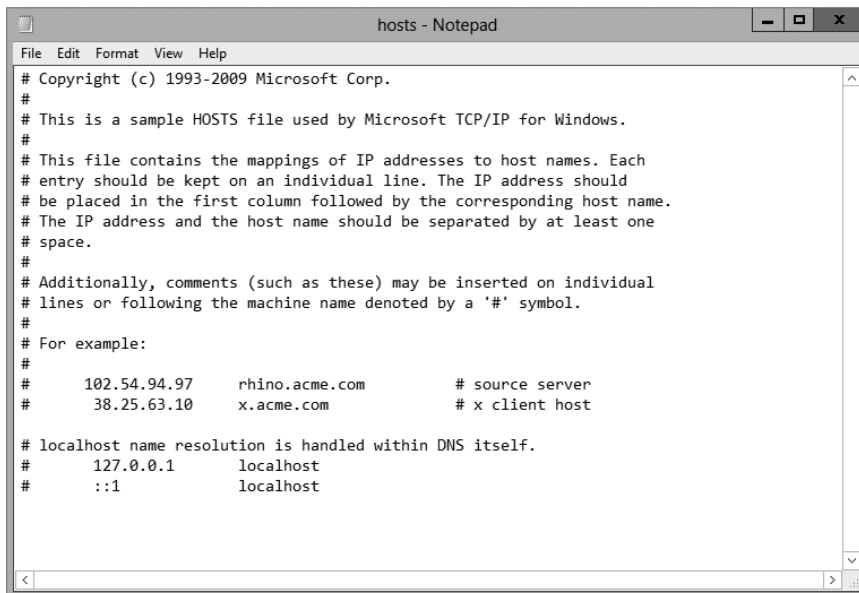
As stated earlier, IPv4 addresses are made up of octets, or the decimal (base 10) representation of 8 bits. It takes four octets to add up to the 32 bits required. IPv6 expands the address space to 128 bits. The address is usually represented in hexadecimal notation as follows:

```
2001:0DB8:0000:0000:1234:0000:A9FE:133E
```

You can tell that the implementation of DNS would make life a lot easier for everyone, even those of us who like to use alphanumeric values. (For example, some of us enjoy pinging the address in lieu of the name.) Fortunately, DNS already has the ability to handle IPv6 addresses using an AAAA record. An A record in IPv4's addressing space is 32 bits, and an AAAA record (4 As) in IPv6's is 128 bits.

Nowadays, most computer users are quite familiar with navigating to DNS-based resources, such as `www.microsoft.com`. To resolve these “friendly” names to TCP/IP addresses that the network stack can use, you need a method for mapping them. Originally, ASCII flat files (often called HOSTS files, as shown in Figure 2.1) were used for this purpose. In some cases, they are still used today in small networks, and they can be useful in helping to troubleshoot name resolution problems.

FIGURE 2.1 HOSTS file



```

File Edit Format View Help
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#       102.54.94.97       rhino.acme.com           # source server
#       38.25.63.10      x.acme.com              # x client host

# localhost name resolution is handled within DNS itself.
#       127.0.0.1        localhost
#       ::1              localhost

```

As the number of machines and network devices grew, it became unwieldy for administrators to manage all of the manual updates required to enter new mappings to a master HOSTS file and distribute it. Clearly, a better system was needed.

As you can see from the sample HOSTS file in Figure 2.1, you can conduct a quick test of the email server's name resolution as follows:

1. Open the HOSTS file: `C:\Windows\System32\drivers\etc`.
2. Add the IP-address-to-hostname mapping.
3. Try to ping the server using the hostname to verify that you can reach it using an easy-to-remember name.

Following these steps should drive home the concept of DNS for you because you can see it working to make your life easier. Now you don't have to remember 10.0.0.10; you only need to remember exchange03. However, you can also see how this method can become unwieldy if you have many hosts that want to use easy-to-remember names instead of IP addresses to locate resources on your network.

When dealing with large networks, users and network administrators must be able to locate the resources they require with minimal searching. Users don't care about the actual physical or logical network address of the machine; they just want to be able to connect to it using a simple name that they can remember.

From a network administrator's standpoint, however, each machine must have its own logical address that makes it part of the network on which it resides. Therefore, some scalable and easy-to-manage method for resolving a machine's logical name to an IP address and then to a domain name is required. DNS was created just for this purpose.

DNS is a hierarchically distributed database. In other words, its layers are arranged in a definite order, and its data is distributed across a wide range of machines, each of which can exert control over a portion of the database. DNS is a standard set of protocols that defines the following:

- A mechanism for querying and updating address information in the database
- A mechanism for replicating the information in the database among servers
- A schema of the database



DNS is defined by a number of requests for comments (RFCs), though primarily by RFC 1034 and RFC 1035.

DNS was originally developed in the early days of the Internet (called ARPAnet at the time) when it was a small network created by the Department of Defense for research purposes. Before DNS, computer names, or hostnames, were manually entered into a HOSTS file located on a centrally administered server. Each site that needed to resolve hostnames outside of its organization had to download this file. As the number of computers on the Internet grew, so did the size of this HOSTS file—and along with it the problems of its management. The need for a new system that would offer features such as scalability, decentralized administration, and support for various data types became more and more obvious. DNS, introduced in 1984, became this new system.

With DNS, the hostnames reside in a database that can be distributed among multiple servers, decreasing the load on any one server and providing the ability to administer this naming system on a per-partition basis. DNS supports hierarchical names and allows for the registration of various data types in addition to the hostname-to-IP-address mapping used in HOSTS files. Database performance is ensured through its distributed nature as well as through caching.

The DNS distributed database establishes an inverted logical tree structure called the *domain namespace*. Each node, or domain, in that space has a unique name. At the top of the tree is the root. This may not sound quite right, which is why the DNS hierarchical

model is described as being an inverted tree, with the root at the top. The root is represented by the null set "". When written, the root node is represented by a single dot (.).

Each node in the DNS can branch out to any number of nodes below it. For example, below the root node are a number of other nodes, commonly referred to as *top-level domains (TLDs)*. These are the familiar .com, .net, .org, .gov, .edu, and other such names. Table 2.1 lists some of these TLDs.

TABLE 2.1 Common top-level DNS domains

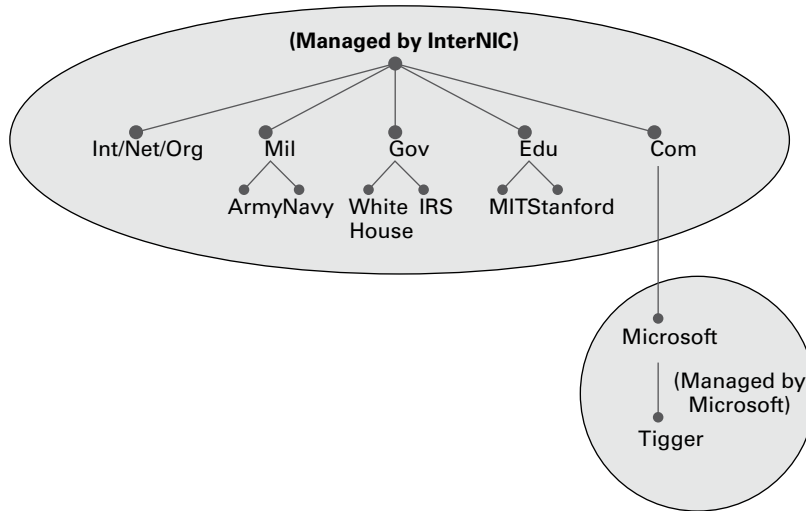
Common top-level domain names	Type of organization
com	Commercial (for example, stellacon.com for Stellacon Training Corporation).
edu	Educational (for example, gatech.edu for the Georgia Institute of Technology)
gov	Government (for example, whitehouse.gov for the White House in Washington, D.C.)
int	International organizations (for example, nato.int for NATO); this top-level domain is fairly rare
mil	Military organizations (for example, usmc.mil for the Marine Corps); there is a separate set of root name servers for this domain
net	Networking organizations and Internet providers (for example, hiwaay.net for HiWAAY Information Systems); many commercial organizations have registered names under this domain too
org	Noncommercial organizations (for example, fidonet.org for FidoNet)
au	Australia
uk	United Kingdom
ca	Canada
us	United States
jp	Japan

Each of these nodes then branches out into another set of domains, and they combine to form what we refer to as domain names, such as microsoft.com. A *domain name* identifies the domain's position in the logical DNS hierarchy in relation to its parent domain by separating each branch of the tree with a dot. Figure 2.2 shows a few of the top-level

domains, where the Microsoft domain fits, and a host called Tigger within the microsoft.com domain. If someone wanted to contact that host, they would use the *fully qualified domain name (FQDN)*, tigger.microsoft.com.

An FQDN includes the trailing dot (.) to indicate the root node, but it's commonly left off in practice.

FIGURE 2.2 The DNS hierarchy



As previously stated, one of the strengths of DNS is the ability to delegate control over portions of the DNS namespace to multiple organizations. For example, the Internet Corporation for Assigned Names and Numbers (ICANN) assigns the control over TLDs to one or more organizations. In turn, those organizations delegate portions of the DNS namespace to other organizations. For example, when you register a domain name, let's call it example.com, you control the DNS for the portion of the DNS namespace within example.com. The registrar controlling the .com TLD has delegated control over the example.com node in the DNS tree. No other node can be named example directly below the .com within the DNS database.

Within the portion of the domain namespace that you control (example.com), you could create host and other records (more on these later). You could also further subdivide example.com and delegate control over those divisions to other organizations or departments. These divisions are called *subdomains*. For example, you might create subdomains named for the cities in which the company has branch offices and then delegate control over those subdomains to the branch offices. The subdomains might be named losangeles.example.com, chicago.example.com, portsmouth.example.com, and so on.

Each domain (or delegated subdomain) is associated with DNS name servers. In other words, for every node in the DNS, one or more servers can give an authoritative answer to

queries about that domain. At the root of the domain namespace are the root servers. More on these later.



Domain names and hostnames must contain only characters a to z, A to Z, 0 to 9, and - (hyphen). Other common and useful characters, such as the & (ampersand), / (slash), . (period), and _ (underscore), are not allowed. This is in conflict with NetBIOS's naming restrictions. However, you'll find that Windows Server 2012 R2 is smart enough to take a NetBIOS name, like `Server_1`, and turn it into a legal DNS name, like `server1.example.com`.

DNS servers work together to resolve hierarchical names. If a server already has information about a name, it simply fulfills the query for the client. Otherwise, it queries other DNS servers for the appropriate information. The system works well because it distributes the authority over separate parts of the DNS structure to specific servers. A *DNS zone* is a portion of the DNS namespace over which a specific DNS server has authority (DNS zone types are discussed in detail later in this chapter).



There is an important distinction to make between DNS zones and Active Directory (AD) domains. Although both use hierarchical names and require name resolution, DNS zones do not map directly to AD domains.

Within a given DNS zone, resource records (RRs) contain the hosts and other database information that make up the data for the zone. For example, an RR might contain the host entry for `www.example.com`, pointing it to the IP address `192.168.1.10`.

Understanding Servers, Clients, and Resolvers

You will need to know a few terms and concepts in order to manage a DNS server. Understanding these terms will make it easier to understand how the Windows Server 2012 R2 DNS server works.

DNS Server Any computer providing domain name services is a *DNS name server*. No matter where the server resides in the DNS namespace, it's still a DNS name server. For example, 13 root name servers at the top of the DNS tree are responsible for delegating the TLDs. The *root servers* provide referrals to name servers for the TLDs, which in turn provide referrals to an authoritative name server for a given domain.



The Berkeley Internet Name Domain (BIND) was originally the only software available for running the root servers on the Internet. However, a few years ago the organizations responsible for the root servers undertook an effort to diversify the software running on these important machines. Today, root servers run multiple types of name server software. BIND is still primarily on Unix-based machines, and it is also the most popular for Internet providers. No root servers run Windows DNS.

Any DNS server implementation supporting Service Location Resource Records (see RFC 2782) and Dynamic Updates (RFC 2136) is sufficient to provide the name service for any operating system running Windows 2003 software and newer.

DNS Client A *DNS client* is any machine that issues queries to a DNS server. The client hostname may or may not be registered in a DNS database. Clients issue DNS requests through processes called *resolvers*. You'll sometimes see the terms *client* and *resolver* used synonymously.

Resolver *Resolvers* are software processes, sometimes implemented in software libraries that handle the actual process of finding the answers to queries for DNS data. The resolver is also built into many larger pieces of software so that external libraries don't have to be called to make and process DNS queries. Resolvers can be what you'd consider client computers or other DNS servers attempting to resolve an answer on behalf of a client (for example, Internet Explorer).

Query A *query* is a request for information sent to a DNS server. Three types of queries can be made to a DNS server: recursive, inverse, and iterative. I'll discuss the differences between these query types in the section "DNS Queries," a bit later in the chapter.

Understanding the DNS Process

To help you understand the DNS process, I will start by covering the differences between Dynamic DNS and Non-Dynamic DNS. During this discussion, you will learn how Dynamic DNS populates the DNS database. You'll also see how to implement security for Dynamic DNS. I will then talk about the workings of different types of DNS queries. Finally, I will discuss caching and time to live (TTL). You'll learn how to determine the best setting for your organization.

Dynamic DNS and Non-Dynamic DNS

To understand Dynamic DNS and Non-Dynamic DNS, you must go back in time. (Here is where the TV screen always used to get wavy.) Many years ago when many of us worked on Windows NT 3.51 and Windows NT 4.0, almost all Microsoft networks used Windows Internet Name Service (WINS) to do their TCP/IP name resolution. Windows versions 95/98 and NT 4.0 Professional were all built on the idea of using WINS. This worked out well for administrators because WINS was dynamic (which meant that once it was installed, it automatically built its own database). Back then, there was no such thing as Dynamic DNS; administrators had to enter DNS records into the server manually. This is important to know even today. If you have clients still running any of these older operating systems (95/98 or NT 4), these clients cannot use Dynamic DNS.

Now let's move forward in time to the release of Windows Server 2000. Microsoft announced that DNS was going to be the name resolution method of choice. Many administrators (myself included) did not look forward to the switch. Because there was no such thing as Dynamic DNS, most administrators had nightmares about manually entering

records. However, luckily for us, when Microsoft released Windows Server 2000, DNS had the ability to operate dynamically. Now when you're setting up Windows Server 2012 R2 DNS, you can choose what type of dynamic update you would like to use, if any. Let's talk about why you would want to choose one over the other.

The *Dynamic DNS (DDNS) standard*, described in RFC 2136, allows DNS clients to update information in the DNS database files. For example, a Windows Server 2012 R2 DHCP server can automatically tell a DDNS server which IP addresses it has assigned to what machines. Windows 2000, 2003, 2008, XP Pro, Vista, Windows 7, and Windows 8 DHCP clients can do this too. For security reasons, however, it's better to let the DHCP server do it. The result: IP addresses and DNS records stay in sync so that you can use DNS and DHCP together seamlessly. Because DDNS is a proposed Internet standard, you can even use the Windows Server 2012 R2 DDNS-aware parts with Unix/Linux-based DNS servers.

Non-Dynamic DNS (NDDNS) does not automatically populate the DNS database. The client systems do not have the ability to update to DNS. If you decide to use Non-Dynamic DNS, an administrator will need to populate the DNS database manually. Non-Dynamic DNS is a reasonable choice if your organization is small to midsized and you do not want extra network traffic (clients updating to the DNS server) or if you need to enter the computer's TCP/IP information manually because of strict security measures.



Dynamic DNS has the ability to be secure, and the chances are slim that a rogue system (a computer that does not belong in your DNS database) could update to a secure DNS server. Nevertheless, some organizations have to follow stricter security measures and are not allowed to have dynamic updates.

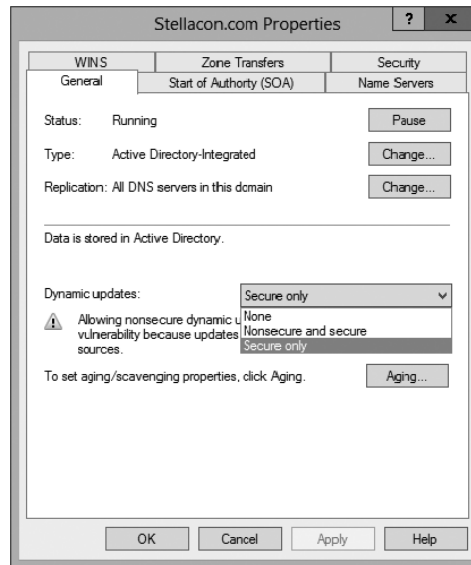
The major downside to entering records into DNS manually occurs when the organization is using the *Dynamic Host Configuration Protocol (DHCP)*. When using DHCP, it is possible for users to end up with different TCP/IP addresses every day. This means an administrator has to update DNS manually each day to keep it accurate.

If you choose to allow Dynamic DNS, you need to decide how you want to set it up. When setting up dynamic updates on your DNS server, you have three choices (see Figure 2.3).

None This means your DNS server is Non-Dynamic.

Nonsecure and Secure This means that any machine (even if it does not have a domain account) can register with DNS. Using this setting could allow rogue systems to enter records into your DNS server.

Secure Only This means that only machines with accounts in Active Directory can register with DNS. Before DNS registers any account in its database, it checks Active Directory to make sure that account is an authorized domain computer.

FIGURE 2.3 Setting the Dynamic Updates option

How Dynamic DNS Populates the DNS Database

TCP/IP is the protocol used for network communications on a Microsoft Windows Server 2012 R2 network. Users have two ways to receive a TCP/IP number:

- Static (administrators manually enter the TCP/IP information)
- Dynamic (using DHCP)

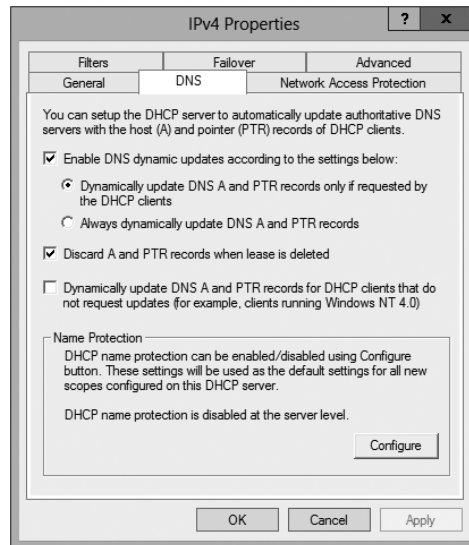
When an administrator sets up TCP/IP, DNS can also be configured.

Once a client gets the address of the DNS server, if that client is allowed to update with DNS, the client sends a registration to DNS or requests DHCP to send the registration. DNS then does one of two things, depending on which Dynamic Updates option is specified:

- Check with Active Directory to see whether that computer has an account (Secure Only updates), and if it does, enter the record into the database.
- Enter the record into its database (Nonsecure and Secure updates).

What if you have clients that cannot update DNS? Well, there is a solution—DHCP. In the DNS tab of the IPv4 Properties window, check the option labeled “Dynamically update DNS A and PTR records for DHCP clients that do not request updates (for example, clients running Windows NT 4.0),” which is shown in Figure 2.4.

DHCP, along with Dynamic DNS clients, allows an organization to update its DNS database dynamically without the time and effort of having an administrator manually enter DNS records.

FIGURE 2.4 DHCP settings for DNS

DNS Queries

As stated earlier, a client can make three types of queries to a DNS server: recursive, inverse, and iterative. Remember that the client of a DNS server can be a resolver (what you'd normally call a client) or another DNS server.

Iterative Queries

Iterative queries are the easiest to understand: A client asks the DNS server for an answer, and the server returns the best answer. This information likely comes from the server's cache. The server never sends out an additional query in response to an iterative query. If the server doesn't know the answer, it may direct the client to another server through a referral.

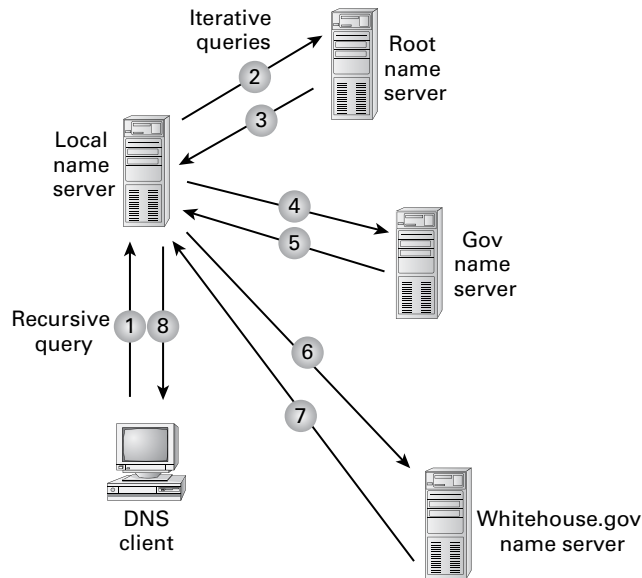
Recursive Queries

In a *recursive query*, the client sends a query to a name server, asking it to respond either with the requested answer or with an error message. The error states one of two things:

- The server can't come up with the right answer.
- The domain name doesn't exist.

In a recursive query, the name server isn't allowed just to refer the client to some other name server. Most resolvers use recursive queries. In addition, if your DNS server uses a forwarder, the requests sent by your server to the forwarder will be recursive queries.

Figure 2.5 shows an example of both recursive and iterative queries. In this example, a client within the Microsoft Corporation is querying its DNS server for the IP address for `www.whitehouse.gov`.

FIGURE 2.5 A sample DNS query

Here's what happens to resolve the request:

1. The resolver sends a recursive DNS query to its local DNS server asking for the IP address of `www.whitehouse.gov`. The local name server is responsible for resolving the name, and it cannot refer the resolver to another name server.
2. The local name server checks its zones, and it finds no zones corresponding to the requested domain name.
3. The root name server has authority for the root domain, and it will reply with the IP address of a name server for the `.gov` top-level domain.
4. The local name server sends an iterative query for `www.whitehouse.gov` to the Gov name server.
5. The Gov name server replies with the IP address of the name server servicing the `whitehouse.gov` domain.
6. The local name server sends an iterative query for `www.whitehouse.gov` to the `whitehouse.gov` name server.
7. The `whitehouse.gov` name server replies with the IP address corresponding to `www.whitehouse.gov`.
8. The local name server sends the IP address of `www.whitehouse.gov` back to the original resolver.

Inverse Queries

Inverse queries use pointer (PTR) records. Instead of supplying a name and then asking for an IP address, the client first provides the IP address and then asks for the name. Because there's no direct correlation in the DNS namespace between a domain name and its associated IP address, this search would be fruitless without the use of the `in-addr.arpa` domain. Nodes in the `in-addr.arpa` domain are named after the numbers in the dotted-octet representation of IP addresses. However, because IP addresses get more specific from left to right and domain names get less specific from left to right, the order of IP address octets must be reversed when building the `in-addr.arpa` tree. With this arrangement, administration of the lower limbs of the DNS `in-addr.arpa` tree can be given to companies as they are assigned their Class A, B, or C subnet address or delegated even further down thanks to Variable Length Subnet Masking (VLSM).

Once the domain tree is built into the DNS database, a special PTR record is added to associate the IP addresses with the corresponding hostnames. In other words, to find a hostname for the IP address 206.131.234.1, the resolver would query the DNS server for a PTR record for `1.234.131.206.in-addr.arpa`. If this IP address is outside of the local domain, the DNS server will start at the root and sequentially resolve the domain nodes until arriving at `234.131.206.in-addr.arpa`, which would contain the PTR record for the desired host.

Caching and Time to Live

When a name server is processing a recursive query, it may be required to send out several queries to find the definitive answer. Name servers, acting as resolvers, are allowed to cache all of the received information during this process; each record contains information called *time to live (TTL)*. The TTL specifies how long the record will be held in the local cache until it must be resolved again. If a query comes in that can be satisfied by this cached data, the TTL that's returned with it equals the current amount of time left before the data is flushed.

There is also a negative cache TTL. The *negative cache TTL* is used when an authoritative server responds to a query indicating that the record queried doesn't exist, and it indicates the amount of time that this negative answer may be held. Negative caching is quite helpful in preventing repeated queries for names that don't exist.

The administrator for the DNS zone sets TTL values for the entire zone. The value can be the same across the zone, or the administrator can set a separate TTL for each RR within the zone. Client resolvers also have data caches and honor the TTL value so that they know when to flush.

Choosing Appropriate TTL Values

For zones that you administer, you can choose the TTL values for the entire zone, for negative caching, and for individual records. Choosing an appropriate TTL depends on a number of factors, including the following:

- Amount of change you anticipate for the records within the zone
- Amount of time you can withstand an outage that might require changing an IP address
- Amount of traffic you believe the DNS server can handle

Resolvers query the name server every time the TTL expires for a given record. A low TTL, say 60 seconds, can burden the name server, especially for popular DNS records. (DNS queries aren't particularly intensive for a server to handle, but they can add up quickly if you mistakenly use 60 seconds instead of 600 seconds for the TTL on a popular record.) Set a low TTL only when you need to respond quickly to a changing environment.

A high TTL, say 604,800 seconds (that's one week), means that if you need to make a change to the DNS record, clients might not see the change for up to a week. This consideration is especially important when making changes to the network, and it's one that's all too frequently overlooked. I can't count the number of times I've worked with clients who had recently made a DNS change to a new IP for their email or website only to ask why it's not working for some clients. The answer can be found in the TTL value. If the record is being cached, then the only thing that can solve their problem is time.

You should choose a TTL that's appropriate for your environment. Take the following factors into account:

- The amount of time that you can afford to be offline if you need to make a change to a DNS record that's being cached
- The amount of load that a low TTL will cause on the DNS server

In addition, you should plan well ahead of any major infrastructure changes and change the TTL to a lower value to lessen the effect of the downtime by reducing the amount of time that the record(s) can be cached.

Introducing DNS Database Zones

As mentioned earlier in this chapter, a DNS zone is a portion of the DNS namespace over which a specific DNS server has authority. Within a given DNS zone, there are resource records (RRs) that define the hosts and other types of information that make up the database for the zone. You can choose from several different zone types. Understanding the characteristics of each will help you choose which is right for your organization.



The DNS zones discussed in this book are all Microsoft Windows Server 2012 / 2012 R2 zones. Non-Windows (for example, Unix) systems set up their DNS zones differently.

In the following sections, I will discuss the different zone types and their characteristics.

Understanding Primary Zones

When you're learning about zone types, things can get a bit confusing. But it's really not difficult to understand how they work and why you would want to choose one type of zone over the other. Zones are databases that store records. By choosing one zone type over another, you are basically just choosing how the database works and how it will be stored on the server.

The primary zone is responsible for maintaining all of the records for the DNS zone. It contains the primary copy of the DNS database. All record updates occur on the primary zone. You will want to create and add primary zones whenever you create a new DNS domain.

There are two types of primary zones:

- Primary zone
- Primary zone with Active Directory Integration (Active Directory DNS)



From this point forward, I refer to a primary zone with Active Directory Integration as an *Active Directory DNS*. When I use only the term *primary zone*, Active Directory is not included.

To install DNS as a primary zone, you must first install DNS using the Server Manager MMC. Once DNS is installed and running, you create a new zone and specify it as a primary zone.



The process of installing DNS and its zones will be discussed later in this chapter. In addition, there will be step-by-step exercises to walk you through how to install these components.

Primary zones have advantages and disadvantages. Knowing the characteristics of a primary zone will help you decide when you need the zone and when it fits into your organization.

Local Database

Primary DNS zones get stored locally in a file (with the suffix `.dns`) on the server. This allows you to store a primary zone on a domain controller or a member server. In addition, by loading DNS onto a member server, you can help a small organization conserve resources. Such an organization may not have the resources to load DNS on an Active Directory domain controller.

Unfortunately, the local database has many disadvantages:

Lack of Fault Tolerance Think of a primary zone as a contact list on your smartphone. All of the contacts in the list are the records in your database. The problem is that, if you lose your phone or the phone breaks, you lose your contact list. Until your phone gets fixed or you swap out your phone card, the contacts are unavailable.

It works the same way with a primary zone. If the server goes down or you lose the hard drive, DNS records on that machine are unreachable. An administrator can install a secondary zone (explained later in the next section), and that provides temporary fault tolerance. Unfortunately, if the primary zone is down for an extended period of time, the secondary server's information will no longer be valid.

Additional Network Traffic Let's imagine that you are looking for a contact number for John Smith. John Smith is not listed in your smartphone directory, but he is listed in your partner's smartphone. You have to contact your partner to get the listing. You cannot directly access your partner's phone's contacts.

When a resolver sends a request to DNS to get the TCP/IP address for Jsmith (in this case Jsmith is a computer name) and the DNS server does not have an answer, it does not have the ability to check the other server's database directly to get an answer. Thus it forwards the request to another DNS. When DNS servers are replicating zone databases with other DNS servers, this causes additional network traffic.

No Security Staying with the smartphone example, let's say that you call your partner looking for John Smith's phone number. When your partner gives you the phone number over your wireless phone, someone with a scanner can pick up your conversation. Unfortunately, wireless telephone calls are not very secure.

Now a resolver asks a primary zone for the Jsmith TCP/IP address. If someone on the network has a packet sniffer, they can steal the information in the DNS packets being sent over the network. The packets are not secure unless you implement some form of secondary security. Also, the DNS server has the ability to be dynamic. A primary zone accepts all updates from DNS servers. You cannot set it to accept secure updates only.

Understanding Secondary Zones

In Windows Server 2012 R2 DNS, you have the ability to use secondary DNS zones. Secondary zones are noneditable copies of the DNS database. You use them for *load balancing* (also referred to as *load sharing*), which is a way of managing network overloads on a single server. A secondary zone gets its database from a primary zone.

A *secondary zone* contains a database with all of the same information as the primary zone, and it can be used to resolve DNS requests. Secondary zones have the following advantages:

- A secondary zone provides fault tolerance, so if the primary zone server becomes unavailable, name resolution can still occur using the secondary zone server.
- Secondary DNS servers can also increase network performance by offloading some of the traffic that would otherwise go to the primary server.

Secondary servers are often placed within the parts of an organization that have high-speed network access. This prevents DNS queries from having to run across slow wide area network (WAN) connections. For example, if there are two remote offices within the stellacon.com organization, you may want to place a secondary DNS server in each remote office. This way, when clients require name resolution, they will contact the nearest server for this IP address information, thus preventing unnecessary WAN traffic.

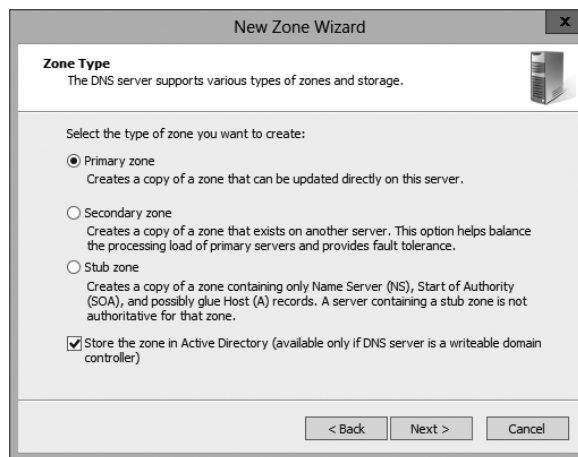


Having too many secondary zone servers can actually cause an increase in network traffic because of replication (especially if DNS changes are fairly frequent). Therefore, you should always weigh the benefits and drawbacks and properly plan for secondary zone servers.

Understanding Active Directory Integrated DNS

Windows Server 2000 introduced *Active Directory Integrated DNS* to the world. This zone type was unique and was a separate choice during setup. In Windows Server 2003, this zone type became an add-on to a primary zone. In Windows Server 2012 R2, it works the same way. After choosing to set up a primary zone, you check the box labeled Store The Zone In Active Directory (see Figure 2.6).

FIGURE 2.6 Setting up an Active Directory Integrated zone



Disadvantages of Active Directory Integrated DNS

The main disadvantage of Active Directory Integrated DNS is that it has to reside on a domain controller because the DNS database is stored in Active Directory. As a result, you cannot load this zone type on a member server, and small organizations might not have the resources to set up a dedicated domain controller.

Advantages of Active Directory Integrated DNS

The advantages of using an Active Directory Integrated DNS zone well outweigh the disadvantage just discussed. The following are some of the major advantages to an Active Directory Integrated zone:

Full Fault Tolerance Think of an Active Directory Integrated zone as a database on your server that stores contact information for all your clients. If you need to retrieve John Smith's phone number, as long as it was entered, you can look it up on the software.

If John Smith's phone number was stored only on your computer and your computer stopped working, no one could access John Smith's phone number. But since John Smith's phone number is stored in a database to which everyone has access, if your computer stops working, other users can still retrieve John Smith's phone number.

An Active Directory Integrated zone works the same way. Since the DNS database is stored in Active Directory, all Active Directory DNS servers can have access to the same data. If one server goes down or you lose a hard drive, all other Active Directory DNS servers can still retrieve DNS records.

No Additional Network Traffic As previously discussed, an Active Directory Integrated zone is stored in Active Directory. Since all records are now stored in Active Directory, when a resolver needs a TCP/IP address for Jsmith, any Active Directory DNS server can access Jsmith's address and respond to the resolver.

When you choose an Active Directory Integrated zone, DNS zone data can be replicated automatically to other DNS servers during the normal Active Directory replication process.

DNS Security An Active Directory Integrated zone has a few security advantages over a primary zone:

- An Active Directory Integrated zone can use secure dynamic updates.
- As explained earlier, the Dynamic DNS standard allows secure-only updates or dynamic updates, not both.
- If you choose secure updates, then only machines with accounts in Active Directory can register with DNS. Before DNS registers any account in its database, it checks Active Directory to make sure that it is an authorized domain computer.
- An Active Directory Integrated zone stores and replicates its database through Active Directory replication. Because of this, the data gets encrypted as it is sent from one DNS server to another.

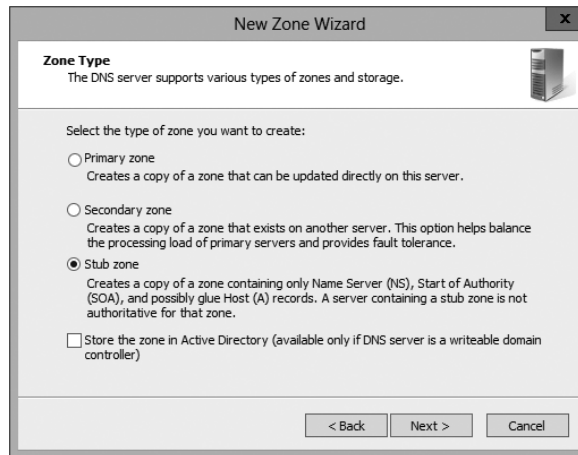
Background Zone Loading Background zone loading (discussed in more detail later in this chapter) allows an Active Directory Integrated DNS zone to load in the background. As a result, a DNS server can service client requests while the zone is still loading into memory.

Understanding Stub Zones

Stub zones work a lot like secondary zones—the database is a noneditable copy of a primary zone. The difference is that the stub zone's database contains only the information

necessary (three record types) to identify the authoritative DNS servers for a zone (see Figure 2.7). You should not use stub zones to replace secondary zones, nor should you use them for redundancy and load balancing.

FIGURE 2.7 DNS stub zone type



Stub zone databases contain only three record types: name server (NS), start of authority (SOA), and glue host (A) records. Understanding these records will help you on the Microsoft certification exams. Microsoft asks many questions about stub zones on all DNS-related exams.

When to Use Stub Zones

Stub zones become particularly useful in a couple of different scenarios. Consider what happens when two large companies merge: `example.com` and `example.net`. In most cases, the DNS zone information from both companies must be available to every employee. You could set up a new zone on each side that acts as a secondary for the other side's primary zone, but administrators tend to be very protective of their DNS databases, and they probably wouldn't agree to this plan.

A better solution is to add to each side a stub zone that points to the primary server on the other side. When a client in `example.com` (which you help administer) makes a request for a name in `example.net`, the stub zone on the `example.com` DNS server would send the client to the primary DNS server for `example.net` without actually resolving the name. At this point, it would be up to `example.net`'s primary server to resolve the name.

An added benefit is that, even if the administrators over at `example.net` change their configuration, you won't have to do anything because the changes will automatically replicate to the stub zone, just as they would for a secondary server.

Stub zones can also be useful when you administer two domains across a slow connection. Let's change the previous example a bit and assume that you have full control over `example.com` and `example.net` but that they connect through a 56Kbps line. In this case, you wouldn't necessarily mind using secondary zones because you personally administer the entire network. However, it could get messy to replicate an entire zone file across that slow line. Instead, stub zones would refer clients to the appropriate primary server at the other site.

GlobalName Zones

Earlier in this chapter, I talked about organizations using WINS to resolve NetBIOS names (also referred to as *computer names*) to TCP/IP addresses. Even today, many organizations still use WINS along with DNS for name resolution. Unfortunately, WINS is slowly becoming obsolete.

To help organizations move forward with an all-DNS network, Microsoft Windows Server 2012 R2 DNS supports *GlobalName zones*. These use single-label names (DNS names that do not contain a suffix such as `.com`, `.net`, and so on). GlobalName zones are not intended to support peer-to-peer networks and workstation name resolution, and they don't support dynamic DNS updates.

GlobalName zones are designed to be used with servers. Because GlobalName zones are not dynamic, an administrator has to enter the records into the zone database manually. In most organizations, the servers have static TCP/IP addresses, and this works well with the GlobalName zone design. GlobalName zones are usually used to map single-label CNAME (alias) resource records to an FQDN.

Zone Transfers and Replication

DNS is such an important part of the network that you should not just use a single DNS server. With a single DNS server, you also have a single point of failure, and in fact, many domain registrars encourage the use of more than two name servers for a domain. Secondary servers or multiple primary Active Directory Integrated servers play an integral role in providing DNS information for an entire domain.

As previously stated, secondary DNS servers receive their zone databases through zone transfers. When you configure a secondary server for the first time, you must specify the primary server that is authoritative for the zone and that will send the zone transfer. The primary server must also permit the secondary server to request the zone transfer.

Zone transfers occur in one of two ways: *full zone transfers (AXFR)* and *incremental zone transfers (IXFR)*.

When a new secondary server is configured for the first time, it receives a full zone transfer from the primary DNS server. The full zone transfer contains all of the information in the DNS database. Some DNS implementations always receive full zone transfers.

After the secondary server receives its first full zone transfer, subsequent zone transfers are incremental. The primary name server compares its zone version number with that of the secondary server, and it sends only the changes that have been made in the interim. This significantly reduces network traffic generated by zone transfers.

The secondary server typically initiates zone transfers when the refresh interval time for the zone expires or when the secondary or stub server boots. Alternatively, you can configure notify lists on the primary server that send a message to the secondary or stub servers whenever any changes to the zone database occur.

When you consider your DNS strategy, you must carefully consider the layout of your network. If you have a single domain with offices in separate cities, you want to reduce the number of zone transfers across the potentially slow or expensive WAN links, although this is becoming less of a concern because of continuous increases in bandwidth.

Active Directory Integrated zones do away with traditional zone transfers altogether. Instead, they replicate across Active Directory with all of the other AD information. This replication is secure and encrypted because it uses the Active Directory security.

How DNS Notify Works

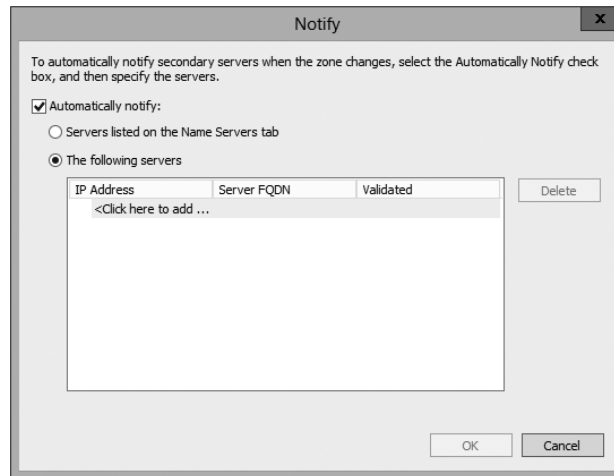
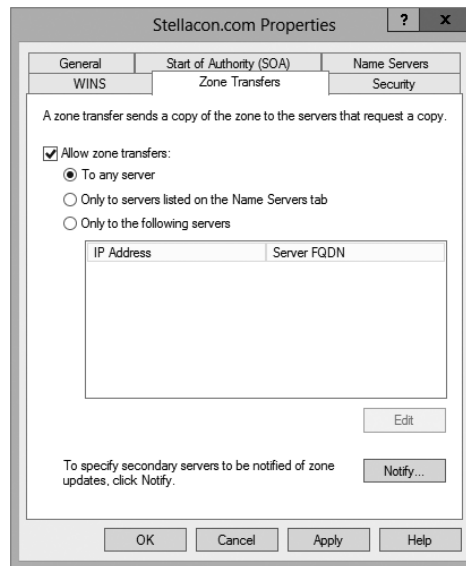
Windows Server 2012 R2 supports DNS Notify. *DNS Notify* is a mechanism that allows the process of initiating notifications to secondary servers when zone changes occur (RFC 1996). DNS Notify uses a push mechanism for communicating to a select set of secondary zone servers when their zone information is updated. (DNS Notify does not allow you to configure a notify list for a stub zone.)

After being notified of the changes, secondary servers can then start a pull zone transfer and update their local copies of the database.



Many different mechanisms use the push/pull relationship. Normally, one object pushes information to another, and the second object pulls the information from the first. Most applications push replication on a change value and pull it on a time value. For example, a system can push replication after 10 updates, or it can be pulled every 30 minutes.

To configure the DNS Notify process, you create a list of secondary servers to notify. List the IP address of the server in the primary master's Notify dialog box (see Figure 2.8). The Notify dialog box is located under the Zone Transfers tab, which is located in the zone Properties dialog box (see Figure 2.9).

FIGURE 2.8 DNS Notify dialog box**FIGURE 2.9** DNS Zone Transfers tab

Configuring Stub Zone Transfers with Zone Replication

In the preceding section, I talked about how to configure secondary server zone transfers. What if you wanted to configure settings for stub zone transfers? This is where zone replication scope comes in.

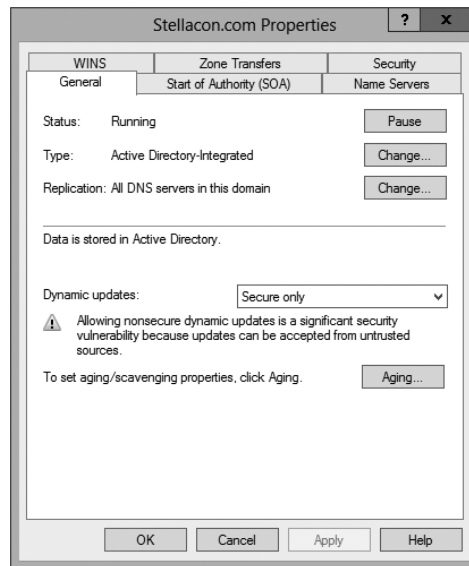
Only Active Directory–Integrated primary and stub zones can configure their replication scope. Secondary servers do not have this ability.

You can configure zone replication scope configurations in two ways. An administrator can set configuration options through the DNS snap-in or through a command-line tool called DNSCmd.

To configure zone replication scope through the DNS snap-in, follow these steps:

1. Click Start > Administrative Tools > DNS.
2. Right-click the zone you want to set up.
3. Choose Properties.
4. In the Properties dialog box, click the Change button next to Replication (see Figure 2.10).

FIGURE 2.10 DNS zone replication scope



5. Choose the replication scope that fits your organization.

Advantages of DNS in Windows Server 2012 R2

DNS in Microsoft Windows Server 2012 R2 has some great advantages over many other versions of Microsoft DNS. Here are some of the improvements of DNS in Windows Server 2012 R2 (some of these became available in Windows Server 2008, but they have been improved upon):

- Background zone loading
- Support for TCP/IP version 6 (IPv6)
- Read-only domain controllers
- GlobalName zone
- DNS Socket Pool
- DNS Cache Locking
- DNS Security Extensions (DNSSEC)
- DNS Devolution
- Zone Level Statistics
- Record Weighting
- Netmask Ordering
- DnsUpdateProxy Group
- Windows PowerShell Support

Background Zone Loading

If an organization had to restart a DNS server with an extremely large Active Directory Integrated DNS zones database in the past, DNS had a common problem with an Active Directory Integrated DNS zone. After the DNS restart, it could take hours for DNS data to be retrieved from Active Directory. During this time, the DNS server was unable to service any client requests.

Microsoft Windows Server 2008 DNS addressed this problem by implementing background zone loading, and Windows Server 2012 R2 has taken it a step further. As the DNS restarts, the Active Directory zone data populates the database in the background. This allows the DNS server to service client requests for data from other zones almost immediately after a restart.

Background zone loading accomplishes this task by loading the DNS zone using separate threads. This allows a DNS server to service requests while still loading the rest of the zone. If a client sends a request to the DNS server for a computer that has not yet loaded into memory, the DNS server retrieves the data from Active Directory and updates the record.

Support for IPv6 Addresses

Over the past few years, the Internet has starting running into a problem that was not foreseen when it was first created—it started running out of TCP/IP addresses. As you probably know, when the Internet was created, it was used for government and academic purposes only. Then, seemingly overnight, it grew to be the information superhighway. Nowadays, asking someone for his or her email address is almost as common as asking for their phone number.

Version 4 (IPv4) was the common version of TCP/IP. The release of TCP/IP version 6 (IPv6) has solved the lack-of-IP-addresses problem. IPv4 addresses are 32 bits long, but IPv6 addresses are 128 bits in length. The longer lengths allow for a much greater number of globally unique TCP/IP addresses.

Microsoft Windows Server 2012 R2 DNS has built-in support to accommodate both IPv4 and IPv6 address records. (DNS records are explained later in this chapter.) DHCP can also issue IPv6 addresses, which lets administrators allow DHCP to register the client with DNS, or the IPv6 client can register their address with the DNS server.

Support for Read-Only Domain Controllers

Windows Server 2008 introduced a new type of domain controller called the *read-only domain controller (RODC)*. This is a full copy of the Active Directory database without the ability to write to Active Directory. The RODC gives an organization the ability to install a domain controller in a location (onsite or offsite) where security is a concern.

Microsoft Windows Server 2012 R2 DNS has implemented a type of zone to help support an RODC. A primary read-only zone allows a DNS server to receive a copy of the application partition (including ForestDNSZones and DomainDNSZones) that DNS uses. This allows DNS to support an RODC because DNS now has a full copy of all DNS zones stored in Active Directory.

A primary, read-only zone is just what it says—a read-only zone; so to make any changes to it, you have to change the primary zones located on the Active Directory Integrated DNS server.

DNS Socket Pools

If your server is running Windows Server 2012 R2, you will be able to take advantage of DNS socket pools. *DNS socket pools* allow source port randomization to protect against DNS cache-poisoning attacks.

If you choose to use source port randomization, when the DNS service starts, the DNS server will randomly pick a source port from a pool of available sockets. This is an advantage because, instead of DNS using a well-known source port when issuing queries, the DNS server uses a random port selected from the socket pool. This helps guard against attacks because a hacker must correctly access the source port of the DNS query. The socket pool is automatically enabled in DNS with the default settings.

When using the DNS Socket Pool, the default size of the DNS socket pool is 2,500. When configuring the socket pool, you have the ability to choose a size value from 0 to 10,000. The larger the value, the greater the protection you will have against DNS spoofing attacks. If you decide to configure your socket pool size with a zero value, only a single socket for remote DNS queries will be used.

DNS Cache Locking

Windows Server 2012 R2 *DNS cache locking* allows cached DNS records to remain safe for the duration of the record's time to live (TTL) value. This means that the cached DNS records cannot be overwritten or changed. Because of this new DNS feature, it's tougher for hackers to perform cache-poisoning attacks against your DNS server.

DNS administrators can set how long a record will remain safe in cache. The configuration is based on a percent value. For example, if you set your cache locking value to 50 percent, then the cached records cannot be overwritten until half of the TTL has been reached. DNS cache locking is set to 100 percent by default. This means that the cached records never get overwritten.

DNS Security Extensions

One major issue that you must always look at is keeping your DNS safe. Think about it—DNS is a database of computer names and IP addresses. As a hacker, if I control DNS, I can control your company. In organizations that do not support extra security like IPsec, DNS security is even more important. This is where DNSSEC can help.

Windows Server 2012 R2 can use a suite of extensions that will help add security to DNS, and that suite is called *Domain Name System Security Extensions (DNSSEC)*, which was introduced in Windows Server 2008 R2. The DNSSEC protocol allows your DNS servers to be secure by validating DNS responses. DNSSEC secures your DNS resource records by accompanying the records with a digital signature.

To allow your DNS resource records to receive digital signatures, DNSSEC is applied to your DNS server by a procedure called *zone signing*. This process begins when a DNS resolver initiates a DNS query for a resource record in a signed DNS zone. When a response is returned, a digital signature (RRSIG) accompanies the response, and this allows the response to be verified. If the verification is successful, then the DNS resolver knows that the data has not been modified or tampered with in any way.

Once you implement a zone with DNSSEC, all of the records that are contained within that zone get individually signed. Since all of the records in the zone get individually signed, this gives administrators the ability to add, modify, or delete records without resigning the entire zone. The only requirement is to resign any updated records.

Trust Anchors

Trust anchors are an important part of the DNSSEC process because trust anchors allow the DNS servers to validate the DNSKEY resource records. *Trust anchors* are preconfigured public keys that are linked to a DNS zone. For a DNS server to perform validation, one or more trust anchors must be configured. If you are running an Active Directory Integrated zone, trust anchors can be stored in the Active Directory Domain Services directory partition of the forest. If you decide to store the trust anchors in the directory partition, then all DNS servers that reside on a domain controller get a copy of this trust anchor. On DNS servers that reside on stand-alone servers, trust anchors are stored in a file called `TrustAnchors.dns`.

If your servers are running Windows Server 2012 R2, then you can view trust anchors in the DNS Manager Console tree in the Trust Points container. You can also use Windows PowerShell or `Dnscmd.exe` to view trust anchors. Windows PowerShell is the recommended command-line method for viewing trust anchors. The following line is a PowerShell command to view the trust anchors for `Contoso.com`:

```
get-dnsservertrustanchor sec.contoso.com
```

DNSSEC Clients

Windows 7, Windows 8, Windows Server 2008/2008 R2, and Windows Server 2012/2012 R2 are all DNS clients that receive a response to a DNS query, examine the response, and then evaluate whether the response has been validated by a DNS server. The DNS client itself is nonvalidating, and the DNS client relies on the local DNS server to indicate that validation was successful. If the server doesn't perform validation, then the DNS client service can be configured to return no results.



If you are interested in learning how to set up DNSSEC in a lab environment, Microsoft has a website that explains all about DNSSEC and how to set up the lab environment. Visit <http://technet.microsoft.com/en-us/library/hh831411.aspx>.

DNS Devolution

Using *DNS devolution*, if a client computer is a member of a child namespace, the client computer will be able to access resources in the parent namespace without the need to provide explicitly the fully qualified domain name (FQDN) of the resource. DNS devolution removes the leftmost label of the namespace to get to the parent suffix. DNS devolution allows the DNS resolver to create the new FQDNs. DNS devolution works by appending the single-label, unqualified domain name with the parent suffix of the primary DNS suffix name.

Zone Level Statistics

DNS zone level server statistics are available in Windows Server 2012 R2 by using the Windows PowerShell cmdlet `Get-DnsServerStatistics`. This powerful Windows PowerShell cmdlet retrieves statistics and data for the DNS server. The following is an example of the `Get-DnsServerStatistics` cmdlet:

```
Get-DnsServerStatistics -ZoneName <String[]> [-AsJob] [-CimSession
<CimSession[]> ] [-Clear] [-ComputerName <String> ] [-ThrottleLimit <Int32> ]
[ <CommonParameters>]
```

The `ZoneName` parameter allows an administrator to get specific statistics for the DNS zone that you have specified. If the `ZoneName` parameter is not specified during the

PowerShell cmdlet, the server-level statistics are retrieved and shown. If an administrator uses the `Clear` parameter in the PowerShell cmdlet along with the `ZoneName` parameter, the statistics for the specified zone will be cleared. So, use this cmdlet carefully.

Record Weighting

Weighting DNS records will allow an administrator to place a value on DNS SRV records. Clients will then randomly choose SRV records proportional to the weight value assigned.

Netmask Ordering

If round robin is enabled, when a client requests name resolution, the first address entered in the database is returned to the resolver, and it is then sent to the end of the list. The next time a client attempts to resolve the name, the DNS server returns the second name in the database (which is now the first name) and then sends it to the end of the list, and so on. Round robin is enabled by default.

Netmask ordering is part of the round-robin process. When an administrator configures netmask ordering, the DNS server will detect the subnet of the querying client. The DNS server will then return a host address available for the same subnet. Netmask ordering is enabled through the DNS Manager console on the Advanced tab of the server Properties dialog box.

DnsUpdateProxy Group

As mentioned previously, the DHCP server can be configured to register host (A) and pointer (PTR) resource records dynamically on behalf of DHCP clients. Because of this, the DNS server can end up with stale resources. To help solve this issue, an administrator can use the built-in security group called *DnsUpdateProxy*.

To use the *DnsUpdateProxy* group, an administrator must first create a dedicated user account and configure the DHCP servers with its credentials. This will protect against the creation of unsecured records. Also, when you create the dedicated user account, members of the *DnsUpdateProxy* group will be able to register records in zones that allow only secured dynamic updates. Multiple DHCP servers can use the same credentials of one dedicated user account.

Now that you have looked at some of the new features of Windows Server 2012 R2 DNS, let's take a look at some of the DNS record types.

Windows PowerShell Support

Microsoft has more than 100 PowerShell cmdlets specifically for DNS. While in Windows PowerShell, you can list all of the DNS cmdlets that are available. To see the entire list, use the following PowerShell command:

```
Get-Command -Module DnsServer cmdlet.
```

For an entire list of all of the Windows PowerShell DNS cmdlets, go to <http://technet.microsoft.com/en-us/library/jj649850.aspx>.

Introducing DNS Record Types

No matter where your zone information is stored, you can rest assured that it contains a variety of DNS information. Although the DNS snap-in makes it unlikely that you'll ever need to edit these files by hand, it's good to know exactly what data is contained there.

As stated previously, zone files consist of a number of resource records. You need to know about several types of resource records to manage your DNS servers effectively. They are discussed in the following sections.

Part of the resource record is its class. *Classes* define the type of network for the resource record. There are three classes: Internet, Chaosnet, and Hesoid. By far, the Internet class is the most popular. In fact, it's doubtful that you'll see either Chaosnet or Hesoid classes in the wild.



The following are some of the more important resource records in a DNS database. For a listing of records in a Microsoft DNS database, visit Microsoft's website at <http://technet.microsoft.com/en-us/library/cc958958.aspx>.

Start of Authority Records

The first record in a database file is the *start of authority (SOA) record*. The SOA defines the general parameters for the DNS zone, including the identity of the authoritative server for the zone.

The SOA appears in the following format:

```
@ IN SOA primary_mastercontact_e-mailserial_number
refresh_timeretry_timeexpiration_timetime_to_live
```

Here is a sample SOA from the domain example.com:

```
@ IN SOA win2k3r2.example.com. hostmaster.example.com. (
    5                ; serial number
    900              ; refresh
    600              ; retry
    86400            ; expire
    3600             ) ; default TTL
```

Table 2.2 lists the attributes stored in the SOA record.

TABLE 2.2 The SOA record structure

Field	Meaning
Current zone	The current zone for the SOA. This can be represented by an @ symbol to indicate the current zone or by naming the zone itself. In the example, the current zone is <code>example.com</code> . The trailing dot (<code>.com.</code>) indicates the zone's place relative to the root of the DNS.
Class	This will almost always be the letters <code>IN</code> for the Internet class.
Type of record	The type of record follows. In this case, it's SOA.
Primary master	The primary master for the zone on which this file is maintained.
Contact email	The Internet email address for the person responsible for this domain's database file. There is no @ symbol in this contact email address because @ is a special character in zone files. The contact email address is separated by a single dot (<code>.</code>). So the email address of <code>root@example.com</code> would be represented by <code>root.example.com</code> in a zone file.
Serial number	This is the "version number" of this database file. It increases each time the database file is changed.
Refresh time	The amount of time (in seconds) that a secondary server will wait between checks to its master server to see if the database file has changed and a zone transfer should be requested.
Retry time	The amount of time (in seconds) that a secondary server will wait before retrying a failed zone transfer.
Expiration time	The amount of time (in seconds) that a secondary server will spend trying to download a zone. Once this time limit expires, the old zone information will be discarded.
Time to live	The amount of time (in seconds) that another DNS server is allowed to cache any resource records from this database file. This is the value that is sent out with all query responses from this zone file when the individual resource record doesn't contain an overriding value.

Name Server Records

Name server (NS) records list the name servers for a domain. This record allows other name servers to look up names in your domain. A zone file may contain more than one name server record. The format of these records is simple:

```
example.com.    IN      NS      Hostname.example.com
```

Table 2.3 explains the attributes stored in the NS record.

TABLE 2.3 The NS record structure

Field	Meaning
Name	The domain that will be serviced by this name server. In this case, I used <code>example.com</code> .
AddressClass	Internet (IN)
RecordType	Name server (NS)
Name Server Name	The FQDN of the server responsible for the domain



Any domain name in the database file that is not terminated with a period will have the root domain appended to the end. For example, an entry that just has the name `sales` will be expanded by adding the root domain to the end, whereas the entry `sales.example.com.` won't be expanded.

Host Record

A *host record* (also called an *A record* for IPv4 and *AAAA record* for IPv6) is used to associate statically a host's name to its IP addresses. The format is pretty simple:

```
host_nameoptional_TTL IN A IP_Address
```

Here's an example from my DNS database:

```
www IN A 192.168.0.204
SMTP IN A 192.168.3.144
```

The A or AAAA record ties a hostname (which is part of an FQDN) to a specific IP address. This makes these records suitable for use when you have devices with statically assigned IP addresses. In this case, you create these records manually using the DNS snap-in. As it turns out, if you enable DDNS, your DHCP server can create these for you. This automatic creation is what enables DDNS to work.

Notice that an optional TTL field is available for each resource record in the DNS. This value is used to set a TTL that is different from the default TTL for the domain. For example, if you wanted a 60-second TTL for the `www` A or AAAA record, it would look like this:

```
www 60 IN A 192.168.0.204
```


Alias Record

Closely related to the host record is the *alias record*, or *canonical name (CNAME) record*. The syntax of an alias record is as follows:

```
aliasoptional_TTL IN CNAME hostname
```

Aliases are used to point more than one DNS record toward a host for which an A record already exists. For example, if the hostname of your web server was actually chaos, you would likely have an A record such as this:

```
chaos IN A 192.168.1.10
```

Then you could make an alias or CNAME for the record so that `www.example.com` would point to chaos:

```
www IN CNAME chaos.example.com.
```

Note the trailing dot (.) on the end of the CNAME record. This means the root domain is not appended to the entry.

Pointer Record

A or AAAA records are probably the most visible component of the DNS database because Internet users depend on them to turn FQDNs like `www.microsoft.com` into the IP addresses that browsers and other components require to find Internet resources. However, the host record has a lesser-known but still important twin: the *pointer (PTR) record*. The format of a PTR record appears as follows:

```
reversed_address.in-addr.arpa. optional_TTL IN PTR targeted_domain_name
```

The A or AAAA record maps a hostname to an IP address, and the PTR record does just the opposite—mapping an IP address to a hostname through the use of the `in-addr.arpa` zone.

The PTR record is necessary because IP addresses begin with the least-specific portion first (the network) and end with the most-specific portion (the host), whereas hostnames begin with the most-specific portion at the beginning and the least-specific portion at the end.

Consider the example `192.168.1.10` with a subnet mask `255.255.255.0`. The portion `192.168.1` defines the network and the final `.10` defines the host, or the most-specific portion of the address. DNS is just the opposite: The hostname `www.example.com.` defines the most-specific portion, `www`, at the beginning and then traverses the DNS tree to the least-specific part, the dot (.), at the root of the tree.

Reverse DNS records, therefore, need to be represented in this most-specific-to-least-specific manner. The PTR record for mapping `192.168.1.10` to `www.example.com` would look like this:

```
10.1.168.192.in-addr.arpa. IN PTR www.example.com.
```

Now a DNS query for that record can follow the logical DNS hierarchy from the root of the DNS tree all the way to the most-specific portion.

Mail Exchanger Record

The *mail exchanger (MX) record* is used to specify which servers accept mail for this domain. Each MX record contains two parameters—a preference and a mail server, as shown in the following example:

```
domain IN MX preference mailserver_host
```

The MX record uses the preference value to specify which server should be used if more than one MX record is present. The preference value is a number. The lower the number, the more preferred the server. Here's an example:

```
example.com.    IN  MX  0  mail.example.com.
example.com.    IN  MX 10  backupmail.example.com.
```

In the example, `mail.example.com` is the default mail server for the domain. If that server goes down for any reason, emailers then use the `backupmail.example.com` mail server.

Service (SRV) Record

Windows Server 2012 R2 depends on some other services, like the Lightweight Directory Access Protocol (LDAP) and Kerberos. Using a service record, which is another type of DNS record, a Windows 2000, XP, Vista, Windows 7, Windows 8, and all Windows Server products can query DNS servers for the location of a domain controller. This makes it much easier (for both the client and the administrator) to manage and distribute logon traffic in large-scale networks. For this approach to work, Microsoft has to have some way to register the presence of a service in DNS. Enter the service (SRV) record.

Service (SRV) records tie together the location of a service (like a domain controller) with information about how to contact the service. SRV records provide seven items of information. Let's review an example to help clarify this powerful concept. (Table 2.4 explains the fields in the following example.)

```
ldap.tcp.example.com. 86400 IN SRV 10 100 389 hsv.example.com
ldap.tcp.example.com. 86400 IN SRV 20 100 389 msy.example.com
```

TABLE 2.4 The SRV record structure

Field	Meaning
Domain name	Domain for which this record is valid (<code>ldap.tcp.example.com.</code>).
TTL	Time to live (86,400 seconds).
Class	This field is always IN, which stands for Internet.

Record type	Type of record (SRV).
Priority	Specifies a preference, similar to the Preference field in an MX record. The SRV record with the lowest priority is used first (10).
Weight	Service records with equal priority are chosen according to their weight (100).
Port number	The port where the server is listening for this service (389).
Target	The FQDN of the host computer (hsv.example.com and msy.example.com).



You can define other types of service records. If your applications support them, they can query DNS to find the services they need.

Configuring DNS

In the following sections, you'll begin to learn about the actual DNS server. You will start by installing DNS. Then I will talk about different zone configuration options and what they mean. Finally, you'll complete an exercise that covers configuring Dynamic DNS, delegating zones, and manually entering records.

DNS requires that you use a static IP address on the server. The reason for this is that clients access the DNS server by using its TCP/IP address, and that's why DNS servers need to install a static IP address.

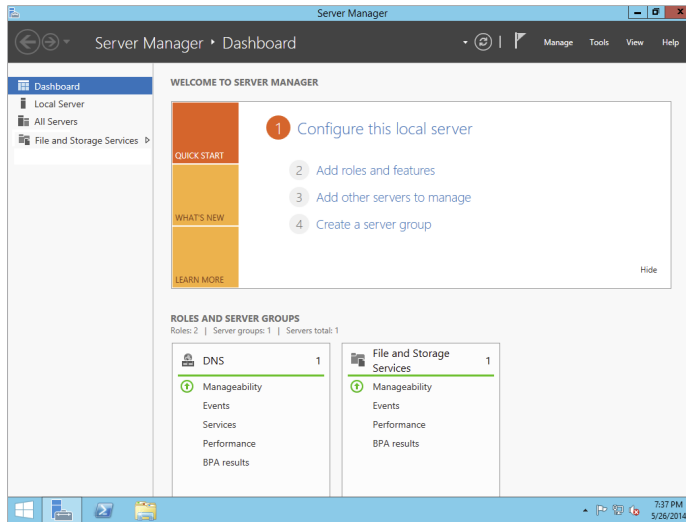
If you are currently using DHCP on your server, change your DNS server to have a static IP address before completing this exercise. I know that most of you already know how to change your IP address, but for anyone new to Microsoft Server 2012 R2, click the Start button and go into Control Panel. Make sure that the VIEW BY in the upper-right corner is set to large icons, and then choose Network and Sharing Center. On the right side of the windows under Access Type, click the access type link you have (Ethernet, Wireless, and so forth). Choose Properties, and then click Internet Protocol Version 4 (TCP/IPv4) and choose Properties. Choose the radio button for Use The Following IP Address and enter your local TCP/IP settings. Click OK to change it from DHCP to a static TCP/IP address.

Installing DNS

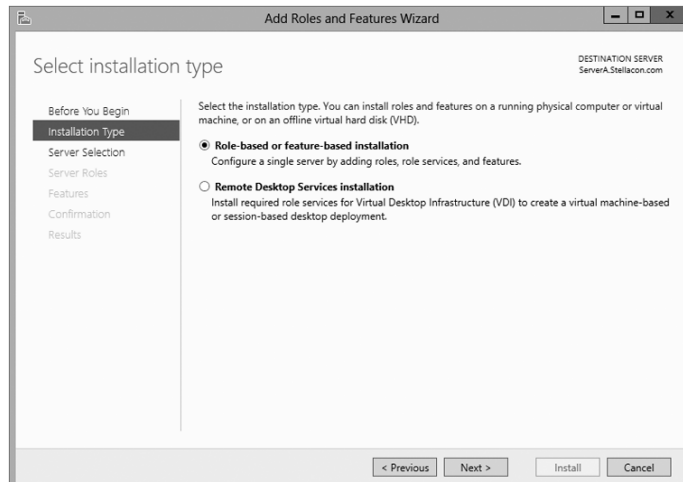
Let's start by installing DNS. Installing DNS is an important part of running a network. Exercise 2.1 walks you through the installation of a DNS server.

EXERCISE 2.1**Installing and Configuring the DNS Service**

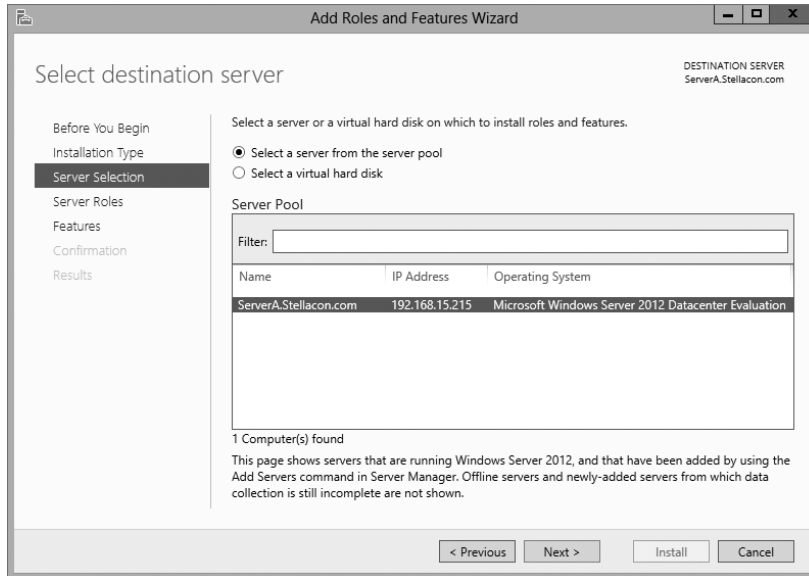
1. Open Server Manager.
2. On the Server Manager dashboard, click the Add Roles And Features link.



3. If a Before You Begin screen appears, click Next.
4. On the Selection type page, choose Role-Based Or Feature-Based Installation and click Next.



5. Click the **Select A Server From The Server Pool** radio button and choose the server under the **Server Pool** section. Click **Next**.



6. Click the **DNS Server Item** in the **Server Role** list. If a pop-up window appears telling you that you need to add additional features, click the **Add Features** button. If you did not give your system a static TCP/IP address, a window appears stating that you need a static TCP/IP address; just click the **Continue** button to bypass the error. Click **Next** to continue.
 7. On the **Add Features** page, just click **Next**.
 8. Click **Next** on the **DNS Server** information screen.
 9. On the **Confirm Installation** screen, choose the **Restart The Destination Server Automatically If Required** check box and then click the **Install** button.
 10. At the **Installation progress** screen, click **Close** after the **DNS server** is installed.
 11. Close **Server Manager**.
-

Load Balancing with Round Robin

Like other DNS implementations, the Windows Server 2012 R2 implementation of DNS supports load balancing through the use of round robin. Load balancing distributes the network load among multiple network hosts if they are available. You set up round-robin load balancing by creating multiple resource records with the same hostname but different IP addresses for multiple computers. Depending on the options you select, the DNS server responds with the addresses of one of the host computers.

If round robin is enabled, when a client requests name resolution, the first address entered in the database is returned to the resolver and is then sent to the end of the list. The next time a client attempts to resolve the name, the DNS server returns the second name in the database (which is now the first name) and then sends it to the end of the list, and so on. Round robin is enabled by default.

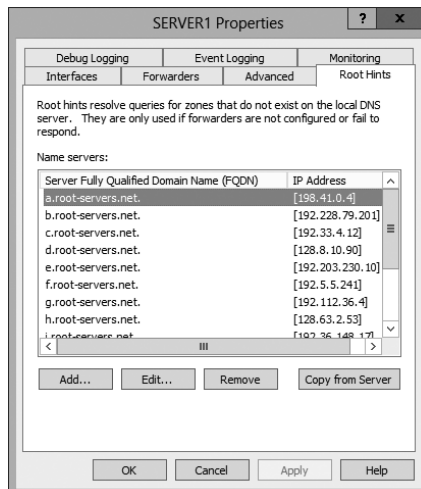
Configuring a Caching-Only Server

Although all DNS name servers cache queries that they have resolved, caching-only servers are DNS name servers that only perform queries, cache the answers, and return the results. They are not authoritative for any domains, and the information that they contain is limited to what has been cached while resolving queries. Accordingly, they don't have any zone files, and they don't participate in zone transfers. When a caching-only server is first started, it has no information in its cache; the cache is gradually built over time.

Caching-only servers are easy to configure. After installing the DNS service, simply make sure the root hints are configured properly:

1. Right-click your DNS server and choose the Properties command.
2. When the Properties dialog box appears, switch to the Root Hints tab (see Figure 2.11).

FIGURE 2.11 The Root Hints tab of the DNS server's Properties dialog box



3. If your server is connected to the Internet, you should see a list of root hints for the root servers maintained by ICANN and the Internet Assigned Numbers Authority (IANA). If not, click the Add button to add root hints as defined in the `cache.dns` file.

You can obtain current `cache.dns` files on the Internet by using a search engine. Just search for `cache.dns` and download one. (I always try to get `cache.dns` files from a university or a company that manages domain names.)

Setting Zone Properties

There are six tabs on the Properties dialog box for a forward or reverse lookup zone. You only use the Security tab to control who can change properties and to make dynamic updates to records on that zone. The other tabs are discussed in the following sections.



Secondary zones don't have a Security tab, and their SOA tab shows you the contents of the master SOA record, which you can't change.

General Tab

The General tab includes the following:

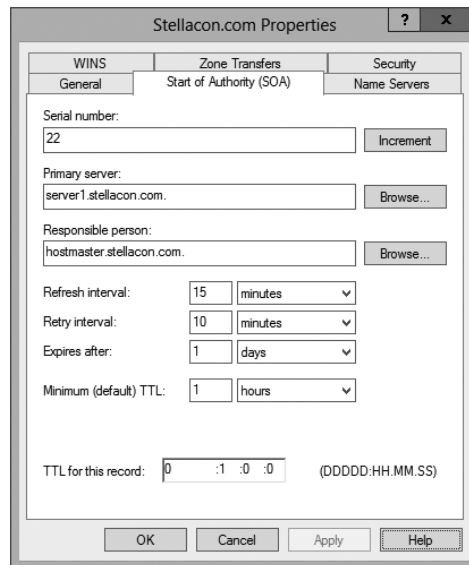
- The Status indicator and the associated Pause button let you see and control whether this zone can be used to answer queries. When the zone is running, the server can use it to answer client queries; when it's paused, the server won't answer any queries it gets for that particular zone.
- The Type indicator and its Change button allow you to select the zone type. The options are Standard Primary, Standard Secondary, and AD-Integrated. (See "Introducing DNS Database Zones" earlier in this chapter.) As you change the type, the controls you see below the horizontal dividing line change too. For primary zones, you'll see a field that lets you select the zone filename; for secondary zones, you'll get controls that allow you to specify the IP addresses of the primary servers. But the most interesting controls are the ones you see for AD Integrated zones. When you change to the AD Integrated zones, you have the ability to make the dynamic zones Secure Only.
- The Replication indicator and its Change button allow you to change the replication scope if the zone is stored in Active Directory. You can choose to replicate the zone data to any of the following:
 - All DNS servers in the Active Directory forest
 - All DNS servers in a specified domain
 - All domain controllers in the Active Directory domain (required if you use Windows 2000 domain controllers in your domain)
 - All domain controllers specified in the replication scope of the application directory partition

- The Dynamic Updates field gives you a way to specify whether you want to support Dynamic DNS updates from compatible DHCP servers. As you learned earlier in the section “Dynamic DNS and Non-Dynamic DNS,” the DHCP server or DHCP client must know about and support Dynamic DNS in order to use it, but the DNS server has to participate too. You can turn dynamic updates on or off, or you can require that updates be secured.

Start Of Authority (SOA) Tab

The options on the Start Of Authority (SOA) tab, shown in Figure 2.12, control the contents of the SOA record for this zone.

FIGURE 2.12 The Start Of Authority (SOA) tab of the zone Properties dialog box



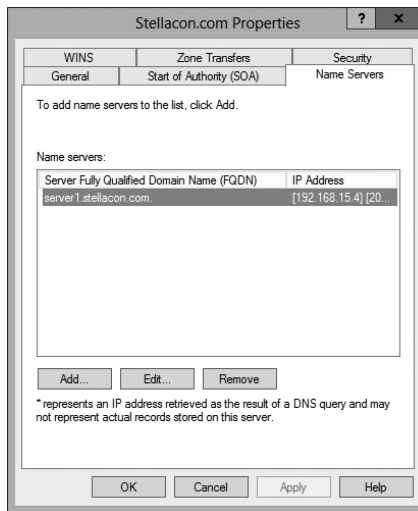
- The Serial Number field indicates which version of the SOA record the server currently holds. Every time you change another field, you should increment the serial number so that other servers will notice the change and get a copy of the updated record.
- The Primary Server and Responsible Person fields indicate the location of the primary name server (NS) for this zone and the email address of the administrator responsible for the maintenance of this zone, respectively. The standard username for this is hostmaster.
- The Refresh Interval field controls how often any secondary zones of this zone must contact the primary zone server and get any changes that have been posted since the last update.
- The Retry Interval field controls how long secondary servers will wait after a zone transfer fails before they try again. They'll keep trying at the interval you specify (which should be shorter than the refresh interval) until they eventually succeed in transferring zone data.

- The Expires After field tells the secondary servers when to throw away zone data. The default of 1 day (24 hours) means that a secondary server that hasn't gotten an update in 24 hours will delete its local copy of the zone data.
- The Minimum (Default) TTL field sets the default TTL for all RRs created in the zone. You can assign specific TTLs to individual records if you want.
- The TTL For This Record field controls the TTL for the SOA record itself.

Name Servers Tab

The *name server (NS) record* for a zone indicates which name servers are authoritative for the zone. That normally means the zone primary server and any secondary servers you've configured for the zone. (Remember, secondary servers are authoritative read-only copies of the zone.) You edit the NS record for a zone using the Name Servers tab (see Figure 2.13). The tab shows you which servers are currently listed, and you use the Add, Edit, and Remove buttons to specify which name servers you want included in the zone's NS record.

FIGURE 2.13 The Name Servers tab of the zone Properties dialog box



WINS Tab

The WINS tab allows you to control whether this zone uses WINS forward lookups. These lookups pass on queries that DNS can't resolve to WINS for action. This is a useful setup if you're still using WINS on your network. You must explicitly turn this option on with the Use WINS Forward Lookup check box on the WINS tab for a particular zone.

Zone Transfers Tab

Zone transfers are necessary and useful because they're the mechanism used to propagate zone data between primary and secondary servers. For primary servers (whether AD Integrated or not), you can specify whether your servers will allow zone transfers and, if so, to whom.

You can use the following controls on the Zone Transfers tab to configure these settings per zone:

- The Allow Zone Transfers check box controls whether the server answers zone transfer requests for this zone at all; when it's not checked, no zone data is transferred. The Allow Zone Transfers selections are as follows:
 - To Any Server allows any server anywhere on the Internet to request a copy of your zone data.
 - Only To Servers Listed On The Name Servers Tab (the default) limits transfers to servers you specify. This is a more secure setting than To Any Server because it limits transfers to other servers for the same zone.
 - Only To The Following Servers allows you to specify exactly which servers are allowed to request zone transfers. This list can be larger or smaller than the list specified on the Name Servers tab.
- The Notify button is for setting up automatic notification triggers that are sent to secondary servers for this zone. Those triggers signal the secondary servers that changes have occurred on the primary server so that the secondary servers can request updates sooner than their normally scheduled interval. The options in the Notify dialog box are similar to those in the Zone Transfers tab. You can enable automatic notification and then choose either Servers Listed On The Name Servers Tab or The Following Servers.

Configuring Zones for Dynamic Updates

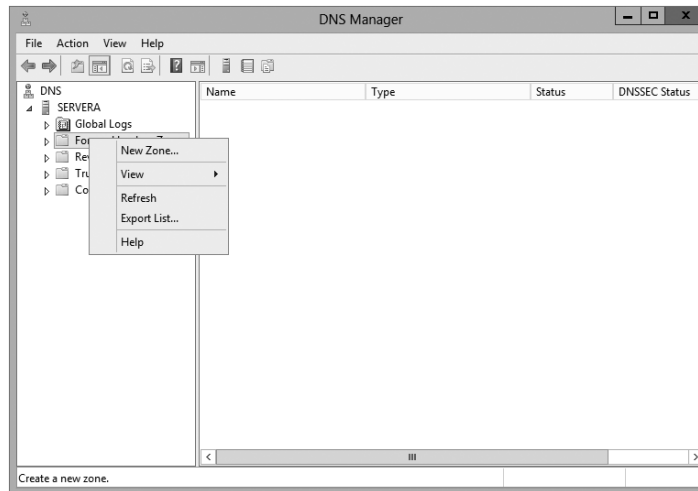
In Exercise 2.2, you will create and then modify the properties of a forward lookup zone. In addition, you'll configure the zone to allow dynamic updates. You are installing this DNS zone as a primary zone *without* AD integration. Even if you installed this DNS server onto a domain controller, do not choose the check box for AD integration.

EXERCISE 2.2

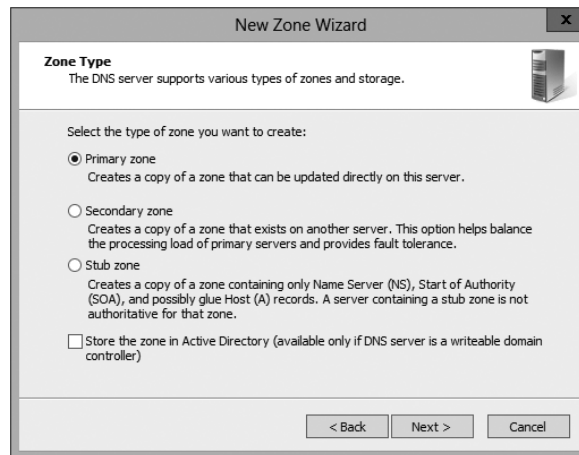


Configuring a Zone for Dynamic Updates

1. Open the DNS management snap-in by selecting Server Manager. Once in Server Manager, click DNS on the left side. In the Servers window (center screen), right-click your server name and choose DNS Manager.
2. Click the DNS server to expand it and then click the Forward Lookup Zones folder. Right-click the Forward Lookup Zones folder and choose New Zone.



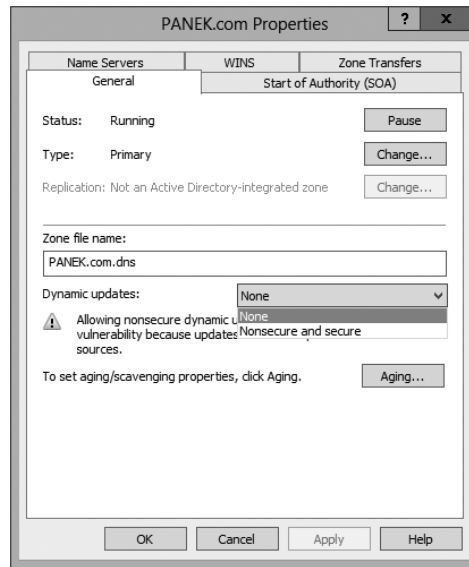
3. At the New Zone Welcome screen, click Next.
4. At the Zone Type screen, choose the Primary Zone option. Again, if your DNS server is also a domain controller, do not check the box to store the zone in Active Directory. Click Next when you are ready.



5. Enter a new zone name in the Zone Name field and click Next. (I used my last name—Panek.com.)
6. Leave the default zone filename and click Next.
7. Select the Do Not Allow Dynamic Updates radio button and click Next.
8. Click Finish to end the wizard.
9. Right-click the zone you just created and choose the Properties command.

EXERCISE 2.2 (continued)

- Click the down arrow next to Dynamic Updates. Notice that there are only two options (None and Nonsecure And Secure). The Secure Only option is not available because you are not using Active Directory Integrated. Make sure Nonsecure And Secure is chosen.



- Click OK to close the Properties box.
- Close the DNS management snap-in.
- Close the Server Manager snap-in.

Delegating Zones for DNS

DNS provides the ability to divide the namespace into one or more zones, which can then be stored, distributed, and replicated to other DNS servers. When deciding whether to divide your DNS namespace to make additional zones, consider the following reasons to use additional zones:

- A need to delegate management of part of your DNS namespace to another location or department within your organization.
- A need to divide one large zone into smaller zones for distributing traffic loads among multiple servers, for improving DNS name-resolution performance, or for creating a more fault-tolerant DNS environment.
- A need to extend the namespace by adding numerous subdomains at once, such as to accommodate the opening of a new branch or site.

Each newly delegated zone requires a primary DNS server just as a regular DNS zone does. When delegating zones within your namespace, be aware that for each new zone you create, you need to place delegation records in other zones that point to the authoritative DNS servers for the new zone. This is necessary both to transfer authority and to provide correct referral to other DNS servers and clients of the new servers being made authoritative for the new zone.

In Exercise 2.3, you'll create a delegated subdomain of the domain you created back in Exercise 2.2. Note that the name of the server to which you want to delegate the subdomain must be stored in an A or CNAME record in the parent domain.

EXERCISE 2.3

Creating a Delegated DNS Zone

1. Open the DNS management snap-in by selecting Server Manager. Once in Server Manager, click DNS on the left side. In the Servers window (center screen), right-click your server name and choose DNS Manager.
2. Expand the DNS server and locate the zone you created in Exercise 2.2.
3. Right-click the zone and choose the New Delegation command.
4. The New Delegation Wizard appears. Click Next to dismiss the initial wizard page.
5. Enter **ns1** (or whatever other name you like) in the Delegated Domain field of the Delegated Domain Name page. This is the name of the domain for which you want to delegate authority to another DNS server. It should be a subdomain of the primary domain (for example, to delegate authority for `farmington.example.net`, you'd enter **farmington** in the Delegated Domain field). Click Next to complete this step.
6. When the Name Servers page appears, click the Add button to add the name(s) and IP address(es) of the servers that will be hosting the newly delegated zone. For the purpose of this exercise, enter the server name you used in Exercise 2.2. Click the Resolve button to resolve this domain name's IP address automatically into the IP address field. Click OK when you are finished. Click Next to continue with the wizard.
7. Click the Finish button. The New Delegation Wizard disappears, and you'll see the new zone you just created appear beneath the zone you selected in step 3. The newly delegated zone's folder icon is drawn in gray to indicate that control of the zone is delegated.

DNS Forwarding

If a DNS server does not have an answer to a DNS request, it may be necessary to send that request to another DNS server. This is called *DNS forwarding*. You need to understand the two main types of forwarding:

External Forwarding When a DNS server forwards an external DNS request to a DNS server outside of your organization, this is considered *external forwarding*. For example, a

resolver requests the host `www.microsoft.com`. Most likely, your internal DNS server is not going to have Microsoft's web address in its DNS database. So, your DNS server is going to send the request to an external DNS (most likely your ISP).

Conditional Forwarding *Conditional forwarding* is a lot like external forwarding except that you are going to forward requests to specific DNS servers based on a condition. Usually, this is an excellent setup for internal DNS resolution. For example, let's say you have two companies, `stellacon.com` and `stellatest.com`. If a request comes in for `Stellacon.com`, it gets forwarded to the Stellacon DNS server, and any requests for `Stellatest.com` will get forwarded to the Stellatest DNS server. Requests are forwarded to a specific DNS server depending on the condition that an administrator sets up.

Manually Creating DNS Records

From time to time, you may find it necessary to add resource records manually to your Windows Server 2012 R2 DNS servers. Although Dynamic DNS frees you from the need to fiddle with A and PTR records for clients and other such entries, you still have to create other resource types (including MX records, required for the proper flow of SMTP email) manually. You can manually create A, PTR, MX, SRV, and many other record types.

There are only two important things to remember for manually creating DNS records:

- You must right-click the zone and choose either the New Record command or the Other New Records command.
- You must know how to fill in the fields of whatever record type you're using.

For example, to create an MX record, you need three pieces of information (the domain, the mail server, and the priority). To create an SRV record, however, you need several more pieces of information.

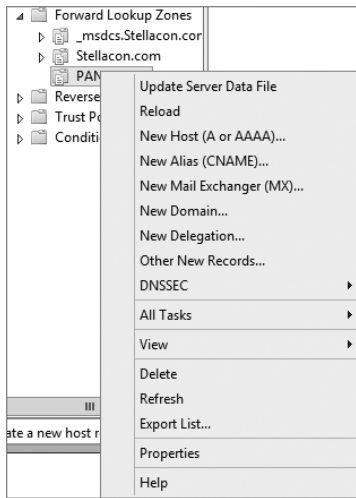
In Exercise 2.4, you will manually create an MX record for a mailtest server in the zone you created in Exercise 2.2.

EXERCISE 2.4



Manually Creating DNS RRs

1. Open the DNS management snap-in by selecting Server Manager. Once in Server Manager, click DNS on the left side. In the Servers window (center screen), right-click your server name and choose DNS Manager.
2. Expand your DNS server, right-click its zone and choose New Host (A record).
3. Enter `mailtest` in the Name field. Enter a TCP/IP number in the IP Address field. (You can use any number for this exercise, for example, 192.168.1.254.) Click the Add Host button.
4. A dialog box appears stating that the host record was created successfully. Click OK. Click Done.



5. Right-click your zone name and choose New Mail Exchanger (MX).
6. Enter **mailtest** in the Host Or Child Domain field and enter **mailtest.yourDomain.com** (or whatever domain name you used in Exercise 2.2) in the Fully-Qualified Domain Name (FQDN) Of Mail Server field; then click OK. Notice that the new record is already visible.
7. Next create an alias (or CNAME) record to point to the mail server. (It is assumed that you already have an A record for mailtest in your zone.) Right-click your zone, and choose New Alias (CNAME).
8. Type **mail** into the Alias Name field.
9. Type **mailtest.yourDomain.com** into the Fully-Qualified Domain Name (FQDN) For Target Host field.
10. Click the OK button.
11. Close the DNS management snap-in.

DNS Aging and Scavenging

When using dynamic updates, computers (or DHCP) will register a resource record with DNS. These records get removed when a computer is shut down properly. A major problem in the industry is that laptops are frequently removed from the network without a proper shutdown. Therefore, their resource records continue to live in the DNS database.

Windows Server 2012 R2 DNS supports two features called *DNS aging* and *DNS scavenging*. These features are used to clean up and remove stale resource records of a primary DNS zone. DNS aging and DNS scavenging flags old resource records that have

not been updated in a certain amount of time (determined by the scavenging interval). These stale records will be scavenged at the next cleanup interval. DNS uses time stamps on the resource records to determine how long they have been listed in the DNS database.

DNS allows an administrator to set up and configure aging and scavenging through the use of the DNS snap-in. DNS aging and scavenging allows an administrator to perform some of the following related tasks for your DNS servers and any of the Active Directory–Integrated zones that they load:

- Administrators can enable or disable the use of scavenging at a DNS server and/or for selected zones at the DNS server.
- You can modify the no-refresh/refresh interval, either as a server default or by specifying an overriding value at selected zones.
- Administrators can specify when periodic scavenging occurs automatically at the DNS server for any of its eligible zones and how often these operations are repeated.
- Manually initiate a single scavenging operation for all eligible zones at the DNS server.

Enabling Scavenging of Stale Records

When you install Windows Server 2012 R2 DNS aging and scavenging features on all DNS servers and any of their zones, administrators should consider the following settings before using these features:

Determine whether you should use aging and scavenging for server-wide settings. When using these settings, you are choosing to affect every one of the zone-level properties for all Active Directory–Integrated zones that are loaded at the server.

Determine whether you should use aging and scavenging for zone settings. When using these settings, you are choosing to use them for zone-specific properties for just the selected zones and not the entire server. These settings apply only to the applicable zone and its resource records, and they do not apply to any other zone on the DNS server. Unless these zone-level properties are otherwise configured, they inherit their defaults from comparable settings that are maintained in server aging and scavenging properties.



Enabling aging and scavenging for use with standard primary zones modifies the format of zone files. This change does not affect zone replication to secondary servers, but the modified zone files cannot be loaded by other versions of DNS servers.

Monitoring and Troubleshooting DNS

Now that you have set up and configured your DNS name server and created some resource records, you will want to confirm that it is resolving and replying to client DNS requests. A couple of tools allow you to do some basic monitoring and managing. Once you are able to monitor DNS, you'll want to start troubleshooting.

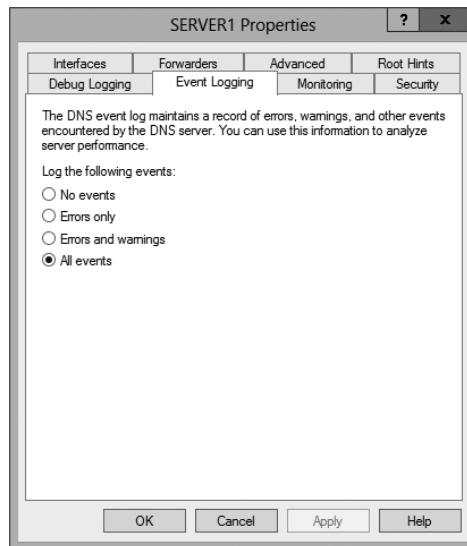
The simplest test is to use the `ping` command to make sure that the server is alive. A more thorough test would be to use `nslookup` to verify that you can actually resolve addresses for items on your DNS server.

In the following sections, you'll look at some of these monitoring and management tools and how to troubleshoot DNS.

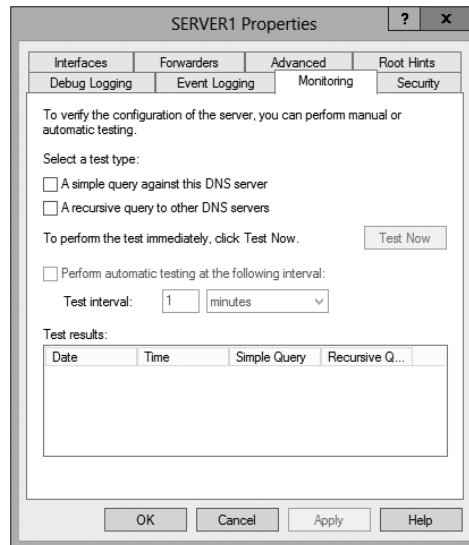
Monitoring DNS with the DNS Snap-In

You can use the DNS snap-in to do some basic server testing and monitoring. More important, you use the snap-in to monitor and set logging options. On the Event Logging tab of the server's Properties dialog box (see Figure 2.14), you can pick which events you want logged. The more events you select, the more logging information you'll get. This is useful when you're trying to track what's happening with your servers, but it can result in a very large log file if you're not careful.

FIGURE 2.14 The Event Logging tab of the server's Properties dialog box



The Monitoring tab (see Figure 2.15) gives you some testing tools. When the check box labeled *A Simple Query Against This DNS Server* is checked, a test is performed that asks for a single record from the local DNS server. It's useful for verifying that the service is running and listening to queries, but not much else. When the check box labeled *A Recursive Query To Other DNS Servers* is checked, the test is more sophisticated—a recursive query checks whether forwarding is working okay. The *Test Now* button and the *Perform Automatic Testing At The Following Interval* check box allow you to run these tests now or later as you require.

FIGURE 2.15 The Monitoring tab of the server's Properties dialog box

Another tab in the server's properties that allows you to monitor the activity of the DNS server is the Debug Logging tab. The Debug Logging tab allows you to monitor all outbound and inbound DNS traffic, packet content, packet type, and which transport protocol (TCP or UDP) that you want to monitor on the DNS server.



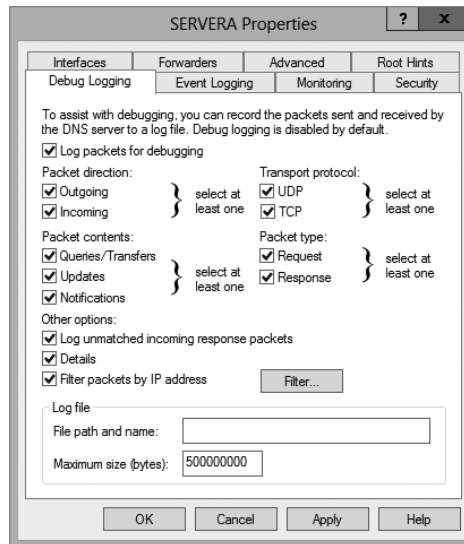
If the simple query fails, check that the local server contains the zone `1.0.0.127.in-addr.arpa`. If the recursive query fails, check that your root hints are correct and that your root servers are running.

In Exercise 2.5, you will enable logging, use the DNS MMC to test the DNS server, and view the contents of the DNS log.

EXERCISE 2.5

Simple DNS Testing

1. Open the DNS management snap-in by selecting Server Manager. Once in Server Manager, click DNS on the left side. In the Servers window (center screen), right-click your server name and choose DNS Manager.
2. Right-click the DNS server name on the top left and select Properties.
3. Switch to the Debug Logging tab, check all of the debug logging options except Filter Packets By IP Address, and enter a full path and filename in the File Path And Name field. Click the Apply button.



4. Switch to the Monitoring tab and check both A Simple Query Against This DNS Server and A Recursive Query To Other DNS Servers.
5. Click the Test Now button several times and then click OK.
6. Press the Windows key on the keyboard (left side between the Ctrl and Alt keys) and then choose Computer. Navigate to the folder that you specified in step 3 and use WordPad or Notepad to view the contents of the log file.

Troubleshooting DNS

When troubleshooting DNS problems, ask yourself the following basic questions:

- What application is failing? What works? What doesn't work?
- Is the problem basic IP connectivity, or is it name resolution? If the problem is name resolution, does the failing application use NetBIOS names, DNS names, or host-names?
- How are the things that do and don't work related?
- Have the things that don't work ever worked on this computer or network? If so, what has changed since they last worked?

Windows Server 2012 R2 provides several useful tools, discussed in the following sections, which can help you answer these questions:

- `Nslookup` is used to perform DNS queries and to examine the contents of zone files on local and remote servers.
- `DNSLint` is a command-line utility used for troubleshooting many common DNS issues.

- Ipconfig allows you to perform the following tasks:
 - View DNS client settings
 - Display and flush the resolver cache
 - Force a dynamic update client to register its DNS records
- The DNS log file monitors certain DNS server events and logs them for your edification.

Using Nslookup

Nslookup is a standard command-line tool provided in most DNS server implementations, including Windows Server 2012 R2. Windows Server 2012 R2 gives you the ability to launch nslookup from the DNS snap-in.



When nslookup is launched from the DNS snap-in, a command prompt window opens automatically. You enter nslookup commands in this window.

Nslookup offers you the ability to perform query testing of DNS servers and to 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. You can do a number of useful things with nslookup:

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



When you are entering queries, it is generally a good idea to enter FQDNs so that 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.

Using Nslookup on the Command Line

To use nslookup in plain-old command-line mode, enter the following in the command prompt window:

```
nslookup DNS_name_or_IP_address server_IP_address
```

This command will look up a DNS name or address using a server at the IP address you specify.

Using *Nslookup* in Interactive Mode

Nslookup is a lot more useful in interactive mode because you can enter several commands in sequence. Entering **nslookup** by itself (without specifying a query or server) puts it in interactive mode, where it will stay until you type **exit** and press Enter. Before that point, you can look up lots of useful stuff. The following are some of the tasks you can perform with *nslookup* in interactive mode:

Setting Options with the set Command While in interactive mode, you can use the **set** command to configure how the resolver will carry out queries. Table 2.5 shows a few of the options available with **set**.

TABLE 2.5 Command-line options available with the **set** command

Option	Purpose
<code>set all</code>	Shows all the options available.
<code>set d2</code>	Puts <i>nslookup</i> in debug mode so that you can examine the query and response packets between the resolver and the server.
<code>set domain=<i>domain name</i></code>	Tells the resolver what domain name to append for unqualified queries.
<code>set timeout=<i>timeout</i></code>	Tells the resolver how long to keep trying to contact the server. This option is useful for slow links where queries frequently time out and the wait time must be lengthened.
<code>set type=<i>record type</i></code>	Tells the resolver which 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 settype=all .

Looking Up a Name While in interactive mode, you can look up a name just by typing it: **stellacon.com**. In this example, *stellacon* is the owner name for the record for which you are searching, and *.com* is the server 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, just type **k*** as your query.

Looking Up a Record Type If you want to query a particular type of record (for instance, an MX record), use the **set type** command. The command `set type=mx` tells *nslookup* you're interested only in seeing MX records that meet your search criteria.

Listing the Contents of a Domain To get a list of the contents of an entire domain, use the `ls` command. To find all of the hosts in your domain, you'd type `set type=a` and then type `ls -t yourdomain.com`.

Troubleshooting Zone Transfers You can simulate zone transfers by using the `ls` command with the `-d` switch. This can help you determine whether the server you are querying allows zone transfers to your computer. To do this, type `ls -d domain__name`.

Nslookup Responses and Error Messages

A successful `nslookup` response looks like this:

```
Server: Name_of_DNS_server
Address: IP_address_of_DNS_server
Response_data
```

`Nslookup` might also return an error message. Some common messages are listed in Table 2.6.

TABLE 2.6 Common `nslookup` error messages

Error message	Meaning
<pre>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</pre>	<p>The resolver did not locate a PTR resource record (containing the hostname) for the server IP address you specified. <code>Nslookup</code> can still query the DNS server, and the DNS server can still answer queries.</p>
<pre>*** Request to Server timed-out</pre>	<p>A request was not fulfilled in the allotted time. This might happen, for example, if the DNS service was not running on the DNS server that is authoritative for the name.</p>
<pre>*** Server can't find Name_or_IP_ address_queried_for: No response from server</pre>	<p>The server is not receiving requests on User Datagram Protocol (UDP) port 53.</p>
<pre>*** Server can't find Name_or_IP_ address_queried_for: Non-existent domain</pre>	<p>The DNS server was unable to find the name or IP address in the authoritative domain. The authoritative domain might be on the remote DNS server or on another DNS server that this DNS server is unable to reach.</p>

```
*** Server can't find Name_or_IP_
address_queried_for: Server failed
```

The DNS server is running, but it 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. You will also receive the error if the DNS server cannot contact its root servers, it is not connected to the Internet, or it has no root hints.

In Exercise 2.6, you'll get some hands-on practice with the `nslookup` tool. You can run this exercise from Windows 7, Windows 8, and Windows Server 2012 R2.

EXERCISE 2.6

Using the `nslookup` Command

1. Click the Start button and in the Search Programs And Files box (above the Start button), type **CMD**. Then hit Enter.
2. Type **nslookup** and press the Enter key. (For the rest of the exercise, use the Enter key to terminate each command.)
3. Try looking up a well-known address: Type **www.microsoft.com**.
4. Try looking up a nonexistent host: Type **www.example.ccccc**. Notice that your server indicates that it can't find the address and times out. This is normal behavior.
5. Type **exit** at the prompt. Type **exit** again to leave the command prompt.

Using *DNSLint*

Microsoft Windows Server 2012 R2 DNS can use the `DNSLint` command-line utility to help diagnose some common DNS name-resolution issues and to help diagnose potential problems of incorrect delegation. You need to download `DNSLint` from the Microsoft Download Center.

`DNSLint` uses three main functions to verify DNS records and to generate a report in HTML:

dnslint/d This function helps diagnose the reasons for “lame delegation” and other related DNS problems.

dnslint/ql This function helps verify a user-defined set of DNS records on multiple DNS servers.

dnslint/ad This function helps verify DNS records pertaining to Active Directory replication.

Here is the syntax for DNSLint:

```
dnslint /d domain_name | /ad [LDAP_IP_address] | /ql input_file
[/c [smtp,pop,imap]] [/no_open] [/r report_name]
[/t] [/test_tcp] [/s DNS_IP_address] [/v] [/y]
```

The following are some sample queries:

```
dnslint /d stellacon.com
dnslint /ad /s 192.168.36.201
dnslint /ql dns_server.txt
dnslint /ql autocreate
dnslint /v /d stellacon.com
dnslint /r newfile /d stellacon.com
dnslint /y /d stellacon.com
dnslint /no_open /d stellacon.com
```

Table 2.7 explains the command options.

TABLE 2.7 DNSLint command options

Command option	Meaning
/d	Domain name that is being tested.
/ad	Resolves DNS records that are used for Active Directory forest replication.
/s	TCP/IP address of host.
/ql	Requests DNS query tests from a list. This switch sends DNS queries specified in an input file.
/v	Turns on verbose mode.
/r filename	Allows you to create a report file.
/y	Overwrites an existing report file without being prompted.
/no_open	Prevents a report from opening automatically.

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 the `ipconfig` command with no parameters, it displays DNS information for each adapter, including the domain name and

DNS servers used for that adapter. Table 2.8 shows some command-line options available with `ipconfig`.

TABLE 2.8 Command-line options available for the `ipconfig` command

Command	What It Does
<code>ipconfig /all</code>	Displays additional information about DNS, including the FQDN and the DNS suffix search list.
<code>ipconfig /flushdns</code>	Flushes and resets the DNS resolver cache. For more information about this option, see the section “Configuring DNS” earlier in this chapter.
<code>ipconfig /displaydns</code>	Displays the contents of the DNS resolver cache. For more information about this option, see the section “Configuring DNS” earlier in this chapter.
<code>ipconfig /registerdns</code>	Refreshes all DHCP leases and registers any related DNS names. This option is available only on Windows 2000 and newer computers that run the DHCP client service.



You should know and be comfortable with the `ipconfig` commands related to DNS for the exam.

Using *DNSScmd*

`DNSScmd` allows you to display and change the properties of DNS servers, zones, and resource records through the use of command-line commands. The `DNSScmd` utility allows you to modify, create, and delete resource records and/or zones manually, and it allows you to force replication between two DNS servers.

Table 2.9 lists some of the `DNSScmd` commands and their explanations.

TABLE 2.9 `DNSScmd` command-line options

Command	Explanation
<code>dnscmd /clearcache</code>	Clears the DNS server cache
<code>dnscmd /config</code>	Resets DNS server or zone configuration
<code>dnscmd /createdirectorypartition</code>	Creates a DNS application directory partition
<code>dnscmd /deletedirectorypartition</code>	Deletes a DNS application directory partition
<code>dnscmd /enumrecords</code>	Shows the resource records in a zone

TABLE 2.9 Dnscmd command-line options (*continued*)

Command	Explanation
dnscmd /exportsettings	Creates a text file of all server configuration information
dnscmd /info	Displays server information
dnscmd /recordadd	Adds a resource record to a zone
dnscmd /recorddelete	Deletes a resource record from a zone
dnscmd /zoneadd	Creates a new DNS zone
dnscmd /zonedelete	Deletes a DNS zone
dnscmd /zoneexport	Creates a text file of all resource records in the zone
dnscmd /zoneinfo	Displays zone information
dnscmd /zonerefresh	Forces replication of the master zone to the secondary zone

Using the DNS Log File

You can configure the DNS server to create a log file that records the following information:

- Queries
- Notification messages from other servers
- Dynamic updates
- Content of the question section for DNS query messages
- 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 or Word to view it.

Once the log file reaches the maximum size, Windows Server 2012 R2 writes over the beginning of the file. You can change the maximum size of the log. If you increase the size value, data persists for a longer time period, but the log file consumes more disk space. If

you decrease the value, the log file uses less disk space, but the data persists for a shorter time period.



Do not leave DNS logging turned on during normal operation because it sucks up both processing and hard disk resources. Enable it only when diagnosing and solving DNS problems.

Troubleshooting the *.(root)* Zone

The *DNS root zone* is the top-level DNS zone in the DNS hierarchy. Windows Server 2012 R2-based DNS servers will build a *.(root)* zone when a connection to the Internet can't be found.

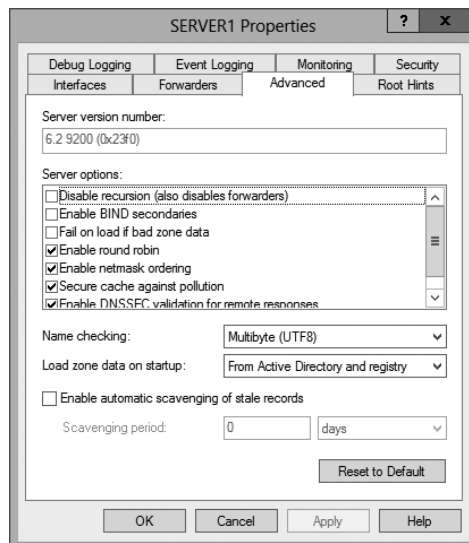
Because of this, the *.(root)* zone may prevent access to the Internet. The DNS forwarding option and DNS root hints will not be configurable. If you want your DNS to work as a DNS forwarder or you want to use root hints, you must remove the *.(root)* zone.

Issues with Non-Microsoft DNS Servers

Another troubleshooting problem that you may run into is working with both Microsoft DNS servers and non-Microsoft DNS servers. One of the most common non-Microsoft DNS servers is the Unix-based BIND DNS server.

If you need to complete a zone transfer from Microsoft DNS to a BIND DNS server, you need to enable BIND Secondaries on the Microsoft DNS server (see Figure 2.16).

FIGURE 2.16 Enabling BIND secondaries



If you need to enable Bind Secondaries, complete the following steps:

1. Open DNS management.
2. Right-click the server name and choose Properties.
3. Click the Advanced tab.
4. Check the Enable BIND Secondaries box.
5. Click OK.

Overview of DHCP

As you will see in Chapter 7, TCP/IP is the priority protocol for Windows Server 2012 R2. There are two ways to have clients and servers get TCP/IP addresses:

- You can manually assign the addresses.
- The addresses can be assigned automatically.

Manually assigning addresses is a fairly simple process. An administrator goes to each of the machines on the network and assigns TCP/IP addresses. The problem with this method arises when the network becomes midsized or larger. Think of an administrator trying to individually assign 4,000 TCP/IP addresses, subnet masks, default gateways, and all other configuration options needed to run the network.

DHCP's job is to centralize the process of IP address and option assignment. You can configure a DHCP server with a range of addresses (called a *pool*) and other configuration information and let it assign all of the IP parameters—addresses, default gateways, DNS server addresses, and so on.



DHCP is defined by a series of request for comments documents, notably 2131 and 2132.

Introducing the DORA Process

An easy way to remember how DHCP works is to learn the acronym DORA. *DORA* stands for Discover, Offer, Request, and Acknowledge. In brief, here is DHCP's DORA process:

1. *Discover*: When IP networking starts up on a DHCP-enabled client, a special message called a DHCPDISCOVER is broadcast within the local physical subnet.
2. *Offer*: Any DHCP server that hears the request checks its internal database and replies with a message called a DHCPOFFER, which contains an available IP address.

The contents of this message depend on how the DHCP server is configured—there are numerous options aside from an IP address that you can specify to pass to the client on a Windows Server DHCP server.

3. *Request*: The client receives one or more DHCPOFFERs (depending on how many DHCP servers exist on the local subnet), chooses an address from one of the offers, and sends a DHCPREQUEST message to the server to signal acceptance of the DHCPOFFER.

This message might also request additional configuration parameters.

Other DHCP servers that sent offers take the request message as an acknowledgment that the client didn't accept their offer.

4. *Acknowledge*: When the DHCP server receives the DHCPREQUEST, it marks the IP address as being in use (that is, usually, though it's not required). Then it sends a DHCPACK to the client.

The acknowledgment message might contain requested configuration parameters.

If the server is unable to accept the DHCPREQUEST for any reason, it sends a DHCPNAK message. If a client receives a DHCPNAK, it begins the configuration process over again.

5. When the client accepts the IP offer, the address is assigned to the client for a specified period of time, called a *lease*. After receiving the DHCPACK message, the client performs a final check on the parameters (sometimes it sends an ARP request for the offered IP address) and makes note of the duration of the lease. The client is now configured. If the client detects that the address is already in use, it sends a DHCPDECLINE.

If the DHCP server has given out all of the IP addresses in its pool, it won't make an offer. If no other servers make an offer, the client's IP network initialization will fail, and the client will use Automatic Private IP Addressing (APIPA).

DHCP Lease Renewal

No matter how long the lease period, the client sends a new lease request message directly to the DHCP server when the lease period is half over (give or take some randomness required by RFC 2131). This period goes by the name *T1* (not to be confused with the T1 type of network connection). If the server hears the request message and there's no reason to reject it, it sends a DHCPACK to the client. This resets the lease period.

If the DHCP server isn't available, the client realizes that the lease can't be renewed. The client continues to use the address, and once 87.5 percent of the lease period has elapsed (again, give or take some randomness), the client sends out another renewal request. This interval is known as *T2*. At that point, any DHCP server that hears the renewal can respond to this *DHCP request message* (which is a request for a lease renewal) with a DHCPACK and renew the lease. If at any time during this process the client gets a negative DHCPNACK message, it must stop using its IP address immediately and start the leasing process over from the beginning by requesting a new lease.

When a client initializes its IP networking, it always attempts to renew its old address. If the client has time left on the lease, it continues to use the lease until its end. If the client is unable to get a new lease by that time, all IP functions stop until a new, valid address can be obtained.

DHCP Lease Release

Although leases can be renewed repeatedly, at some point they might run out. Furthermore, the lease process is “at will.” That is, the client or server can cancel the lease before it ends. In addition, if the client doesn’t succeed in renewing the lease before it expires, the client loses its lease and reverts to APIPA. This release process is important for reclaiming extinct IP addresses used by systems that have moved or switched to a non-DHCP address.

Advantages and Disadvantages of DHCP

DHCP was designed from the start to simplify network management. It has some significant advantages, but it also has some drawbacks.

Advantages of DHCP

The following are advantages of DHCP:

- Configuration of large and even midsized networks is much simpler. If a DNS server address or some other change is necessary to the client, the administrator doesn’t have to touch each device in the network physically to reconfigure it with the new settings.
- Once you enter the IP configuration information in one place—the server—it’s automatically propagated to clients, eliminating the risk that a user will misconfigure some parameters and require you to fix them.
- IP addresses are conserved because DHCP assigns them only when requested.
- IP configuration becomes almost completely automatic. In most cases, you can plug in a new system (or move one) and then watch as it receives a configuration from the server. For example, when you install new network changes, such as a gateway or DNS server, the client configuration is done at only one location—the DHCP server.
- It allows a preboot execution environment (PXE) client to get a TCP/IP address from DHCP. PXE clients (also called Microsoft Windows Deployment Services [WDS] clients) can get an IP address without needing to have an operating system installed. This allows WDS clients to connect to a WDS server through the TCP/IP protocol and download an operating system remotely.

Disadvantages of DHCP

Unfortunately, there are a few drawbacks with DHCP:

- DHCP can become a single point of failure for your network. If you have only one DHCP server and it’s not available, clients can’t request or renew leases.

- If the DHCP server contains incorrect information, the misinformation will automatically be delivered to all of your DHCP clients.
- If you want to use DHCP on a multisegment network, you must put either a DHCP server or a relay agent on each segment, or you must ensure that your router can forward Bootstrap Protocol (BOOTP) broadcasts.

***ipconfig* Lease Options**

The `ipconfig` command-line tool is useful for working with network settings. Its `/renew` and `/release` switches make it particularly handy for DHCP clients. These switches allow you to request renewal of, or give up, your machine's existing address lease. You can do the same thing by toggling the Obtain An IP Address Automatically button in the Internet Protocol (TCP/IP) Properties dialog box, but the command-line option is useful especially when you're setting up a new network.

For example, I spend about a third of my time teaching MCSA or MCSE classes, usually in temporary classrooms set up at conferences, hotels, and so on. Laptops are used in these classes, with one brawny one set up as a DNS/DHCP/DC server. Occasionally, a client will lose its DHCP lease (or not get one, perhaps because a cable has come loose). The quickest way to fix it is to pop open a command-line window and type **`ipconfig/renew`**.

You can configure DHCP to assign options only to certain classes. *Classes*, defined by an administrator, are groups of computers that require identical DHCP options. The `/setclassidclassID` switch of `ipconfig` is the only way to assign a machine to a class.

More specifically, the switches do the following:

`ipconfig /renew` Instructs the DHCP client to request a lease renewal. If the client already has a lease, it requests a renewal from the server that issued the current lease. This is equivalent to what happens when the client reaches the half-life of its lease. Alternatively, if the client doesn't currently have a lease, it is equivalent to what happens when you boot a DHCP client for the first time. It initiates the DHCP mating dance, listens for lease offers, and chooses one it likes.

`ipconfig /release` Forces the client to give up its lease immediately by sending the server a DHCP release notification. The server updates its status information and marks the client's old IP address as "available," leaving the client with no address bound to its network interface. When you use this command, most of the time it will be immediately followed by `ipconfig/renew`. The combination releases the existing lease and gets a new one, probably with a different address. (It's also a handy way to force your client to get a new set of settings from the server before the lease expiration time.)

`ipconfig /setclassidclassID` Sets a new class ID for the client. You will see how to configure class options later in the section "Setting Scope Options for IPv4." For now, you should know that the only way to add a client machine to a class is to use this command. Note that you need to renew the client lease for the class assignment to take effect.

If you have multiple network adapters in a single machine, you can provide the name of the adapter (or adapters) upon which you want the command to work, including an

asterisk (*) as a wildcard. For example, one of my servers has two network cards: an Intel EtherExpress (ELNK1) and a generic 100Mbps card. If you want to renew DHCP settings for both adapters, you can type `ipconfig /renew *`. If you just want to renew the Intel EtherExpress card, you can type `ipconfig /renew ELNK1`.

Understanding Scope Details

By now you should have a good grasp of what a lease is and how it works. To learn how to configure your servers to hand out those leases, however, you need to have a complete understanding of some additional topics: scopes, superscopes, exclusions, reservations, address pool, and relay agents.

Scope

Let's start with the concept of a *scope*, which is a contiguous range of addresses. There's usually one scope per physical subnet, and a scope can cover a Class A, Class B, or Class C network address or a TCP/IP v6 address. DHCP uses scopes as the basis for managing and assigning IP addressing information.

Each scope has a set of parameters, or scope options, that you can configure. *Scope options* control what data is delivered to DHCP clients when they're completing the DHCP negotiation process with a particular server. For example, the DNS server name, default gateway, and default network time server are all separate options that can be assigned. These settings are called *option types*. You can use any of the types provided with Windows Server 2012 R2, or you can specify your own.

Superscope

A *superscope* enables the DHCP server to provide addresses from more than one scope to clients on the same physical subnet. This is helpful when clients within the same subnet have more than one IP network and thus need IPs from more than one address pool. Microsoft's DHCP snap-in allows you to manage IP address assignment in the superscope, though you must still configure other scope options individually for each child scope.

Exclusions and Reservations

The scope defines what IP addresses could potentially be assigned, but you can influence the assignment process in two additional ways by specifying exclusions and reservations:

Exclusions These are IP addresses within the range that you never want automatically assigned. These excluded addresses are off-limits to DHCP. You'll typically use exclusions to tag any addresses that you never want the DHCP server to assign at all. You might use exclusions to set aside addresses that you want to assign permanently to servers that play a vital role in your organization.

Reservations These are IP addresses within the range for which you want a permanent DHCP lease. They essentially reserve a particular IP address for a particular device. The device still goes through the DHCP process (that is, its lease expires and it asks for a new one), but it always obtains the same addressing information from the DHCP server.



Exclusions are useful for addresses that you don't want to participate in DHCP at all. *Reservations* are helpful for situations in which you want a client to get the same settings each time they obtain an address.



An address cannot be simultaneously reserved and excluded. Be aware of this fact for the exam, possibly relating to a troubleshooting question.



Real World Scenario

Using Reservations and Exclusions

Deciding when to assign a reservation or exclusion can sometimes be confusing. In practice, you'll find that certain computers in the network greatly benefit by having static IP network information. Servers such as DNS servers, the DHCP server itself, SMTP servers, and other low-level infrastructure servers are good candidates for static assignment. There are usually so few of these servers that the administrator is not overburdened if a change in network settings requires going out to reconfigure each individually. Chances are that the administrator would still need to reconfigure these servers manually (by using `ipconfig /release` and then `ipconfig /renew`), even if they did not have IP addresses reserved. Even in large installations, I find it preferable to manage these vital servers by hand rather than to rely on DHCP.

Reservations are also appropriate for application servers and other special but nonvital infrastructure servers. With a reservation in DHCP, the client device will still go through the DHCP process but will always obtain the same addressing information from the DHCP server. The premise behind this strategy is that these nonvital servers can withstand a short outage if DHCP settings change or if the DHCP server fails.

Address Pool

The range of IP addresses that the DHCP server can assign is called its *address pool*. For example, let's say you set up a new DHCP scope covering the 192.168.1 subnet. That gives you 255 IP addresses in the pool. After adding an exclusion from 192.168.1.240 to 192.168.1.254, you're left with 241 (255 - 14) IP addresses in the pool. That means (in theory, at least) that you can service 241 unique clients at a time before you run out of IP addresses.

DHCP Relay Agent

By design, DHCP is intended to work only with clients and servers on a single IP network to communicate. But RFC 1542 sets out how BOOTP (on which DHCP is based) should work in circumstances in which the client and server are on different IP networks. If no DHCP server is available on the client's network, you can use a DHCP relay agent to forward DHCP broadcasts from the client's network to the DHCP server. The relay agent acts like a radio repeater, listening for DHCP client requests and retransmitting them through the router to the server.

Installing and Authorizing DHCP

Installing DHCP is easy using the Windows Server 2012/2012 R2 installation mechanism. Unlike some other services discussed in this book, the installation process installs just the service and its associated snap-in, starting it when the installation is complete. At that point, it's not delivering any DHCP service, but you don't have to reboot.

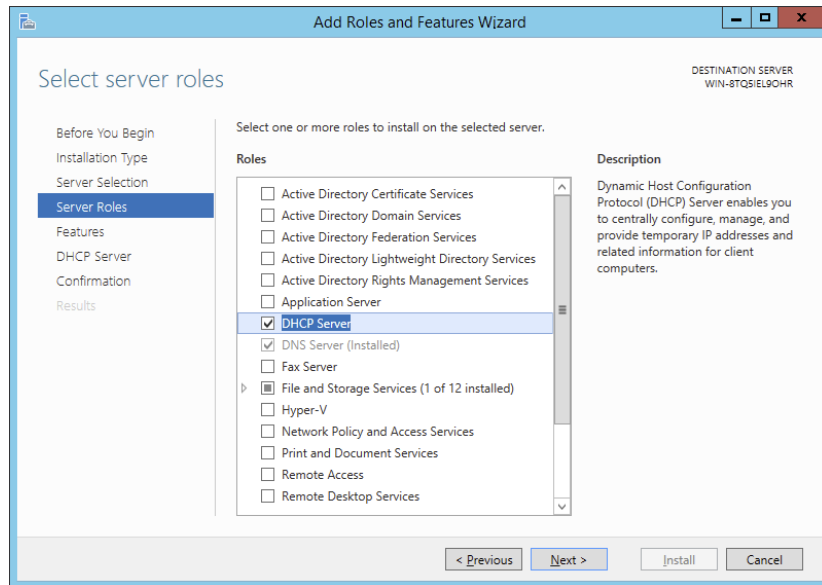
Installing DHCP

Exercise 2.7 shows you how to install DHCP Server using Server Manager. This exercise was completed on a Windows Server 2012 R2 Member Server since Active Directory is not installed yet.

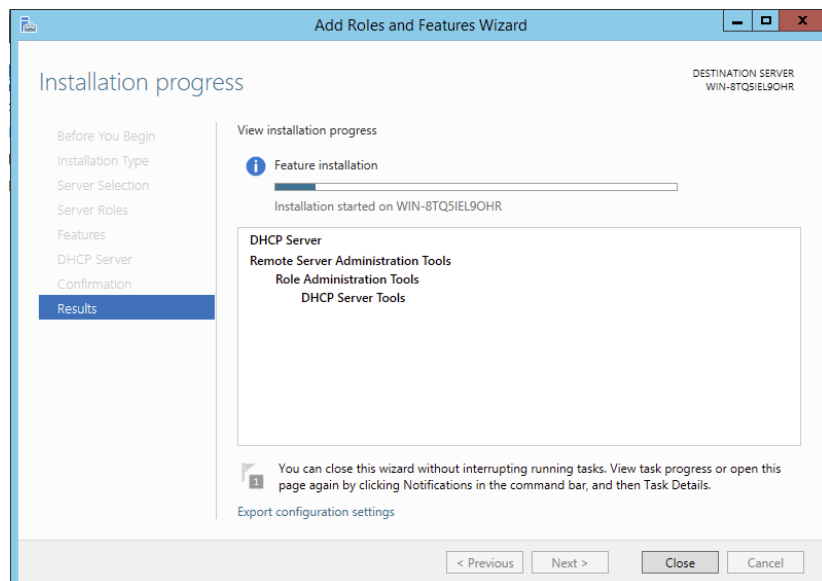
EXERCISE 2.7

Installing the DHCP Service

1. Choose Server Manager by clicking the Server Manager icon on the taskbar.
2. Click Add Roles And Features.
3. Choose role-based or feature-based installation and click Next.
4. Choose your server and click Next.
5. Choose DHCP and click Next.

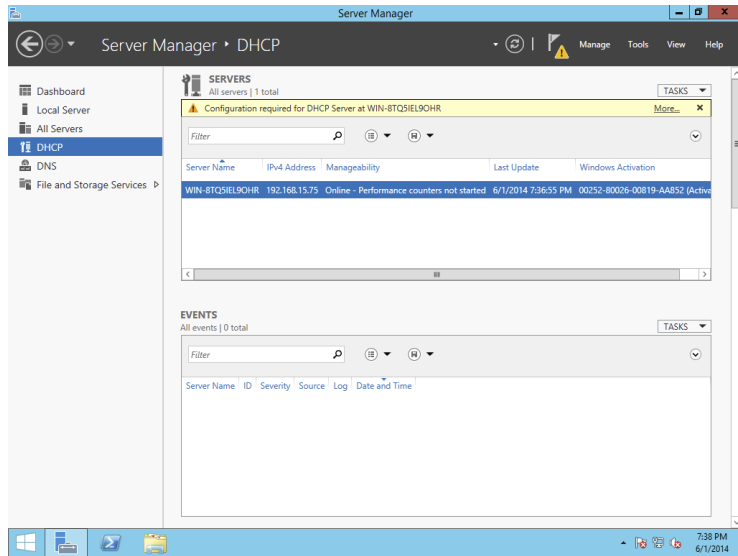


6. At the Features screen, click Next.
7. Click Next at the DHCP screen.
8. At the DHCP confirmation screen, click the Install button.

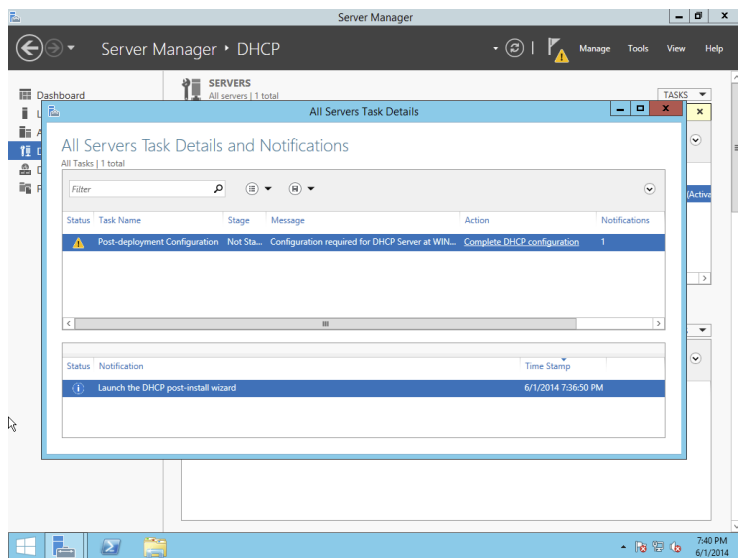


EXERCISE 2.7 (continued)

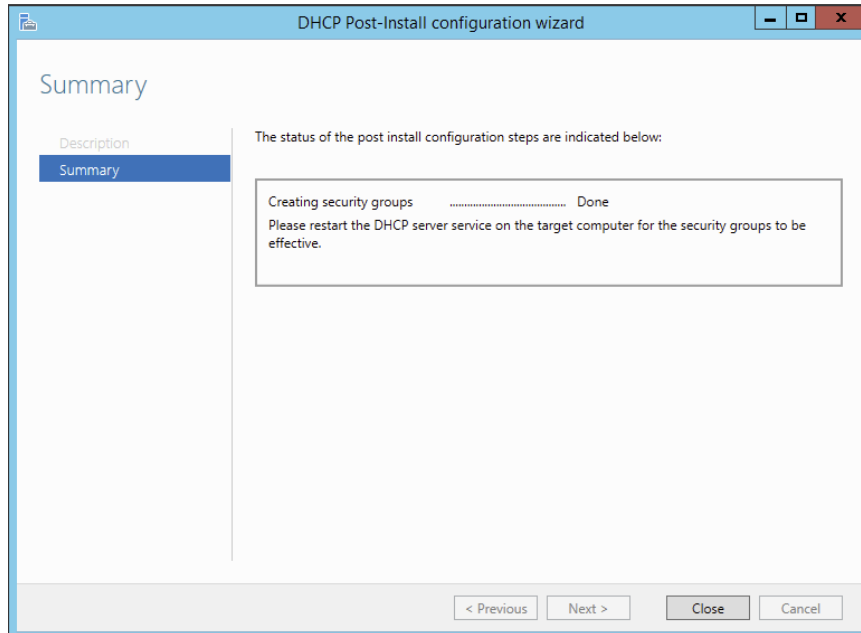
9. When the installation is complete, click the Close button.
10. On the left side, click the DHCP link.
11. Click the More link next to Configuration Required For DHCP Server.



12. Under Action, click Complete DHCP Configuration.



13. At the DHCP Description page, click Commit.
14. Click Close at the Summary screen.



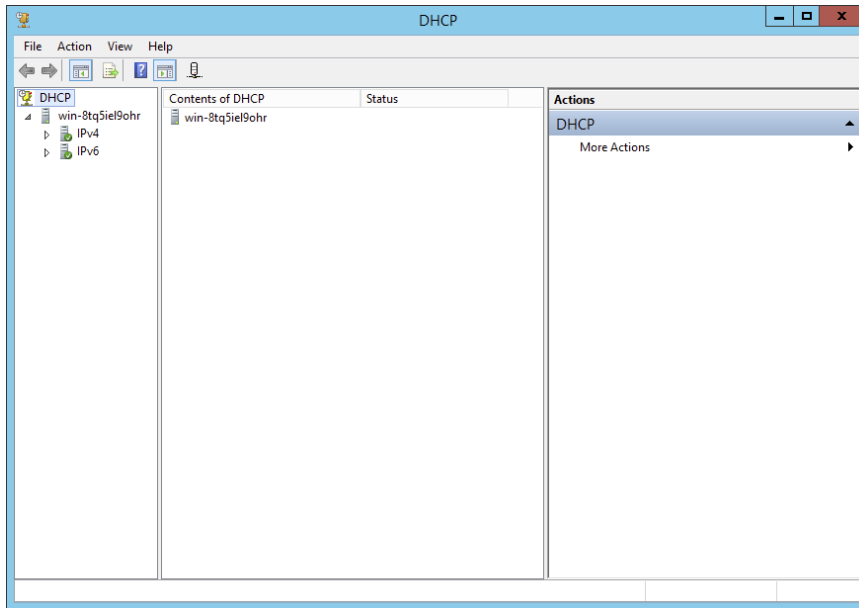
15. Close Server Manager.
-

Introducing the DHCP Snap-In

When you install the DHCP server, the DHCP snap-in is also installed. You can open it by selecting Administrative Tools > DHCP. Figure 2.17 shows the snap-in.

As you can see, the snap-in follows the standard MMC model. The left pane displays IPv4 and IPv6 sections and which servers are available; you can connect to servers other than the one to which you're already connected. A Server Options folder contains options that are specific to a particular DHCP server. Each server contains subordinate items grouped into folders. Each scope has a folder named after the scope's IP address range. Within each scope, four subordinate views show you interesting things about the scope, such as the following:

- The Address Pool view shows what the address pool looks like.
- The Address Leases view shows one entry for each current lease. Each lease shows the computer name to which the lease was issued, the corresponding IP address, and the current lease expiration time.

FIGURE 2.17 DHCP snap-in

- The Reservations view shows the IP addresses that are reserved and which devices hold them.
- The Scope Options view lists the set of options you've defined for this scope.

Authorizing DHCP for Active Directory

Authorization creates an Active Directory object representing the new server. It helps keep unauthorized servers off your network. Unauthorized servers can cause two kinds of problems. They may hand out bogus leases, or they may fraudulently deny renewal requests from legitimate clients.

When you install a DHCP server using Windows Server 2012/2012 R2 and Active Directory is present on your network, the server won't be allowed to provide DHCP services to clients until it has been authorized. If you install DHCP on a member server in an Active Directory domain or on a stand-alone server, you'll have to authorize the server manually. When you authorize a server, you're adding its IP address to the Active Directory object that contains the IP addresses of all authorized DHCP servers.



You also have the ability to authorize a DHCP server during the installation of DHCP if you are installing DHCP onto an Active Directory machine.

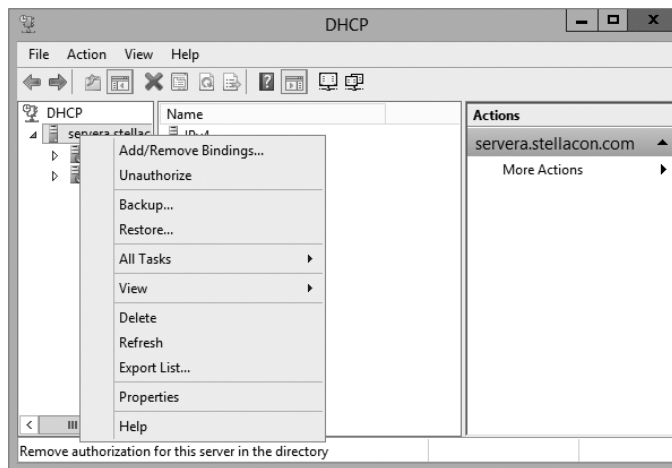
At start time, each DHCP server queries the directory, looking for its IP address on the “authorized” list. If it can’t find the list or if it can’t find its IP address on the list, the DHCP service fails to start. Instead, it adds a message to the event log, indicating that it couldn’t service client requests because the server wasn’t authorized.

Exercise 2.8 and Exercise 2.9 show you how to authorize and unauthorize a DHCP server onto a network with Active Directory. If you installed DHCP onto a network with a domain, you can complete the following two exercises, but if you are still on a member server, you *cannot* do these exercises. These are here to show you how to do it after you have Active Directory on your network.

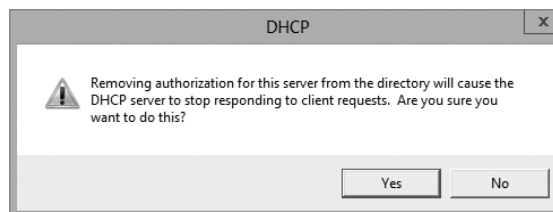
EXERCISE 2.8

Unauthorizing a DHCP Server

1. From Administrative Tools, choose DHCP to open the DHCP snap-in.
2. Right-click the server you want to unauthorize and choose the Unauthorize command.

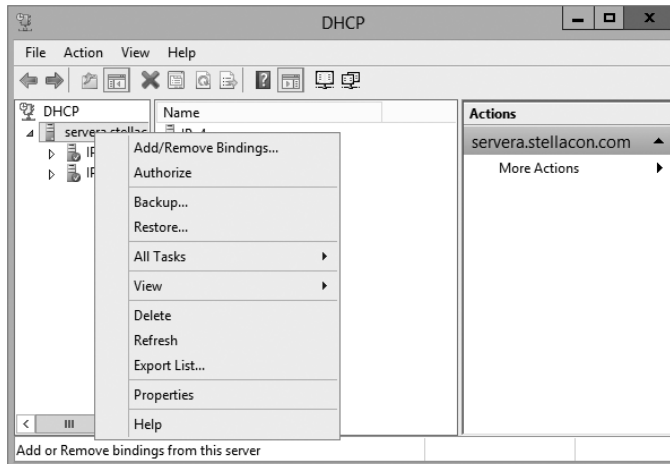


3. Click Yes on the dialog box asking if you are sure you want to complete this action.



EXERCISE 2.9**Authorizing a DHCP Server**

1. From Administrative Tools, choose DHCP to open the DHCP snap-in.
2. Right-click the server you want to authorize and choose the Authorize command.



3. Wait a short time (30 to 45 seconds) to allow the authorization to take place.
4. Right-click the server again. Verify that the Unauthorize command appears in the pop-up menu. This indicates that the server is now authorized.

Creating and Managing DHCP Scopes

You can use any number of DHCP servers on a single physical network if you divide the range of addresses that you want assigned into multiple scopes. Each scope contains a number of useful pieces of data, but before you can understand them, you need to know some additional terminology.

You can perform the following management tasks on DHCP scopes:

- Create a scope
- Configure scope properties
- Configure reservations and exclusions
- Set scope options

- Activate and deactivate scopes
- Create a superscope
- Create a multicast scope
- Integrate Dynamic DNS and DHCP

I will cover each task in the following sections.

Creating a New Scope in IPv4

Like many other things in Windows Server 2012 R2, a wizard drives the process of creating a new scope. You will most likely create a scope while installing DHCP, but you may need to create more than one. The overall process is simple, as long as you know beforehand what the wizard is going to ask. If you think about what defines a scope, you'll be well prepared. You need to know the following:

- The IP address range for the scope you want to create.
- Which IP addresses, if any, you want to exclude from the address pool.
- Which IP addresses, if any, you want to reserve.
- Values for the DHCP options you want to set, if any. This item isn't strictly necessary for creating a scope. However, to create a useful scope, you'll need to have some options to specify for the clients.

To create a scope, under the server name, right-click the IPv4 option in the DHCP snap-in, and use the Action > New Scope command. This starts the New Scope Wizard (see Figure 2.18). You will look at each page of the wizard in the following sections.

FIGURE 2.18 Welcome page of the New Scope Wizard



Setting the Screen Name

The Scope Name page allows you to enter a name and description for your scope. These will be displayed by the DHCP snap-in.



It's a good idea to pick sensible names for your scopes so that other administrators will be able to figure out the purpose of the scope. For example, the name DHCP is likely not very helpful, whereas a name like 1st Floor Subnet is more descriptive and can help in troubleshooting.

Defining the IP Address Range

The IP Address Range page (see Figure 2.19) is where you enter the start and end IP addresses for your range. The wizard does minimal checking on the addresses you enter, and it automatically calculates the appropriate subnet mask for the range. You can modify the subnet mask if you know what you're doing.

FIGURE 2.19 IP Address Range page of the New Scope Wizard

New Scope Wizard

IP Address Range
You define the scope address range by identifying a set of consecutive IP addresses.

Enter the range of addresses that the scope distributes.

Start IP address:

End IP address:

A subnet mask defines how many bits of an IP address to use for the network/subnet IDs and how many bits to use for the host ID. You can specify the subnet mask by length or as an IP address.

Length:

Subnet mask:

< Back Next > Cancel

Adding Exclusions and Delay

The Add Exclusions And Delay page (see Figure 2.20) allows you to create exclusion ranges. Exclusions are TCP/IP numbers that are in the pool, but they do not get issued to clients. To exclude one address, put it in the Start IP Address field. To exclude a range, also fill in the End IP Address field. The delay setting is a time duration by which the server will delay the transmission of a DHCP OFFER message.



Although you can always add exclusions later, it's best to include them when you create the scope so that no excluded addresses are ever passed out to clients.

FIGURE 2.20 Add Exclusions And Delay page of the New Scope Wizard

New Scope Wizard

Add Exclusions and Delay

Exclusions are addresses or a range of addresses that are not distributed by the server. A delay is the time duration by which the server will delay the transmission of a DHCP OFFER message.

Type the IP address range that you want to exclude. If you want to exclude a single address, type an address in Start IP address only.

Start IP address: End IP address:

Excluded address range:

Subnet delay in mill second:

< Back Next > Cancel

Setting a Lease Duration

The Lease Duration page (see Figure 2.21) allows you to set how long a device gets to use an assigned IP address before it has to renew its lease. The default lease duration is eight days. You may find that a shorter or longer duration makes sense for your network. If your network is highly dynamic, with lots of arrivals, departures, and moving computers, set a shorter lease duration; if it's less active, make it longer.

FIGURE 2.21 Lease Duration page of the New Scope Wizard

New Scope Wizard

Lease Duration

The lease duration specifies how long a client can use an IP address from this scope.

Lease durations should typically be equal to the average time the computer is connected to the same physical network. For mobile networks that consist mainly of portable computers or dial-up clients, shorter lease durations can be useful. Likewise, for a stable network that consists mainly of desktop computers at fixed locations, longer lease durations are more appropriate.

Set the duration for scope leases when distributed by this server.

Limited to:

Days: Hours: Minutes:

< Back Next > Cancel

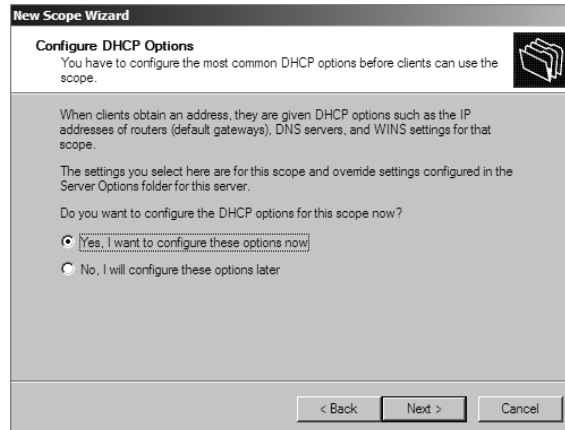


Remember that renewal attempts begin when approximately half of the lease period is over (give or take a random interval), so don't set them too short.

Configuring Basic DHCP Options

The Configure DHCP Options page (see Figure 2.22) allows you to choose whether you want to set up basic DHCP options such as default gateway and DNS settings. The options are described in the following sections. If you choose not to configure options, you can always do so later. However, you should not activate the scope until you've configured the options you want assigned.

FIGURE 2.22 Configure DHCP Options page of the New Scope Wizard



Configuring a Router

The first option configuration page is the Router (Default Gateway) page (see Figure 2.23), in which you enter the IP addresses of one or more routers (more commonly referred to as *default gateways*) that you want to use for outbound traffic. After entering the IP addresses of the routers, use the Up and Down buttons to order the addresses. Clients will use the routers in the order specified when attempting to send outgoing packets.

Providing DNS Settings

On the Domain Name And DNS Servers page (see Figure 2.24), you specify the set of DNS servers and the parent domain you want passed down to DHCP clients. Normally, you'll want to specify at least one DNS server by filling in its DNS name or IP address. You can also specify the domain suffix that you want clients to use as the base domain for all connections that aren't fully qualified. For example, if your clients are used to navigating based on server name alone rather than the fully qualified domain name (FQDN) of `server.willpanek.com`, then you'll want to place your domain here.

FIGURE 2.23 Router (Default Gateway) page of the New Scope Wizard

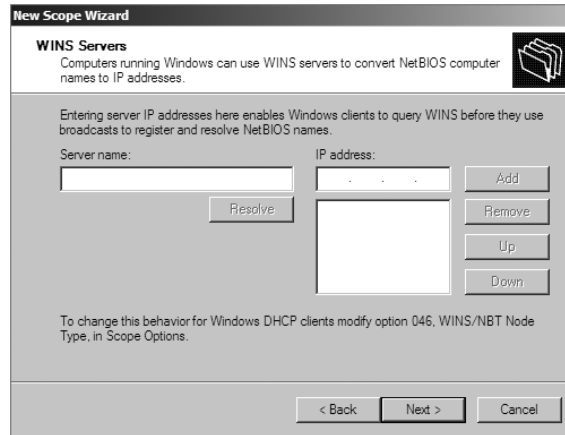
The screenshot shows the 'Router (Default Gateway)' page of the New Scope Wizard. The title bar reads 'New Scope Wizard'. Below the title bar, the page is titled 'Router (Default Gateway)' with a subtitle: 'You can specify the routers, or default gateways, to be distributed by this scope.' To the right of the subtitle is a folder icon. The main content area contains the instruction: 'To add an IP address for a router used by clients, enter the address below.' Below this is an 'IP address:' label followed by a text input field. To the right of the input field is an 'Add' button. Below the input field is a list box. To the right of the list box are four buttons: 'Remove', 'Up', and 'Down'. At the bottom of the window are three buttons: '< Back', 'Next >', and 'Cancel'.

FIGURE 2.24 Domain Name And DNS Servers page of the New Scope Wizard

The screenshot shows the 'Domain Name and DNS Servers' page of the New Scope Wizard. The title bar reads 'New Scope Wizard'. Below the title bar, the page is titled 'Domain Name and DNS Servers' with a subtitle: 'The Domain Name System (DNS) maps and translates domain names used by clients on your network.' To the right of the subtitle is a folder icon. The main content area contains the instruction: 'You can specify the parent domain you want the client computers on your network to use for DNS name resolution.' Below this is a 'Parent domain:' label followed by a text input field. Below the input field is the instruction: 'To configure scope clients to use DNS servers on your network, enter the IP addresses for those servers.' Below this is a 'Server name:' label followed by a text input field. To the right of the input field is an 'IP address:' label followed by a text input field. To the right of the IP address input field is an 'Add' button. Below the input fields is a list box. To the right of the list box are four buttons: 'Resolve', 'Remove', 'Up', and 'Down'. At the bottom of the window are three buttons: '< Back', 'Next >', and 'Cancel'.

Providing WINS Settings

If you're still using Windows Internet Name Service (WINS) on your network, you can configure DHCP so that it passes WINS server addresses to your Windows clients. (If you want the Windows clients to honor it, you'll also need to define the WINS/NBT Node Type option for the scope.) As on the DNS server page, on the WINS Servers page (see Figure 2.25) you can enter the addresses of several servers and move them into the order in which you want clients to try them. You can enter the DNS or NetBIOS name of each server, or you can enter an IP address.

FIGURE 2.25 WINS Servers page of the New Scope Wizard

Here are some of the more common options you can set on a DHCP server:

003 Router Used to provide a list of available routers or default gateways on the same subnet.

006 DNS Servers Used to provide a list of DNS servers.

015 DNS Domain Name Used to provide the DNS suffix.

028 Broadcast Address Used to configure the broadcast address, if different than the default, based on the subnet mask.

44 WINS/NBNS Servers Used to configure the IP addresses of WINS servers.

46 WINS/NBT Node Type Used to configure the preferred NetBIOS name resolution method. There are four settings for node type:

B node (0x1) Broadcast for NetBIOS resolution

P node (0x2) Peer-to-peer (WINS) server for NetBIOS resolution

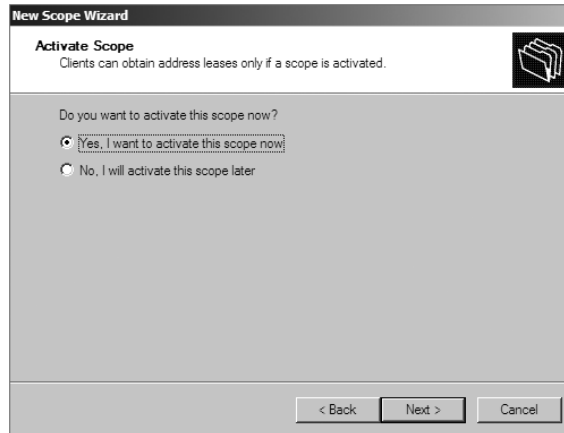
M node (0x4) Mixed node (does a B node and then a P node)

H node (0x8) Hybrid node (does a P node and then a B node)

051 Lease Used to configure a special lease duration.

Activating the Scope

The Activate Scope page (see Figure 2.26) gives you the option to activate the scope immediately after creating it. By default, the wizard assumes that you want the scope activated unless you select the No, I Will Activate This Scope Later radio button, in which case the scope will remain dormant until you activate it manually.

FIGURE 2.26 Activate Scope page of the New Scope Wizard

Be sure to verify that there are no other DHCP servers assigned to the address range you choose!

In Exercise 2.10, you will create a new scope for the 192.168.0.x private Class C network. First you need to complete Exercise 2.7 before beginning this exercise.

EXERCISE 2.10



Creating a New Scope

1. Open the DHCP snap-in by selecting Administrative Tools > DHCP.
2. Right-click the IPv4 folder and choose New Scope. The New Scope Wizard appears.
3. Click the Next button on the welcome page.
4. Enter a name and a description for your new scope and click the Next button.
5. On the IP Address Range page, enter **192.168.0.2** as the start IP address for the scope and **192.168.0.250** as the end IP address. Leave the subnet mask controls alone (though when creating a scope on a production network, you might need to change them). Click the Next button.
6. On the Add Exclusions And Delay page, click Next without adding any excluded addresses or delays.
7. On the Lease Duration page, set the lease duration to 3 days and click the Next button.
8. On the Configure DHCP Options page, click the Next button to indicate you want to configure default options for this scope.

EXERCISE 2.10 (continued)

9. On the Router (Default Gateway) page, enter **192.168.0.1** for the router IP address and then click the Add button. Once the address is added, click the Next button.
 10. On the Domain Name And DNS Servers page, enter the IP address of a DNS server on your network in the IP Address field (for example, you might enter **192.168.0.251**) and click the Add button. Click the Next button.
 11. On the WINS Servers page, click the Next button to leave the WINS options unset.
 12. On the Activate Scope page, if your network is currently using the 192.168.0.x range, select Yes, I Want To Activate This Scope Now. Click the Next button.
 13. When the wizard's summary page appears, click the Finish button to create the scope.
-

Creating a New Scope in IPv6

Now that you have seen how to create a new scope in IPv4, I'll go through the steps to create a new scope in IPv6.

To create a scope, right-click the IPv6 option in the DHCP snap-in under the server name and select the Action > New Scope command. This starts the New Scope Wizard. Just as with creating a scope in IPv4, the welcome page of the wizard tells you that you've launched the New Scope Wizard. You will look at each page of the wizard in the following sections.

Setting the Screen Name

The Scope Name page (see Figure 2.27) allows you to enter a name and description for your scope. These will be displayed by the DHCP snap-in.

FIGURE 2.27 IPv6 Scope Name page of the New Scope Wizard

New Scope Wizard

Scope Name
You have to provide an identifying scope name. You also have the option of providing a description.

Type a name and description for this scope. This information helps you quickly identify how the scope is to be used on your network.

Name:

Description:

< Back Next > Cancel



It's a good idea to pick a sensible name for your scopes so that other administrators will be able to figure out what the scope is used for.

Scope Prefix

The Scope Prefix page (see Figure 2.28) gets you started creating the IPv6 scope. IPv6 has three types of addresses, which can be categorized by type and scope.

FIGURE 2.28 Scope Prefix page of the New Scope Wizard

Unicast Addresses *One-to-one*. A packet from one host is delivered to another host. The following are some examples of IPv6 unicast:

- The unicast prefix for site-local addresses is FEC0::/48.
- The unicast prefix for link-local addresses is FE80::/64.

The 6to4 address allows communication between two hosts running both IPv4 and IPv6. The way to calculate the 6to4 address is by combining the global prefix 2002::/16 with the 32 bits of a public IPv4 address of the host. This gives you a 48-bit prefix. 6to4 is described in RFC 3056.

Multicast addresses *One-to-many*. A packet from one host is delivered to multiple hosts (but not everyone). The prefix for multicast addresses is FF00::/8.

Anycast addresses A packet from one host is delivered to the nearest of multiple hosts (in terms of routing distance).

Adding Exclusions

As with the IPv4 New Scope Wizard, the Add Exclusions page (see Figure 2.29) allows you to create exclusion ranges. *Exclusions* are TCP/IP numbers that are in the pool but do not get issued to clients. To exclude one address, put it in the Start IPv6 Address field. To exclude a range, also fill in the End IPv6 Address field.

FIGURE 2.29 Add Exclusions page of the New Scope Wizard

Setting a Lease Duration

The Scope Lease page (see Figure 2.30) allows you to set how long a device gets to use an assigned IP address before it has to renew its lease. You can set two different lease durations. The section labeled Non Temporary Address (IANA) is the lease time for your more permanent hosts (such as printers and server towers). The one labeled Temporary Address (IATA) is for hosts that might disconnect at any time, such as laptops.

Activating the Scope

The Completing The New Scope Wizard page (see Figure 2.31) gives you the option to activate the scope immediately after creating it. By default, the wizard will assume you want the scope activated. If you want to wait to activate the scope, choose No in the Activate Scope Now box.

Changing Scope Properties (IPv4 and IPv6)

Each scope has a set of properties associated with it. Except for the set of options assigned by the scope, you can find these properties on the General tab of the scope's Properties

FIGURE 2.30 Scope Lease page of the New Scope Wizard

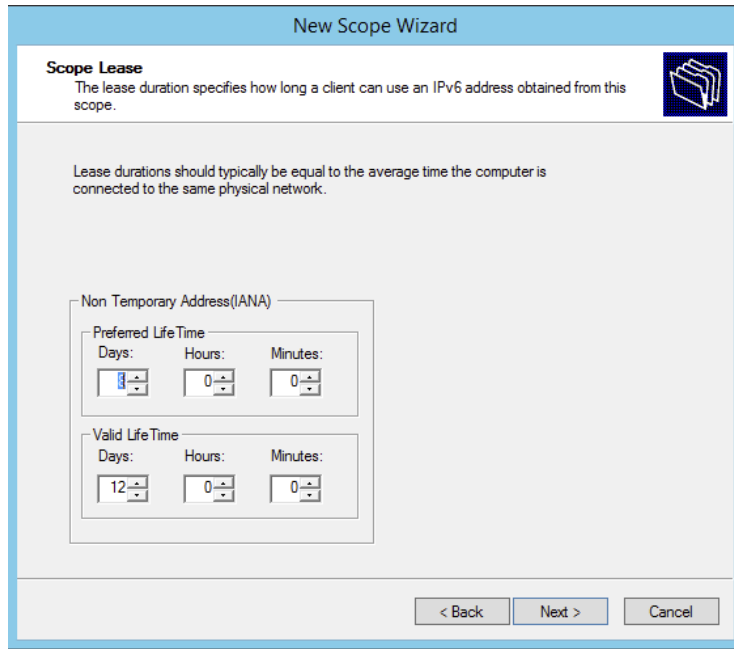
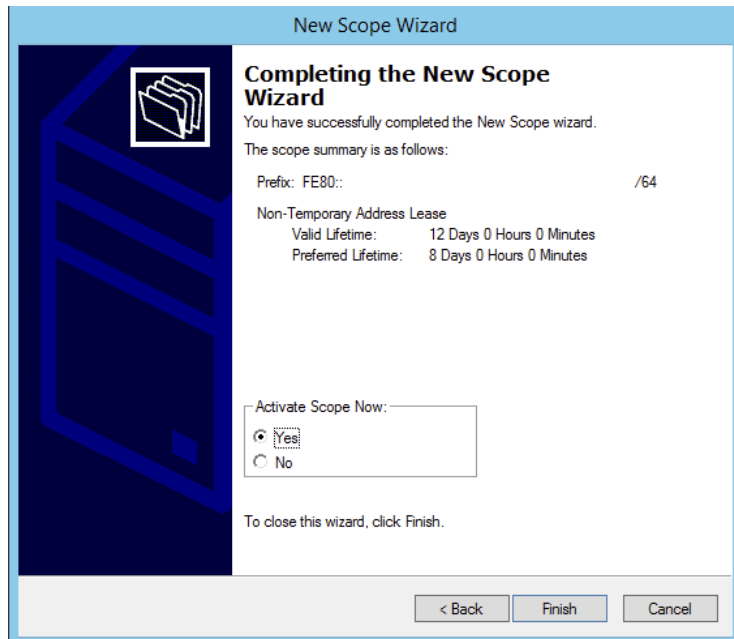
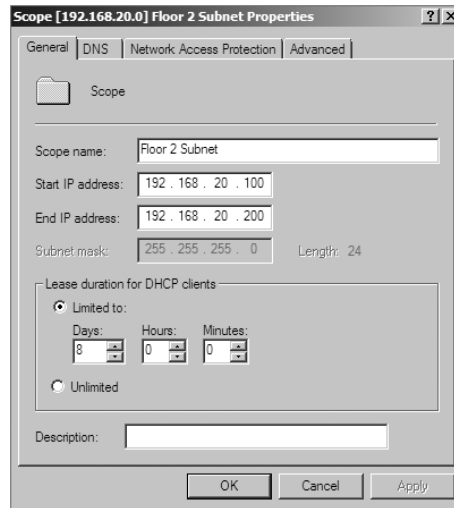


FIGURE 2.31 Completing The New Scope Wizard page of the New Scope Wizard



dialog box (see Figure 2.32). Some of these properties, such as the scope name and description, are self-explanatory. Others require a little more explanation.

FIGURE 2.32 General tab of the scope's Properties dialog box for an IPv4 scope



- The Start IP Address and End IP Address fields allow you to set the range of the scope.
- For IPv4 scopes, the settings in the section Lease Duration For DHCP Clients control how long leases in this scope are valid.

The IPv6 scope dialog box includes a Lease tab where you set the lease properties.



When you make changes to these properties, they have no effect on existing leases. For example, say you create a scope from 172.30.1.1 to 172.30.1.199. You use that scope for a while and then edit its properties to reduce the range from 172.30.1.1 to 172.30.1.150. If a client has been assigned the address 172.30.1.180, which was part of the scope before you changed it, the client will retain that address until the lease expires but will not be able to renew it.

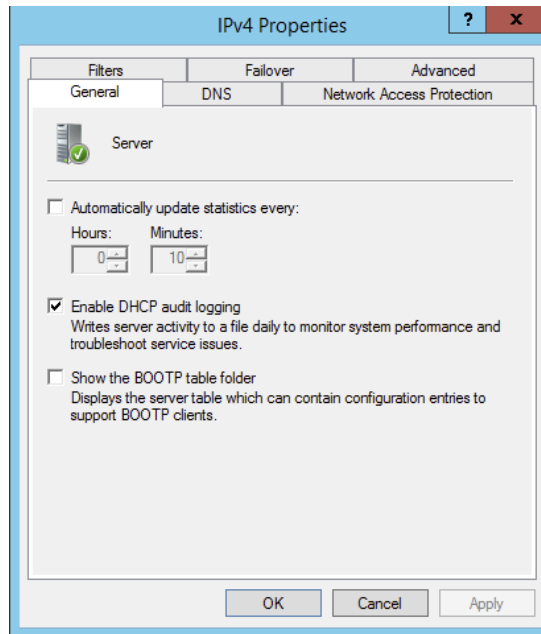
Changing Server Properties

Just as each scope has its own set of properties, so too does the server itself. You access the server properties by right-clicking the IPv4 or IPv6 object within the DHCP management console and selecting Properties.

IPv4 Server Properties

Figure 2.33 shows the IPv4 Properties dialog box.

FIGURE 2.33 General tab of the IPv4 Properties dialog box for the server



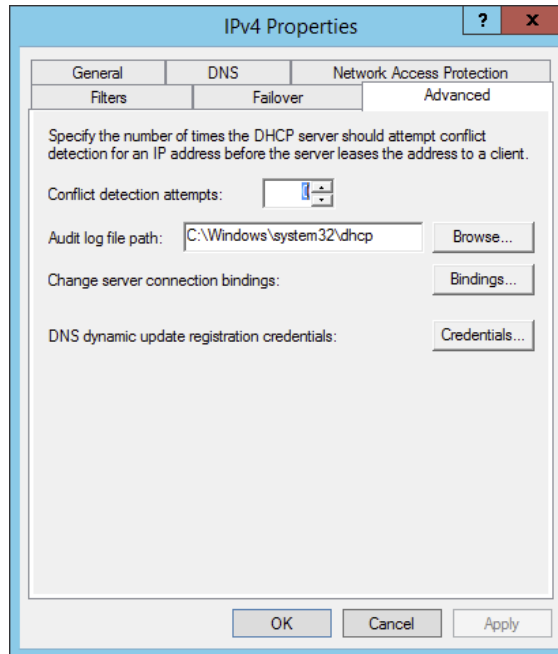
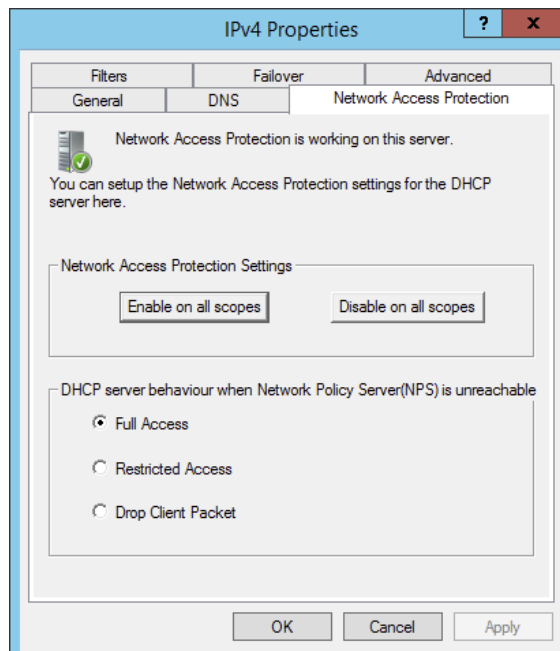
The IPv4 Properties dialog box has four tabs: General, DNS, Network Access Protection, and Advanced.

The Advanced tab, shown in Figure 2.34, contains the following configuration parameters:

- Audit Log File Path is where you enter the location for log files.
- Conflict Detection Attempts specifies how many ICMP echo requests (pings) the server sends for an address it is about to offer. The default is 0. Conflict detection is a way to verify that the DHCP server is not issuing IP addresses that are already being used on the network.

The Network Access Protection tab (see Figure 2.35) allows you to set up *Network Access Protection (NAP)*. With NAP, which is a Windows Server 2012 R2 service, an administrator can perform the following tasks:

- Carry out computer health policy validation
- Ensure ongoing compliance with health policies
- Optionally restrict the access of computers that do not meet the computer health requirements

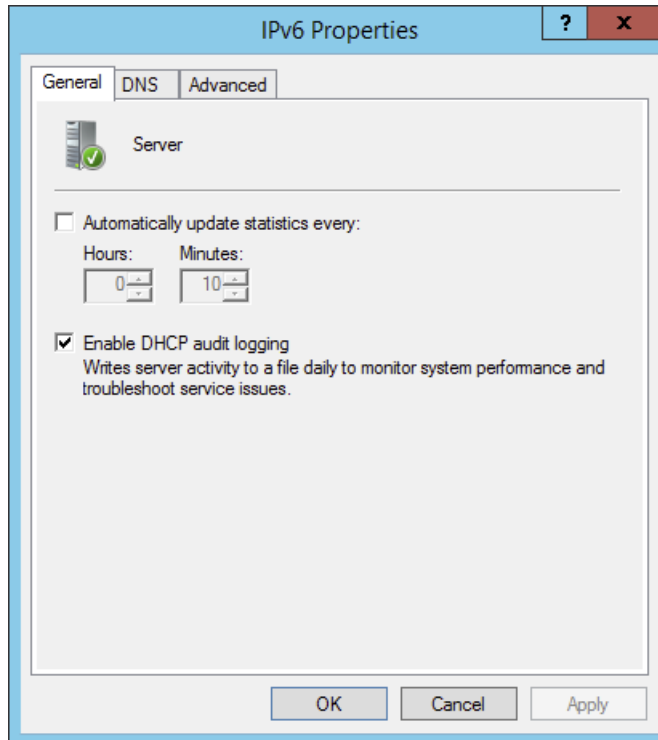
FIGURE 2.34 The Advanced tab of the IPv4 Properties dialog box for the server**FIGURE 2.35** The Network Access Protection tab of the IPv4 Properties dialog box for the server

IPv6 Server Properties

The IPv6 Properties dialog box for the server has two tabs: General and Advanced. On the General tab (see Figure 2.36), you can configure the following settings:

- Frequency with which statistics are updated
- DHCP auditing

FIGURE 2.36 Server's IPv6 Properties, General tab

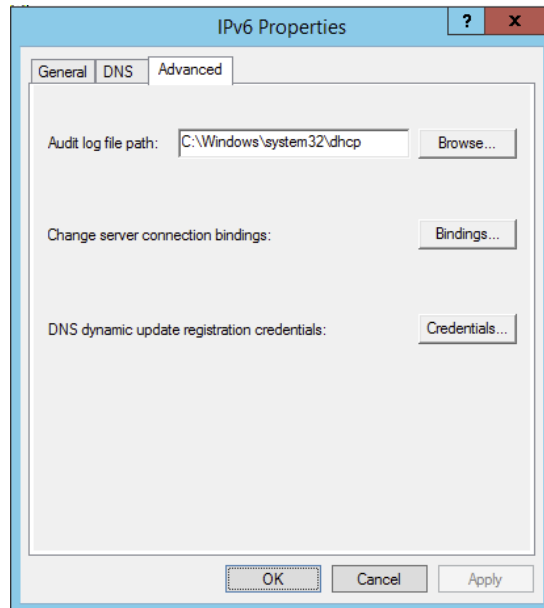


The Advanced tab (see Figure 2.37) allows you to configure the following settings:

- Database path for the audit log file path.
- Connection bindings.
- Registration credentials for dynamic DNS. The registration credential is the user account that DHCP will use to register clients with Active Directory.

Managing Reservations and Exclusions

After defining the address pool for your scope, the next step is to create reservations and exclusions, which reduce the size of the pool. In the following sections, you will learn how to add and remove exclusions and reservations.

FIGURE 2.37 Server's IPv6 Properties, Advanced tab

Adding and Removing Exclusions

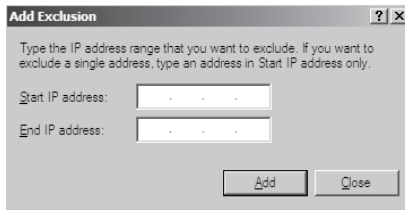
When you want to exclude an entire range of IP addresses, you need to add that range as an exclusion. Ordinarily, you'll want to do this before you enable a scope because that prevents any of the IP addresses you want excluded from being leased before you have a chance to exclude them. In fact, you can't create an exclusion that includes a leased address—you have to get rid of the lease first.

Adding an Exclusion Range

Here's how to add an exclusion range:

1. Open the DHCP snap-in and find the scope to which you want to add an exclusion (either IPv4 or IPv6).
2. Expand the scope so that you can see its Address Pool item for IPv4 or the Exclusion section for IPv6.
3. Right-click the Address Pool or Exclusion section and choose the New Exclusion Range command.
4. When the Add Exclusion dialog box appears (see Figure 2.38), enter the IP addresses you want to exclude. To exclude a single address, type it in the Start IP Address field. To exclude a range of addresses, also fill in the End IP Address field.
5. Click the Add button to add the exclusion.

When you add exclusions, they appear in the Address Pool node, which is under the Scope section for IPv4 and under the Exclusion section of IPv6.

FIGURE 2.38 Add Exclusion dialog boxes for IPv4 and IPv6


Add Exclusion [?] [X]

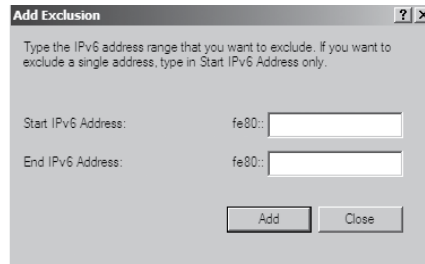
Type the IP address range that you want to exclude. If you want to exclude a single address, type an address in Start IP address only.

Start IP address: [. . .]

End IP address: [. . .]

[Add] [Close]

IPv4 Add Exclusion dialog box



Add Exclusion [?] [X]

Type the IPv6 address range that you want to exclude. If you want to exclude a single address, type in Start IPv6 Address only.

Start IPv6 Address: fe80:: []

End IPv6 Address: fe80:: []

[Add] [Close]

IPv6 Add Exclusion dialog box

Removing an Exclusion Range

To remove an exclusion, just right-click it and choose the Delete command. After confirming your command, the snap-in removes the excluded range and the addresses become immediately available for issuance.

Adding and Removing Reservations

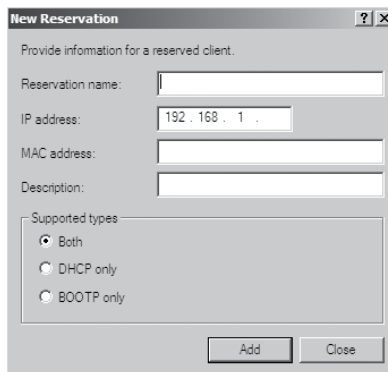
Adding a reservation is simple as long as you have the MAC address of the device for which you want to create a reservation. Because reservations belong to a single scope, you create and remove them within the Reservations node beneath each scope.

Adding a Reservation

To add a reservation, perform the following tasks:

1. Right-click the scope and select New Reservation.

This displays the New Reservation dialog box, shown in Figure 2.39.

FIGURE 2.39 New Reservation dialog boxes for IPv4 and IPv6


New Reservation [?] [X]

Provide information for a reserved client.

Reservation name: []

IP address: [192 . 168 . 1]

MAC address: []

Description: []

Supported types:

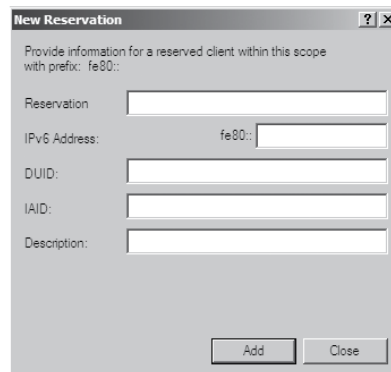
Both

DHCP only

BOOTP only

[Add] [Close]

IPv4 New Reservation dialog box



New Reservation [?] [X]

Provide information for a reserved client within this scope with prefix: fe80::

Reservation: []

IPv6 Address: fe80:: []

DUID: []

IAID: []

Description: []

[Add] [Close]

IPv6 New Reservation dialog box

2. Enter the IP address and MAC address or ID for the reservation.



To find the MAC address of the local computer, use the `ipconfig` command. To find the MAC address of a remote machine, use the `nbtstat -acomputername` command.

3. If you want, you can also enter a name and description.
4. For IPv4, in the Supported Types section, choose whether the reservation will be made by DHCP only, BOOTP only (useful for remote-access devices), or both.

Removing a Reservation

To remove a reservation, right-click it and select Delete. This removes the reservation but does nothing to the client device.



There's no way to change a reservation once it has been created. If you want to change any of the associated settings, you'll have to delete and re-create the reservation.

Setting Scope Options for IPv4

Once you've installed a server, authorized it in Active Directory, and fixed up the address pool, the next step is to set scope options that you want sent out to clients, such as router (that is, default gateway) and DNS server addresses. You must configure the options you want sent out before you activate a scope. If you don't, clients may register in the scope without getting any options, rendering them virtually useless. Thus, configure the scope options, along with the IP address and subnet mask that you configured earlier in this chapter.

In the following sections, you will learn how to configure and assign scope options on the DHCP server.

Understanding Option Assignment

You can control which DHCP options are doled out to clients in five (slightly overlapping) ways:

Predefined Options *Predefined options* are templates that are available in the Server, Scope, or Client Options dialog box.

Server Options *Server options* are assigned to all scopes and clients of a particular server. That means if there's some setting you want all clients of a DHCP server to have, no matter what scope they're in, this is where you assign it. Specific options (those that are set at the class, scope, or client level) will override server-level options. That gives you an escape

valve; it's a better idea, though, to be careful about which options you assign if your server manages multiple scopes.

Scope Options If you want a particular option value assigned only to those clients in a certain subnet, you should assign it as a *scope option*. For example, it's common to specify different routers for different physical subnets; if you have two scopes corresponding to different subnets, each scope would probably have a separate value for the router option.

Class Options You can assign different options to clients of different types, that is, *class options*. For example, Windows 2000, XP, Vista, Windows 7, Windows 8, Server 2003, Server 2003 R2, Server 2008, Server 2008 R2, and Server 2012/2012 R2 machines recognize a number of DHCP options that Windows 98, Windows NT, and Mac OS machines ignore, and vice versa. By defining a Windows 2000 or newer class (using the `ipconfig /setclassid` command you saw earlier), you could assign those options only to machines that report themselves as being in that class.

Client Options If a client is using DHCP reservations, you can assign certain options to that specific client. You attach *client options* to a particular reservation. Client options override scope, server, and class options. The only way to override a client option is to configure the client manually. The DHCP server manages client options.



Client options override class options, class options override scope options, and scope options override server options.

Assigning Options

You can use the DHCP snap-in to assign options at the scope, server, reserved address, or class level. The mechanism you use to assign these options is the same for each; the only difference is where you set the options.

When you create an option assignment, remember that it applies to all of the clients in the server or the scope from that point forward. Option assignments aren't retroactive, and they don't migrate from one scope to another.

Creating and Assigning a New Option

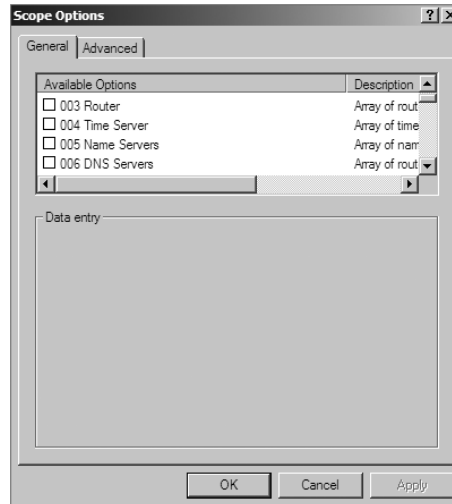
To create a new option and have it assigned, follow these steps:

1. Select the scope or server where you want the option assigned.
2. Select the corresponding Options node and choose Action > Configure Options.

To set options for a reserved client, right-click its entry in the Reservations node and select Configure Options.

Then you'll see the Scope Options dialog box (see Figure 2.40), which lists all of the options that you might want to configure.

3. To select an individual option, check the box next to it and then use the controls in the Data Entry control group to enter the value you want associated with the option.

FIGURE 2.40 The Scope Options dialog box

4. Continue to add options until you've specified all of the ones you want attached to the server or scope. Then click OK.

Configuring the DHCP Server for Classes

You saw how to assign classes to individual machines earlier in the chapter. Now you will learn how to configure the DHCP server to recognize your customized classes and configure options for them. In Exercise 2.11, you will create a new user class and configure options for the new class. Before you begin, make sure that the computers you want to use in the class have been configured with the `ipconfig /setclassid` command, as described in the section “Ipconfig Lease Options” earlier in this chapter.

EXERCISE 2.11

Configuring User Class Options

1. Open the DHCP snap-in by selecting Administrative Tools > DHCP.
2. Right-click the IPv4 item and select Define User Classes.
3. Click the Add button in the DHCP User Classes dialog box.
4. In the New Class dialog box, enter a descriptive name for the class in the Display Name field. Enter a class ID in the ID field. (Typically, you will enter the class ID in the ASCII portion of the ID field.) When you have finished, click OK.
5. The new class appears in the DHCP User Classes dialog box. Click the Close button to return to the DHCP snap-in.

6. Right-click the Scope Options node and select Configure Options.
 7. Click the Advanced tab. Select the class you defined in step 4 from the User Class pop-up menu.
 8. Configure the options you want to set for the class. Click OK when you have finished. Notice that the options you configured (and the class with which they are associated) appear in the right pane of the DHCP window.
-

About the Default Routing and Remote Access Predefined User Class

Windows Server 2012/2012 R2 includes a predefined user class called the *Default Routing and Remote Access class*. This class includes options important to clients connecting to Routing and Remote Access, notably the 051 Lease option.



Be sure to know that the 051 Lease option is included within this class and that it can be used to assign a shorter lease duration for clients connecting to Routing and Remote Access.

Activating and Deactivating Scopes

When you've completed the steps in Exercise 2.5 and you're ready to unleash your new scope so that it can be used to make client assignments, the final required step is activating the scope. When you activate a scope, you're just telling the server that it's OK to start handing out addresses from that scope's address pool. As soon as you activate a scope, addresses from its pool may be assigned to clients. Of course, this is a necessary precondition to getting any use out of your scope.

If you later want to stop using a scope, you can, but be aware that it's a permanent change. When you deactivate a scope, DHCP tells all clients registered with the scope that they need to release their leases immediately and renew them someplace else—the equivalent of a landlord who evicts tenants when the building is condemned!



Don't deactivate a scope unless you want clients to stop using it immediately.

Creating a Superscope for IPv4

A *superscope* allows the DHCP server to provide multiple logical subnet addresses to DHCP clients on a single physical network. You create superscopes with the New Superscope command, which triggers the New Superscope Wizard.



You can have only one superscope per server.

The steps in Exercise 2.12 take you through the process of creating a superscope.

EXERCISE 2.12

Creating a Superscope

1. Open the DHCP snap-in by selecting Administrative Tools > DHCP.
2. Follow the instructions in Exercise 2.10 to create two scopes: one for 192.168.0.2 through 192.168.0.127 and one for 192.168.1.12 through 192.168.1.127.
3. Right-click IPv4 and choose the New Superscope command. The New Superscope Wizard appears. Click the Next button.
4. On the Superscope Name page, name your superscope and click the Next button.
5. The Select Scopes page appears, listing all scopes on the current server. Select the two scopes you created in step 2 and then click the Next button.
6. The wizard's summary page appears. Click the Finish button to create your scope.
7. Verify that your new superscope appears in the DHCP snap-in.

Deleting a Superscope

You can delete a superscope by right-clicking it and choosing the Delete command. A superscope is just an administrative convenience, so you can safely delete one at any time—it doesn't affect the “real” scopes that make up the superscope.

Adding a Scope to a Superscope

To add a scope to an existing superscope, find the scope you want to add, right-click it, and choose Action > Add To Superscope. A dialog box appears, listing all of the superscopes known to this server. Pick the one to which you want the current scope appended and click the OK button.

Removing a Scope from a Superscope

To remove a scope from a superscope, open the superscope and right-click the target scope. The pop-up menu provides a Remove From Superscope command that will do the deed.

Activating and Deactivating Superscopes

Just as with regular scopes, you can activate and deactivate superscopes. The same restrictions and guidelines apply. You must activate a superscope before it can be used, and

you must not deactivate it until you want all of your clients to lose their existing leases and be forced to request new ones.

To activate or deactivate a superscope, right-click the superscope name, and select Activate or Deactivate, respectively, from the pop-up menu.

Creating IPv4 Multicast Scopes

Multicasting occurs when one machine communicates to a network of subscribed computers rather than specifically addressing each computer on the destination network. It's much more efficient to multicast a video or audio stream to multiple destinations than it is to unicast it to the same number of clients, and the increased demand for multicast-friendly network hardware has resulted in some head scratching about how to automate the multicast configuration.

In the following sections, you will learn about MADCAP, the protocol that controls multicasting, and about how to build and configure a multicast scope.

Understanding the Multicast Address Dynamic Client Allocation Protocol

DHCP is usually used to assign IP configuration information for *unicast* (or one-to-one) network communications. With multicast, there's a separate type of address space assigned from 224.0.0.0 through 239.255.255.255. Addresses in this space are known as *Class D addresses*, or simply *multicast addresses*. Clients can participate in a multicast just by knowing (and using) the multicast address for the content they want to receive. However, multicast clients also need to have an ordinary IP address.

How do clients know what address to use? Ordinary DHCP won't help because it's designed to assign IP addresses and option information to one client at a time. Realizing this, the Internet Engineering Task Force (IETF) defined a new protocol: *Multicast Address Dynamic Client Allocation Protocol (MADCAP)*. MADCAP provides an analog to DHCP but for multicast use. A MADCAP server issues leases for multicast addresses only. MADCAP clients can request a multicast lease when they want to participate in a multicast.

DHCP and MADCAP have some important differences. First you have to realize that the two are totally separate. A single server can be a DHCP server, a MADCAP server, or both; no implied or actual relation exists between the two. Likewise, clients can use DHCP and/or MADCAP at the same time—the only requirement is that every MADCAP client has to get a unicast IP address from somewhere.



Remember that DHCP can assign options as part of the lease process but MADCAP cannot. The only thing MADCAP does is dynamically assign multicast addresses.

Building Multicast Scopes

Most of the steps you go through when creating a multicast scope are identical to those required for an ordinary unicast scope. Exercise 2.13 highlights the differences.

EXERCISE 2.13

Creating a New Multicast Scope

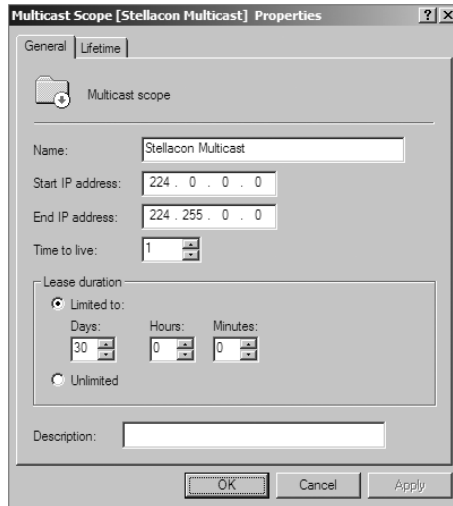
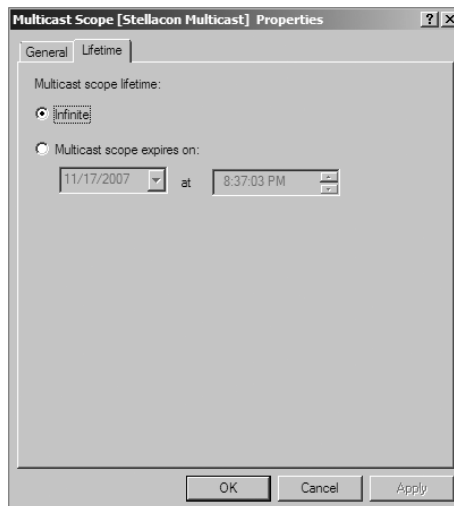
1. Open the DHCP snap-in by selecting Administrative Tools > DHCP.
 2. Right-click IPv4 and choose New Multicast Scope. The New Multicast Scope Wizard appears. Click the Next button on the welcome page.
 3. In the Multicast Scope Name page, name your multicast scope (and add a description if you'd like). Click the Next button.
 4. The IP Address Range page appears. Enter a start IP address of **224.0.0.0** and an end IP address of **224.255.0.0**. Adjust the TTL to 1 to make sure that no multicast packets escape your local network segment. Click the Next button when you're finished.
 5. The Add Exclusions page appears; click its Next button.
 6. The Lease Duration page appears. Since multicast addresses are used for video and audio, you'd ordinarily leave multicast scope assignments in place somewhat longer than you would with a regular unicast scope, so the default lease length is 30 days (instead of 8 days for a unicast scope). Click the Next button.
 7. The wizard asks you if you want to activate the scope now. Click the No radio button and then the Next button.
 8. The wizard's summary page appears; click the Finish button to create your scope.
 9. Verify that your new multicast scope appears in the DHCP snap-in.
-

Setting Multicast Scope Properties

Once you create a multicast scope, you can adjust its properties by right-clicking the scope name and selecting Properties.

The Multicast Scope Properties dialog box has two tabs. The General tab (see Figure 2.41) allows you to change the scope's name, its start and end addresses, its Time To Live (TTL) value, its lease duration, and its description—in essence, all of the settings you provided when you created it in the first place.

The Lifetime tab (see Figure 2.42) allows you to limit how long your multicast scope will be active. By default, a newly created multicast scope will live forever, but if you're creating a scope to provide MADCAP assignments for a single event (or a set of events of limited duration), you can specify an expiration time for the scope. When that time is reached, the scope disappears from the server but not before making all of its clients give up their multicast address leases. This is a nice way to make sure that the lease cleans up after itself when you're finished with it.

FIGURE 2.41 General tab of the Multicast Scope Properties dialog box**FIGURE 2.42** Lifetime tab of the Multicast Scope Properties dialog box

Integrating Dynamic DNS and IPv4 DHCP

DHCP integration with Dynamic DNS is a simple concept but powerful in action. By setting up this integration, you can pass addresses to DHCP clients while still maintaining the integrity of your DNS services.

The DNS server can be updated in two ways. One way is for the DHCP client to tell the DNS server its address. Another way is for the DHCP server to tell the DNS server when it registers a new client.

Neither of these updates will take place, however, unless you configure the DNS server to use Dynamic DNS. You can make this change in two ways:

- If you change it at the scope level, it will apply only to the scope.
- If you change it at the server level, it will apply to all scopes and superscopes served by the server.

Which of these options you choose depends on how widely you want to support Dynamic DNS; most of the sites I visit have enabled DNS updates at the server level.

To update the settings at either the server or scope level, you need to open the scope or server properties by right-clicking the appropriate object and choosing Properties. The DNS tab of the Properties dialog box (see Figure 2.43) includes the following options:

Enable DNS Dynamic Updates According To The Settings Below This check box controls whether this DHCP server will attempt to register lease information with a DNS server. It must be checked to enable Dynamic DNS.

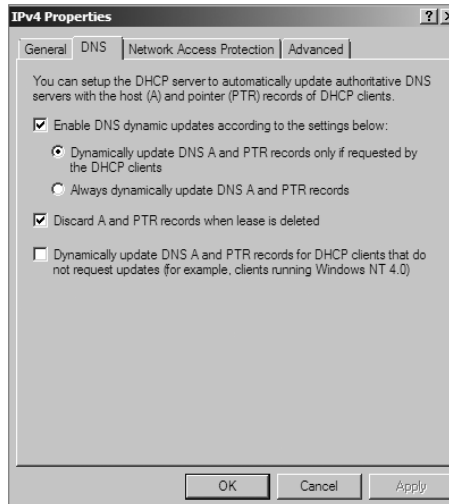
Dynamically Update DNS A And PTR Records Only If Requested By The DHCP Clients This radio button (which is on by default) tells the DHCP server to register the update only if the DHCP client asks for DNS registration. When this button is active, DHCP clients that aren't hip to DDNS won't have their DNS records updated. However, Windows 2000, XP, Vista, Windows 7, Windows 8, Server 2003/2003 R2, Server 2008/2008 R2, and Server 2012/2012 R2 DHCP clients are smart enough to ask for the updates.

Always Dynamically Update DNS A And PTR Records This radio button forces the DHCP server to register any client to which it issues a lease. This setting may add DNS registrations for DHCP-enabled devices that don't really need them, such as print servers. However, it allows other clients (such as Mac OS, Windows NT, and Linux machines) to have their DNS information automatically updated.

Discard A And PTR Records When Lease Is Deleted This check box has a long name but a simple function. When a DHCP lease expires, what should happen to the DNS registration? Obviously, it would be nice if the DNS record associated with a lease vanished when the lease expired. When this check box is checked (as it is by default), that's exactly what happens. If you uncheck this box, your DNS will contain entries for expired leases that are no longer valid. When a particular IP address is reissued on a new lease, the DNS will be updated, but in between leases you'll have incorrect data in your DNS—something that's always best to avoid.

Dynamically Update DNS A And PTR Records For DHCP Clients That Do Not Request Updates This check box lets you handle these older clients graciously by making the updates using a separate mechanism.

In Exercise 2.14, you will enable a scope to participate in Dynamic DNS updates.

FIGURE 2.43 DNS tab of the scope's IPv4 Properties dialog box**EXERCISE 2.14****Enabling DHCP-DNS Integration**

1. Open the DHCP snap-in by selecting Administrative Tools > DHCP.
2. Right-click the IPv4 item and select Properties.
3. The Server Properties dialog box appears. Click the DNS tab.
4. Verify that the check box labeled Enable DNS Dynamic Updates According To The Settings Below is checked and verify that the radio button labeled Dynamically Update DNS A And PTR Records Only If Requested By The DHCP Clients is selected.
5. Verify that the check box labeled Discard A And PTR Records When Lease Is Deleted is checked. If not, then check it.
6. Click the OK button to apply your changes and close the Server Properties dialog box.

Using Multiple DHCP Servers

DHCP can become a single point of failure within a network if there is only one DHCP server. If that server becomes unavailable, clients will not be able to obtain new leases or renew existing leases. For this reason, it is recommended that you have more than one DHCP server in the network. However, more than one DHCP server can create problems if they both are configured to use the same scope or set of addresses. Microsoft recommends the 80/20 rule for redundancy of DHCP services in a network.

Implementing the 80/20 rule calls for one DHCP server to make approximately 80 percent of the addresses for a given subnet available through DHCP while another server makes the remaining 20 percent of the addresses available. For example, with a /24 network of 254 addresses, say 192.168.1.1 to 192.168.1.254, you might have Server 1 offer 192.168.1.10 to 192.168.1.210 while Server 2 offers 192.168.1.211 to 192.168.254.

DHCP Load Sharing

Load sharing is the normal default way that you use multiple DHCP servers (as explained earlier). Both servers cover the same subnets (remember that a DHCP server can handle multiple subnets at the same time) simultaneously, and both servers assign IP addresses and options to clients on the assigned subnets. The client requests are load balanced and shared between the two servers.

This is a good option for a company that has multiple DHCP servers in the same physical location. The DHCP servers are set up in a failover relationship at the same site, and both servers respond to all DHCP client requests from the subnets to which they are associated. The DHCP server administrator can set the load distribution ratio between the multiple DHCP servers.

DHCP Hot Standby

When thinking of a DHCP hot standby setup, think of the old server failover cluster. You have two servers where one server does all of the work and the other server is a standby server in the event that the first server crashes or goes down.

In a DHCP hot standby situation, the two DHCP servers operate in a failover relationship where one server acts as an active server and is responsible for leasing IP addresses to all clients in a scope or subnet. The secondary DHCP server assumes the standby role, and it is ready to go in the event that the primary DHCP server becomes unavailable. If the primary server becomes unavailable, the secondary DHCP server is given the role of the primary DHCP server and takes over all the responsibilities of the primary DHCP server.

This failover situation is best suited to DHCP deployments where a company has DHCP servers in multiple locations.



To learn more about DHCP failover situations, please visit Microsoft at <http://technet.microsoft.com/en-us/library/hh831385.aspx>. Microsoft has been known for taking questions right off its websites, and this website is the perfect solution for doing this.

Working with the DHCP Database Files

DHCP uses a set of database files to maintain its knowledge of scopes, superscopes, and client leases. These files, which live in the *systemroot\System32\DHCP* folder, are always

open when the DHCP service is running. DHCP servers use Joint Engine Technology (JET) databases to maintain their records.



You shouldn't modify or alter the DHCP database files when the service is running.

The primary database file is `dhcp.mdb`—it has all of the scope data in it. The following files are also part of the DHCP database:

Dhcp.tmp This is a backup copy of the database file created during reindexing of the database. You normally won't see this file, but if the service fails during reindexing, it may not remove the file when it should.

J50.log This file (plus a number of files named `J50xxxxx.log`, where `xxxxxx` stands for 00001, 00002, 00003, and so on) is a log file that stores changes before they're written to the database. The DHCP database engine can recover some changes from these files when it restarts.

J50.chk This is a checkpoint file that tells the DHCP engine which log files it still needs to recover.

In the following sections, you will see how to manipulate the DHCP database files.

Removing the Database Files

If you're convinced that your database is corrupt because the lease information that you see doesn't match what's on the network, the easiest repair mechanism is to remove the database files and start over with an empty database.



If you think the database is corrupt because the DHCP service fails at startup, you should check the event log.

To start over, follow these steps:

1. Stop the DHCP service by typing **net stop dhcpserver** at the command prompt.
2. Remove all of the files from the `systemroot\system32\DHCP` folder.
3. Restart the service (at command prompt type **net start dhcpserver**).
4. Reconcile the scope.

Changing the Database Backup Interval

By default, the DHCP service backs up its databases every 60 minutes. You can adjust this setting by editing the Backup Interval value under `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\DHCPserver\Parameters`. This allows you to make backups either more frequently (if your database changes a lot or if you seem to have ongoing corruption problems) or less often (if everything seems to be on an even keel).

Moving the DHCP Database Files

You may find that you need to dismantle or change the role of your DHCP server and offload the DHCP functions to another computer. Rather than spend the time re-creating the DHCP database on the new machine by hand, you can copy the database files and use them directly. This is especially helpful if you have a complicated DHCP database with lots of reservations and option assignments.

By copying the files, you also minimize the amount of human error that could be introduced by reentering the information by hand.

Compacting the DHCP Database Files

There may be a time when you need to compact the DHCP database. Microsoft has a utility called `jetpack.exe` that allows you to compact the JET database. Microsoft JET databases are used for WINS and DHCP databases. If you wanted to use the `jetpack` command, the proper syntax is

```
JETPACK.EXE <database name><temp database name>
```

After you compact the database, you rename the temp database to `dhcp.mdb`.

Summary

DNS was designed to be a robust, scalable, and high-performance system for resolving friendly names to TCP/IP host addresses. This chapter presented an overview of the basics of DNS and how DNS names are generated. We then looked at the many new features available in the Microsoft Windows Server 2012 R2 version of DNS, and we focused on how to install, configure, and manage the necessary services. Microsoft's DNS is based on a widely accepted set of industry standards. Because of this, Microsoft's DNS can work with both Windows- and non-Windows-based networks.

This chapter also covered the DHCP lease process as it relates to TCP/IP configuration information for clients. The following stages were covered: IP discovery, IP lease offer, IP lease selection, and IP lease acknowledgment. You learned how to install and configure the DHCP server on Windows Server 2012 R2 and how to create and manage DHCP scopes and scope options. I also discussed the authorization of DHCP servers within Active Directory and scopes for IPv4 and IPv6 and showed how to create them. Finally, I covered superscopes as well as managing client leases with the options therein.

Exam Essentials

Understand the purpose of DNS. DNS is a standard set of protocols that defines a mechanism for querying and updating address information in the database, a mechanism for replicating the information in the database among servers, and a schema of the database.

Understand the different parts of the DNS database. The SOA record defines the general parameters for the DNS zone, including who is the authoritative server. NS records list the name servers for a domain; they allow other name servers to look up names in your domain. A host record (also called an address record or an A record) statically associates a host's name with its IP addresses. Pointer records (PTRs) map an IP address to a hostname, making it possible to do reverse lookups. Alias records allow you to use more than one name to point to a single host. The MX record tells you which servers can accept mail bound for a domain. SRV records tie together the location of a service (like a domain controller) with information about how to contact the service.

Know how DNS resolves names. With iterative queries, a client asks the DNS server for an answer, and the client, or resolver, returns the best kind of answer it has available. In a recursive query, the client sends a query to one name server, asking it to respond either with the requested answer or with an error. The error states either that the server can't come up with the right answer or that the domain name doesn't exist. With inverse queries, instead of supplying a name and then asking for an IP address, the client first provides the IP address and then asks for the name.

Understand the differences among DNS servers, clients, and resolvers. Any computer providing domain name services is a DNS server. A DNS client is any machine issuing queries to a DNS server. A resolver handles the process of mapping a symbolic name to an actual network address.

Know how to install and configure DNS. DNS can be installed before, during, or after installing the Active Directory service. When you install the DNS server, the DNS snap-in is installed too. Configuring a DNS server ranges from easy to difficult, depending on what you're trying to make it do. In the simplest configuration, for a caching-only server, you don't have to do anything except to make sure the server's root hints are set correctly. You can also configure a root server, a normal forward lookup server, and a reverse lookup server.

Know how to create new forward and reverse lookup zones. You can use the New Zone Wizard to create a new forward or reverse lookup zone. The process is basically the same for both types, but the specific steps and wizard pages differ somewhat. The wizard walks you through the steps, such as specifying a name for the zone (in the case of forward lookup zones) or the network ID portion of the network that the zone covers (in the case of reverse lookup zones).

Know how to configure zones for dynamic updates. The DNS service allows dynamic updates to be enabled or disabled on a per-zone basis at each server. This is easily done in the DNS snap-in.

Know how to delegate zones for DNS. DNS provides the ability to divide the namespace into one or more zones; these can then be stored, distributed, and replicated to other DNS servers. When delegating zones within your namespace, be aware that for each new zone you create, you need delegation records in other zones that point to the authoritative DNS servers for the new zone.

Understand the tools that are available for monitoring and troubleshooting DNS. You can use the DNS snap-in to do some basic server testing and monitoring. More important, you use the snap-in to monitor and set logging options. Windows Server 2012 R2 automatically logs DNS events in the event log under a distinct DNS server heading. Nslookup offers the ability to perform query testing of DNS servers and to obtain detailed responses at the command prompt. 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. Finally, you can configure the DNS server to create a log file that records queries, notification messages, dynamic updates, and various other bits of DNS information.

Know how to install and authorize a DHCP server. You install the DHCP service using the Add/Remove Windows Components Wizard. You authorize the DHCP server using the DHCP snap-in. When you authorize a server, you're actually adding its IP address to the Active Directory object that contains a list of the IP addresses of all authorized DHCP servers.

Know how to create a DHCP scope. You use the New Scope Wizard to create a new scope for both IPv4 and IPv6. Before you start, you'll need to know the IP address range for the scope you want to create; which IP addresses, if any, you want to exclude from the address pool; which IP addresses, if any, you want to reserve; and the values for the DHCP options you want to set, if any.

Understand how relay agents help with multiple physical network segments. A question about relay agents on the exam may appear to be a DHCP-related question. Relay agents assist DHCP message propagation across network or router boundaries where such messages ordinarily wouldn't pass.

Understand the difference between exclusions and reservations. When you want to exclude an entire range of IP addresses, you need to add that range as an exclusion. Any IP addresses within the range for which you want a permanent DHCP lease are known as reservations. Remember that exclusions are TCP/IP numbers in a pool that do not get issued and reservations are numbers in a TCP/IP pool that get issued only to the same client each time.

Review Questions

1. You are the network administrator for the ABC Company. Your network consists of two DNS servers named *DNS1* and *DNS2*. The users who are configured to use *DNS2* complain because they are unable to connect to Internet websites. The following table shows the configuration of both servers:

DNS1	DNS2
_msdcs.abc.comabc.com	.(root)_msdcs.abc.comabc.com

The users connected to *DNS2* need to be able to access the Internet. What needs to be done?

- A. Build a new Active Directory Integrated zone on *DNS2*.
 - B. Delete the *.(root)* zone from *DNS2* and configure conditional forwarding on *DNS2*.
 - C. Delete the current *cache.dns* file.
 - D. Update your *cache.dns* file and root hints.
2. You are the network administrator for a large company that has one main site and one branch office. Your company has a single Active Directory forest, *ABC.com*. You have a single domain controller (*ServerA*) in the main site that has the DNS role installed. *ServerA* is configured as a primary DNS zone. You have decided to place a domain controller (*ServerB*) in the remote site and implement the DNS role on that server. You want to configure DNS so that, if the WAN link fails, users in both sites can still update records and resolve any DNS queries. How should you configure the DNS servers?
 - A. Configure *ServerB* as a secondary DNS server. Set replication to occur every five minutes.
 - B. Configure *ServerB* as a stub zone.
 - C. Configure *ServerB* as an Active Directory Integrated zone and convert *ServerA* to an Active Directory Integrated zone.
 - D. Convert *ServerA* to an Active Directory Integrated zone and configure *ServerB* as a secondary zone.
 3. You are the network administrator for a midsize computer company. You have a single Active Directory forest, and your DNS servers are configured as Active Directory Integrated zones. When you look at the DNS records in Active Directory, you notice that there are many records for computers that do not exist on your domain. You want to make sure that only domain computers register with your DNS servers. What should you do to resolve this issue?
 - A. Set dynamic updates to None.
 - B. Set dynamic updates to Nonsecure And Secure.
 - C. Set dynamic updates to Domain Users Only.
 - D. Set dynamic updates to Secure Only.

4. Your company consists of a single Active Directory forest. You have a Windows Server 2012 R2 domain controller that also has the DNS role installed. You also have a Unix-based DNS server at the same location. You need to configure your Windows DNS server to allow zone transfers to the Unix-based DNS server. What should you do?
 - A. Enable BIND secondaries.
 - B. Configure the Unix machine as a stub zone.
 - C. Convert the DNS server to Active Directory Integrated.
 - D. Configure the Microsoft DNS server to forward all requests to the Unix DNS server.

5. You are the network administrator for Stellacon Corporation. Stellacon has two trees in its Active Directory forest, stellacon.com and abc.com. Company policy does not allow DNS zone transfers between the two trees. You need to make sure that when anyone in abc.com tries to access the stellacon.com domain, all names are resolved from the stellacon.com DNS server. What should you do?
 - A. Create a new secondary zone in abc.com for stellacon.com.
 - B. Configure conditional forwarding on the abc.com DNS server for stellacon.com.
 - C. Create a new secondary zone in stellacon.com for abc.com.
 - D. Configure conditional forwarding on the stellacon.com DNS server for abc.com.

6. You are the network administrator for your organization. A new company policy states that all inbound DNS queries need to be recorded. What can you do to verify that the IT department is compliant with this new policy?
 - A. Enable Server Auditing – Object Access.
 - B. Enable DNS debug logging.
 - C. Enable server database query logging.
 - D. Enable DNS Auditing – Object Access.

7. You are the network administrator for a small company with two DNS servers: DNS1 and DNS2. Both DNS servers reside on domain controllers. DNS1 is set up as a standard primary zone, and DNS2 is set up as a secondary zone. A new security policy was written stating that all DNS zone transfers must be encrypted. How can you implement the new security policy?
 - A. Enable the Secure Only setting on DNS1.
 - B. Enable the Secure Only setting on DNS2.
 - C. Configure Secure Only on the Zone Transfers tab for both servers.
 - D. Delete the secondary zone on DNS2. Convert both DNS servers to use Active Directory Integrated zones.

8. You are responsible for DNS in your organization. You look at the DNS database and see a large number of older records on the server. These records are no longer valid. What should you do?
 - A. In the zone properties, enable Zone Aging and Scavenging.
 - B. In the server properties, enable Zone Aging and Scavenging.
 - C. Manually delete all of the old records.
 - D. Set Dynamic Updates to None.

9. Your IT team has been informed by the compliance team that it needs copies of the DNS Active Directory Integrated zones for security reasons. You need to give the Compliance department a copy of the DNS zone. How should you accomplish this goal?
 - A. Run `dnscmd /zonecopy`.
 - B. Run `dnscmd /zoneinfo`.
 - C. Run `dnscmd /zoneexport`.
 - D. Run `dnscmd /zonefile`.

10. You are the network administrator for a Windows Server 2012 R2 network. You have multiple remote locations connected to your main office by slow satellite links. You want to install DNS into these offices so that clients can locate authoritative DNS servers in the main location. What type of DNS servers should be installed in the remote locations?
 - A. Primary DNS zones
 - B. Secondary DNS zones
 - C. Active Directory Integrated zones
 - D. Stub zones

