



Configuring Dynamic DNS Update

Dynamic DNS update integrates DNS with DHCP. The two protocols are complementary—DHCP centralizes and automates IP address allocation, while DNS automatically records the association between assigned addresses and host names. When you use DHCP and dynamic DNS update, this configures a host automatically for network access whenever it attaches to the IP network. You can locate and reach the host using its permanent, unique DNS host name. Mobile hosts, for example, can move freely without user or administrator intervention.

This chapter explains how to use dynamic DNS update with Cisco CNS Network Registrar servers, and the special relevance to Windows 2000 client systems.

Dynamic DNS Update Process

To configure dynamic DNS update, you need to configure at least a DHCP scope, and supply host names or request that Network Registrar generate them. You can only update a primary DNS server that supports dynamic DNS update.

This is the process to configure dynamic DNS update:

1. Configure the DHCP scopes for dynamic DNS update.
2. Configure the DNS zones to accept dynamic DNS updates.
3. Make configuration adjustments to allow for Windows 2000 clients, if necessary, such as for dual zone updates.
4. Reload the DNS and DHCP servers.

The remainder of this chapter describes each step in detail.

Dynamic DNS Update Configuration Considerations

Consider these two issues when configuring dynamic DNS updates:

- For security purposes, Network Registrar's dynamic DNS update process does not modify or delete a name an administrator manually enters in the DNS database.
- If you enable dynamic DNS update for large deployments, divide primary DNS and DHCP servers across multiple clusters. Dynamic DNS update generates an additional load on the servers.

Configuring Dynamic DNS for Scopes

For dynamic DNS update, DHCP uses the host name passed to the server in the *host-name* option (12). The name is set on the client's computer.



Note

With Microsoft clients, the host name appears in the Control Panel/Network/Identification dialog box, not the Control Panel/Network/Protocols/Microsoft TCP/IP Properties/DNS dialog box.)

Step 1 Create a DHCP scope.

Step 2 Enable or set these scope attributes:

- a. Enable *dynamic-dns*.
- b. Set *dns-zone-name* to the forward zone to which a client's host name (A record) should be added.
- c. Set *dns-server-addr* to the IP address of the primary DNS server for the zone into which the server should add A records.
- d. Set *dns-reverse-zone-name* to the reverse (in.addr.arpa) zone to be updated with PTR and TXT records. This is optional and defaults to the forward zone address. If unset and the DHCP server's *synthesize-reverse-zone* attribute is enabled, the server makes up a reverse zone name based on the address of each lease, the scope's subnet number, and the scope's *dns-host-bytes* attribute value.
- e. Set *dns-rev-server-addr* to the address of the primary DNS server for the zone into which the server should add PTR records.
- f. If necessary, set the *synthetic-name-stem* value and enable *synthesize-name* on the scope.

You can set the stem of the default host name to use if clients do not supply host names by using the *synthetic-name-stem* scope attribute. The default synthetic name stem is *dhcp*. You can then enable the *enable synthesize-name* scope attribute to trigger the DHCP server to synthesize unique names for clients based on the value of the *synthetic-name-stem*.

- g. If necessary, enable the *use-dns-update-prereqs* server attribute.

By default (and as recommended), the DHCP server uses prerequisites in its DNS update messages when it performs DNS updates on behalf of clients. When you disable the *use-dns-update-prereqs* attribute, the server does not include prerequisites when performing updates for leases from this scope. Without prerequisites, the server associates the last client who uses a given domain name with that name, even if another client was already associated with it.

Step 3 Reload the DHCP server.

Enabling Dynamic Update for Zones

You must enable dynamic DNS updates for the DNS server by enabling it for each zone.

Step 1 Create a primary forward zone.

Step 2 Enable the following attributes for the zone:

- *dynamic*
- *update-acl* (see the [“Transaction Security and Access Control Lists”](#) section on page 15-3)

- Step 3** Create a primary reverse zone with the same settings.
- Step 4** If necessary, enable the DNS server attribute *update-relax-zone-name*—You can choose to relax the restriction imposed by RFC 2136 on the dynamic update zone name record that requires it to be an actual zone name. You may want to do this to construct host names so that each scope generates names with different prefixes, but where those prefixes are not necessarily all configured as independent zones. With the restriction normally in effect (the default), the server has no way of identifying which actual zones these addresses are in and would create a packet that the DNS server considers invalid.
- With the restriction removed, the name can then be any name in an authoritative zone. For example, DNS has a forward zone configured as example.com. You then create three scopes (net1, net2, and net3) and identify their forward zones as net1.example.com, net2.example.com, and net3.example.com.
- Step 5** Reload the DNS server.
-

Transaction Security and Access Control Lists

Transaction Signatures (TSIG) authenticate DNS packets. They enable the DNS server to ensure that the contents of a message were not tampered with and verify that the message is coming from a trusted source. Access control lists (ACL) authorize a client. They enable the server to allow or disallow the request or action defined in a packet.

TSIG is supported only as of Network Registrar Release 6.0, and in that release only for dynamic DNS updates. As of Network Registrar Release 6.1, support was added for queries and zone transfers.

ACLs are authorization lists. Earlier versions of Network Registrar had simple IP address-based ACLs. As of Network Registrar Release 6.0, the lists are much more sophisticated and have the ability to include TSIG keys.

For each specific action (such as a DNS query, update, or zone transfer) that is to be secured, an ACL must be set up to provide permission control, because TSIG is only an authentication mechanism. TSIG processing is only performed on messages that contain TSIG information. A message that does not contain, or is stripped of, this information bypasses the authentication process.

For a totally secure solution, messages should be authorized by the same key as that with which they are authenticated. For example, if the DHCP server is configured to use TSIG for dynamic DNS updates and the same TSIG key is included in the ACL for the zones to be updated, then any packet that does not contain TSIG information fails the authorization step. This secures the update transactions and that messages are both authenticated and authorized before making zone changes.

Access Control Lists

You assign ACLs on the DNS server or zone level. ACLs can include one or more of these elements:

- IP address—In dotted decimal notation, such as 192.168.1.2.
- Network address—In dotted decimal and slash notation, such as 192.168.0.0/24. In this example, only hosts on that network can update the DNS server.
- Another ACL—That ACL must be predefined. You cannot delete the latter ACL, because it is included as a value to the ACL being defined.
- Transaction Signature (TSIG) key—The value must be in the form **key value**, with the keyword **key** followed by the secret value. Because of space characters, the entire list must be enclosed in double quotes. For TSIG keys, see the [“Transaction Security” section on page 15-4](#).

You assign each ACL a unique name. However, the following ACL names have special meanings and you cannot use them for regular ACL names:

- **any**—Anyone can perform a certain action
- **none**—No one can perform a certain action
- **localhost**—Any of the local host addresses can perform a certain action
- **localnets**—Any of the local networks can perform a certain action

In the local cluster Web UI, on the Primary Navigation bar, click **Administration**. On the Secondary Navigation bar, click **ACLs**. Add an ACL name and match list. Note that a **key value** pair should not be in quotes.

In the CLI, use the **acl** command, which takes a name and one or more ACL elements. The ACL list is comma-separated, with double quotes surrounding it if there is a space character.

For example, the following commands create three ACLs. The first is a key with a value, the second is for a network, and the third points to the first ACL. Including the **!** symbol before a value negates that value, so that you can exclude it in a series of values:

```
nrcmd> acl sec-acl create "key h-a.h-b.example.com."
nrcmd> acl dyn-update-acl create "192.168.2.0/24,!192.168.2.13"
nrcmd> acl main-acl create sec-acl
```



Tip

You can use the 0.0.0.0/0 network specification in your ACL to allow everyone to update a zone.

Configuring DNS Servers or Zones for Access Control Lists

To configure ACLs for the DNS server or zones, use the *update-acl* attribute. An ACL set at the zone level overrides the server value. Including the **!** symbol before a value negates that value, so that you can exclude it. ACLs are ORed lists and are searched from the left to right. If you include negated elements in the list, they should come before the other ones to assure that the negated ones are handled first.

Transaction Security

Transaction Signatures (TSIG) enable the DNS server to authenticate each message that it receives. Communication between servers is not encrypted but it becomes digitally signed, which allows validation of the authenticity of the data and the source of the packet.

When you configure the Network Registrar DHCP server to use TSIG for dynamic DNS updates, the server appends a TSIG resource record to the messages. Part of the TSIG record is a digital signature.

When the DNS server receives a message, it looks for the TSIG record. If it finds one, it first verifies that the key name in it is one of the keys it recognizes. It then verifies that the time stamp in the update is reasonable (to help fight against traffic replay attacks). Finally, the server looks up the key's shared secret that was sent in the packet and calculates its own signature. If the resulting calculated signature matches the signature included in the packet, then the contents are considered to be authentic.

In Network Registrar, the TSIG key name is associated with a shared secret value. The key name should reflect the name of the hosts using this key. Entry of a name also requires a shared secret.

Generating Keys

It is recommended that you use the Network Registrar **cnr_keygen** utility to generate TSIG keys so that you add them or import them using the **import keys** command.

Execute the **cnr_keygen** key generator utility from a DOS prompt, or a Solaris or Linux shell:

- On Windows, the utility is in the *install-path*\bin folder.
- On Solaris and Linux, the utility is in the *install-path*/usrbin directory.

An example of its usage (on Solaris and Linux) is:

```
> /opt/nwreg2/local/usrbin/cnr_keygen -n a.b.example.com. -a hmac-md5 -t TSIG -b 16
-s 300
    key "a.b." {
    algorithm hmac-md5;
    secret "xGVCsFZ0/6e0N97HGF50eg==";
    # cnr-time-skew 300;
    # cnr-security-type TSIG;
};
```

The only required input is the key name. The options are described in [Table 15-1](#).

Table 15-1 Options for the cnr_keygen Utility

Option	Description
-n name	Key name. Required. The maximum length is 255 bytes.
-a hmac-md5	Algorithm. Optional. Only hmac-md5 is currently supported.
-b bytes	Byte size of the secret. Optional. The default is 16 bytes. The valid range is 1 through 64 bytes.
-s skew	Time skew for the key, in seconds. This is the maximum difference between the time stamp in packets signed with this key and the local system time. Optional. The default is five minutes. The range is one second through one hour.
-t tsig	Type of security used. Optional. Only TSIG is currently supported.
-h	Help. Optional. Displays the syntax and options of the utility.
-v	Version. Optional. Displays the version of the utility.

The resulting secret is base64-encoded as a random string.

You can also redirect the output to a file if you use the **>** or **>>** indicators at the end of the command line. The **>** writes or overwrites a given file, while the **>>** appends to an existing file. For example:

```
> /opt/nwreg2/local/usrbin/cnr_keygen -n example.com > keyfile.txt
> /opt/nwreg2/local/usrbin/cnr_keygen -n example.com >> addtokeyfile.txt
```

You can then import the key file into Network Registrar using the CLI to generate the keys in the file. The key import can generate as many keys as it finds in the import file. For example:

```
nrcmd> import keys keyfile.txt
```

Considerations for Managing Keys

If you generate your own keys, you must enter them as a base64-encoded string. This means that the only characters allowed are those in the base64 alphabet and the pad character (=). Entering a nonbase64-encoded string results in an error message. Here are some other suggestions:

- Do not add or modify keys using batch commands.
- Change shared secrets frequently; every two months is recommended. Note that Network Registrar does not explicitly enforce this.
- The shared secret length should be at least as long as the keyed message digest (HMAC-MD5 is 16 bytes). Note that Network Registrar does not explicitly enforce this and only checks that the shared secret is a valid base64-encoded string, but it is the policy recommended by RFC 2845.

Adding Supporting TSIG Attributes


Add a key name, an optional time skew value, and a secret value. Then, for the DHCP server or scope, set the following attribute values:

- *dynamic-dns-tsig*—Decide if you want to set this for forward or reverse zones, or both. On the scope level, you can set this to *enable-fwd-rev*, *disable-fwd-rev*, *enable-fwd-only*, *enable-rev-only*, or *use-server-settings*. On the server level, you can set this to *enable-fwd-rev*, *disable-fwd-rev*, *enable-fwd-only*, or *enable-rev-only*.
- *dynamic-tsig-fwd-key*—Key for forward zones only.
- *dynamic-tsig-rev-key*—Key for reverse zones only.

Confirming Dynamic Records

The Network Registrar DHCP server stores all pending DNS update data on disk. If the DHCP server cannot communicate with a DNS server, it periodically tests for re-established communication and submits all pending updates. This test typically occurs every 40 seconds.

To confirm dynamic DNS records in the local cluster Web UI:

-
- | | |
|---------------|---|
| Step 1 | On the Primary Navigation bar, click Zone . |
| Step 2 | On the Secondary Navigation bar, click Zones . |
| Step 3 | Click the View icon () in the Active Server RRs column to open the List/Add DNS Server Resource Records for Zone page. |
-

To confirm that DNS update is working in the CLI, use the `zone name listRR dynamic` command.

Change Sets and Checkpointing

Network Registrar maintains a change set database to capture any dynamic changes to resource records in zones in a performance-enhanced way. It calls the change set database when a server responds to external dynamic DNS updates, full or incremental zone transfers, zone checkpointing, stale record

scavenging, or incremental or full configuration database reloads. For scavenging, see the “[Scavenging Dynamic Records](#)” section on page 15-7. The change set database is backed up during the usual **medshadow** backup.

A change set can range from a single resource record to many records. Dynamic DNS updates or incremental zone changes first go to a change set update buffer that the server manages. Every ten-millisecond transaction interval (or 50000 change sets), the data is flushed from this buffer to a transaction log in the database. The change sets accumulate in the transaction log until they are committed to the database history file, every 20-second checkpoint interval. Each log file can be up to 10 MB in size and the files are trimmed every ten minutes after being committed.

The database file has one history list for each of the server’s zones, and each entry in the list represents a change for that zone. (In the case of a full zone transfer, the history shows only that the transfer occurred.) This committed data becomes the building blocks for outgoing incremental zone transfers. Every 30 seconds the database checks a certain small percent of the history for possible trimming in case it gets too big. As soon as the history reaches 2000 change sets, zone checkpointing occurs, and the database trims the remaining history, except a maximum 400 change sets that it always keeps. (Modify these default settings only under guidance; see the “[Troubleshooting Dynamic DNS Update](#)” section on page 15-9.)

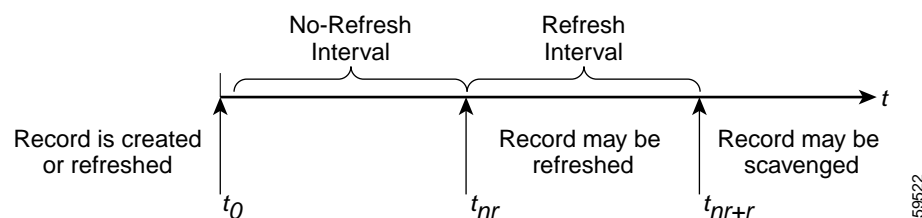
A zone checkpoint is different from a change set checkpoint. Zone checkpointing saves a snapshot of the `auth.db` file to a flat data file and updates it with the newest change sets. In addition to occurring every 2000 change sets, a zone checkpoint occurs by default every three hours. You can adjust this interval, from between one and 168 hours, using the `zone checkpoint-interval` attribute. You would increase the zone checkpoint interval to incur less runtime overhead, at the expense of the change set database retaining more history records. In the CLI, you can force zone checkpointing by using the **zone name chkpt** command, and dump a more humanly readable form of the zone checkpoint file by using the **zone name dumpchkpt** command.

Scavenging Dynamic Records

Microsoft Windows 2000 DNS clients that get DHCP leases can update (refresh) their Address (A) records directly with the DNS server. Because many of these clients are mobile laptops that are not permanently connected, some A records may become obsolete over time. The Windows 2000 DNS server scavenges and purges these primary zone records periodically. Network Registrar provides a similar feature that you can use to configure to periodically purge stale records.

Scavenging is normally disabled by default, but you should enable it for zones that exclusively contain Windows 2000 clients. Zones are configured with `no-refresh` and `refresh` intervals. Records expire once the record ages past its initial creation date plus these two intervals. [Figure 15-1](#) shows the intervals in the scavenging time line.

Figure 15-1 Address Record Scavenging Time Line Intervals



The Network Registrar process is:

1. When the client updates the DNS server with a new A record, this record gets a timestamp, or if the client refreshes its A record, this may update the timestamp (“Record is created or refreshed”).
2. During a no-refresh interval (a default of seven days), if the client keeps sending the same record without an address change, this does not update the record’s timestamp.
3. Once the record ages past the no-refresh interval, it enters the refresh interval (also a default of seven days), during which time dynamic updates refresh the timestamp and put the record back into the no-refresh interval.
4. A record that ages past the refresh interval is available for scavenging when it reaches the scavenge interval.

The following zone attributes affect scavenging:

- *scvg-interval*—Period during which the DNS server checks for stale records in a zone. The default is seven days and the value can range from one hour to 365 days. You can set this at the server and zone levels, although the zone setting overrides the server setting.
- *scvg-no-refresh-interval*—Interval during which actions, such as dynamic or prerequisite-only dynamic updates, do not update the record timestamp. The default is seven days and the value can range from one hour to 365 days. The zone setting overrides the server setting.
- *scvg-refresh-interval*—Interval during which dynamic updates update the record timestamp. After both the no-refresh and refresh intervals expire, the record is a candidate for scavenging. The default is seven days and the value can range from one hour to 365 days. The zone setting overrides the server setting.
- *scvg-ignore-restart-interval*—Ensures that the server does not reset the scavenging time with every server restart. Within this interval, the Network Registrar ignores the time between when the server went down and was restarted, which is usually fairly short. The interval defaults to two hours, but has a maximum value of one day. With any time longer than that set, Network Registrar recalculates the scavenging period to allow for records to be updated that could not do so while the server was stopped. The zone setting overrides the server setting.

Enable scavenging only for zones where a Network Registrar DNS server receives updates exclusively from Windows 2000 clients (or those known to do automatic periodic DNS updates) only. Set the attributes listed above. The Network Registrar scavenging manager starts at server startup. It reports records purged through scavenging to the change set database. Network Registrar also notifies secondary zones by way of zone transfers of any records scavenged from the primary zone. In cases where you create a zone that has scavenging disabled (the records do not have a timestamp) and then subsequently enable it, Network Registrar uses a proxy timestamp as a default timestamp for each record.

In the CLI, you can use the **getScavengeStartTime** action on a zone to find out the next time scavenging is scheduled to start on the zone. If you want to force scavenging at any time, use the **dns scavenge** command for all zones that have scavenging enabled, or the **zone name scavenge** command for a specific zone that has it enabled.

You can monitor scavenging activity using one or more of the log settings **scavenge**, **scavenge-details**, **ddns-refreshes**, and **ddns-refreshes-details**.

Troubleshooting Dynamic DNS Update

To verify if dynamic updates happened correctly to your DNS server, use the **nslookup** tool to do a reverse lookup:

```
$ nslookup
Default Server: server2.example.com
Address: 192.168.1.2
> leasehost1.example.com
Server: server2.example.com
Address: 192.168.1.100
> set type=ptr
> 192.168.1.100
Server: server2.example.com
Address: 192.168.1.100

100.40.168.192.in-addr.arpa name = leasehost1.example.com
40.168,192.in-addr.arpa nameserver = server2.example.com
```

You can monitor dynamic DNS updates on the DNS server by setting the *log-settings* attribute to *ddns*, or show even more details by setting it to *ddns-details*.

Configuring DNS Updates for Windows 2000 Clients

The Microsoft Windows 2000 operating system relies heavily on DNS and, to a lesser extent, DHCP. This reliance represents a significant change from previous versions of Windows and requires careful preparation on the part of network administrators prior to wide-scale Windows 2000 deployments.

Windows 2000 clients can add entries for themselves into DNS by directly updating forward zones with their address (A) record. They cannot update reverse zones with their pointer (PTR) records

Client DNS Updates

It is not recommended that clients be allowed to update DNS directly. (See the [“Recommended Windows 2000 Design Practices”](#) section on page 15-14.)

For a Windows client to directly send address record updates to the DNS server, two conditions must apply:

- The Windows 2000 client must have the **Register this connection’s addresses in DNS** box checked on the **DNS** tab of its TCP/IP control panel settings.
- The DHCP policy must enable direct updating (Network Registrar policies do so by default).

The Windows 2000 client notifies the DHCP server of its intention to update the A record to the DNS server by sending the *client-fqdn* DHCP option (81) in a DHCPREQUEST packet. By indicating the fully qualified domain name (FQDN), the option states unambiguously the client’s location in the domain namespace. Along with the FQDN itself, the client or server can send one of these possible flags in the *client-fqdn* option:

- 0—Client should register its A record directly with the DNS server, and the DHCP server registers the PTR record (done through the policy’s *allow-client-a-record-update* attribute being enabled).
- 1—Client wants the DHCP server to register its A and PTR records with the DNS server.

- 3—DHCP server registers the A and PTR records with the DNS server regardless of the client's request (done through the policy's *allow-client-a-record-update* attribute being disabled, which is the default). Only the DHCP server can set this flag.

The DHCP server returns its own *client-fqdn* response to the client in a DHCPACK based on whether dynamic DNS update is enabled. However, if the 0 flag is set (the *allow-client-a-record-update* attribute is enabled for the policy), enabling or disabling dynamic DNS update is irrelevant, because the client can still send its updates to DNS servers. See [Table 15-2](#) for the actions taken based on how various properties are set.

Table 15-2 Windows 2000 Client DNS Update Options

DHCP Client Action	Dynamic DNS	DHCP Server Action
Checks Register this connection's addresses in DNS and sends <i>client-fqdn</i> ; DHCP server enables <i>allow-client-a-record-update</i>	Enabled or disabled	Responds with <i>client-fqdn</i> that it allows the client to update its A records (sets flag 0), but the DHCP server still updates the PTR records.
Checks Register... and sends <i>client-fqdn</i> ; DHCP disables <i>allow-client-a-record-update</i>	Enabled	Responds with <i>client-fqdn</i> that it does not allow the client to update the DNS server directly (sets flag 3), and updates the A and PTR records.
	Disabled	Does not respond with <i>client-fqdn</i> and does not update the DNS server.
Unchecks Register... and sends <i>client-fqdn</i>	Enabled	Responds with <i>client-fqdn</i> that it is updating the A and PTR records.
	Disabled	Does not respond with <i>client-fqdn</i> and does not update the DNS server.
Does not send <i>client-fqdn</i>	Enabled	Does not respond with <i>client-fqdn</i> , but updates the A and PTR records.
	Disabled	Does not respond with <i>client-fqdn</i> and does not update the DNS server.

A Windows 2000 RC3 DHCP server can set the *client-fqdn* option to ignore the client's request. To enable this in Network Registrar, create a policy for Windows 2000 clients and disable the *allow-client-a-record-update* attribute for this policy.

The following attributes are enabled by default in Network Registrar:

- Server *use-client-fqdn*—The server uses the *client-fqdn* value on incoming packets and does not examine the *host-name*. The DHCP server ignores all characters after the first dot in the domain name value, because it determines the domain from the defined scope for that client. Disable *use-client-fqdn* only if you do not want the server to determine host names from *client-fqdn*, possibly because the client is sending unexpected characters.
- Server *use-client-fqdn-first*—The server examines *client-fqdn* on incoming packets from the client before examining the *host-name* option (12). If *client-fqdn* contains a host name, the server uses it. If the server does not find the option, it uses the *host-name* value. If *use-client-fqdn-first* is disabled, the server prefers the *host-name* value over *client-fqdn*.
- Server *use-client-fqdn-if-asked*—The server returns the *client-fqdn* value in the outgoing packets if the client requests it. For example, the client might want to know the status of DNS activity, and hence request that the DHCP server should present the *client-fqdn* value.

- Policy *allow-client-a-record-update*—The client can update its A record directly with the DNS server, as long as the client sets the *client-fqdn* flag to 0 (requesting direct updating). Otherwise, the server updates the A record based on other configuration properties.

The host names returned to client requests vary depending on these settings, as described in (see [Table 15-3](#)).

Table 15-3 Host Names Returned Based on Client Request Parameters

Client Request	With Server/Policy Settings	Resulting Host Name
Includes <i>host-name</i> (option 12)	<i>use-host-name</i> =true <i>use-client-fqdn</i> =false (or <i>use-client-fqdn-first</i> =false) <i>trim-host-name</i> =true	<i>host-name</i> trimmed at first dot. Example: <i>host-name</i> host1.bob is returned host1
	(same except:) <i>trim-host-name</i> =false	<i>host-name</i> . Example: <i>host-name</i> host1.bob is returned host1.bob
Includes <i>client-fqdn</i> (option 81)	<i>use-client-fqdn</i> =true <i>use-host-name</i> =false (or <i>use-client-fqdn-first</i> =true)	<i>client-fqdn</i> trimmed at first dot. Example: <i>client-fqdn</i> host1.bob is returned host1.
Omits <i>host-name</i> (option 12) and <i>client-fqdn</i> (option 81)	Or: <i>use-host-name</i> =false <i>use-client-fqdn</i> =false	Set by client/policy hierarchy
	(same as previous except:) host name is undefined in the client/policy hierarchy <i>synthesize-name</i> =true	Synthesized following the synthesizing rules
	(same as previous except:) <i>synthesize-name</i> =false	Undefined

Dual Zone Updates for Windows 2000 Clients

Windows 2000 DHCP clients might be part of a DHCP deployment where they have A records in two DNS zones. In this case, the DHCP server returns the *client-fqdn* so that the client can request a dual zone update. To enable a dual zone update, enable the policy attribute *allow-dual-zone-dns-update*.

The DHCP client sends the 0 flag in *client-fqdn* and the DHCP server returns the 0 flag so that the client can update the DNS server with the A record in its main zone. However, the DHCP server also directly sends an A record update based on the client's secondary zone in the client's behalf. If both *allow-client-a-record-update* and the *allow-dual-zone-dns-update* are enabled, allowing the dual zone update takes precedence so that the server can update the secondary zone's A record.

Dynamic DNS Update Settings in Windows 2000 Clients

The Windows 2000 RC3 client can set advanced properties to enable sending the *client-fqdn* option.

- Step 1 On the Windows 2000 RC3 client, go to the Control Panel and open the TCP/IP Settings dialog box.
- Step 2 Click the **Advanced** tab.

- Step 3** Click the **DNS** tab.
- Step 4** To have the client send the *client-fqdn* option in its request, leave the **Register this connection's addresses in DNS** box checked. This indicates that the client wants to do the A record update.
-

Windows 2000 Client Settings in DHCP Servers

You can apply a relevant policy to a scope that includes the Windows 2000 clients, and enable DNS updates for the scope.

-
- Step 1** Create a policy for the scope that includes the Windows 2000 clients. For example:
- Create a policywin2k.
 - Create a win2k scope with the subnet 192.168.1.0/24 and policywin2k as the policy. Add an address range of 192.168.1.10 through 192.168.1.100.
- Step 2** Enable the scope attribute *dynamic-dns*.
- Step 3** Set the zone name, server address (for A records), reverse zone name, and reverse server address (for PTR records), as described in the [“Configuring Dynamic DNS for Scopes” section on page 15-2](#).
- Step 4** If you want the client to update its A records at the DNS server, enable the policy attribute *allow-client-a-record-update* (this is the default). There are a few caveats to this:
- If *allow-client-a-record-update* is enabled and the client sends the *client-fqdn* with the update bit enabled, the *host-name* and *client-fqdn* returned to the client match the client's *client-fqdn*. (However, if the *override-client-fqdn* is also enabled on the server, the host name and FQDN returned to the client are generated by the configured host name and policy domain name.)
 - If, instead, the client does not send the *client-fqdn* with the update bit enabled, the server does the A record update, and the *host-name* and *client-fqdn* (if requested) returned to the client match the name used for the dynamic DNS update.
 - If *allow-client-a-record-update* is disabled, the server does the A record updates, and the *host-name* and *client-fqdn* (with the update bit disabled) values returned to the client match the name used for the dynamic DNS update.
 - If *allow-dual-zone-dns-update* is enabled, the DHCP server always does the A record updates. (See the [“Dual Zone Updates for Windows 2000 Clients” section on page 15-11](#).)
 - If *use-dns-prereqs* is enabled and *update-dns-first* is disabled, the host name and *client-fqdn* returned to the client are not guaranteed to match the DNS update, because of delayed name disambiguation, but the lease data will be updated with the new names.
- Step 5** Reload the DHCP server.
-

SRV Records and Dynamic DNS Updates

Windows 2000 relies heavily on the DNS protocol for advertising services to the network. [Table 15-4](#) describes how Windows 2000 handles service location (SRV) DNS resource records and dynamic DNS updates.

Table 15-4 Windows 2000 SRV Records and Dynamic DNS Updates

Feature	Description
SRV records	<p>Windows 2000 domain controllers use the SRV resource record to advertise services to the network. This resource record is defined in the experimental RFC 2782, "A DNS RR for specifying the location of services (DNS SRV)." The RFC defines the format of the SRV record (DNS type code 33) as:</p> <pre><i>_service._protocol.name ttl class SRV priority weight port target</i></pre> <p>There should always be an A record associated with the SRV record's target, so that the client can resolve the service back to a host. In the Windows 2000 implementation of SRV records, the records might look like this:</p> <pre>myserver.example.com A 10.100.200.11 _ldap._tcp.example.com SRV 0 0 389 myserver.example.com _kdc._tcp.example.com SRV 0 0 88 myserver.example.com _ldap._tcp.dc_msdc.example.com SRV 0 0 88 myserver.example.com</pre> <p>An underscore always precedes the service and protocol names. In the example, <code>_kdc</code> is the Kerberos Data Center. The priority and weight help you choose between target servers providing the same service (the weight differentiating those with equal priorities). With zero priority and weight, the listed order determines the priority. Windows 2000 domain controllers automatically place these SRV records in DNS.</p>
How SRV records are used	<p>When a Windows 2000 client boots up, it tries to initiate the network login process to authenticate against its Windows 2000 domain controller. The client must first discover where the domain controller is, and they do so using the dynamically generated SRV records.</p> <p>Before launching the net-login process, the client queries DNS with a service name, such as <code>_ldap._tcp.dc_msdc.example.com</code>. The DNS server SRV record target, for example, is <code>my-domain-controller.example.com</code>. The Windows 2000 client then queries DNS with the host name <code>my-domain-controller.example.com</code>. DNS returns the host address and the client uses this address to find the domain controller. The net-login process fails without these SRV records.</p>
Dynamic DNS updates	<p>When a Windows 2000 server is configured as a domain controller, you statically configure the name of the domain it manages through the Active Directory management console. This Windows 2000 domain should have a corresponding DNS zone associated with it. The domain controller should also have a series of DNS resolvers configured in its TCP/IP properties control panel.</p> <p>When the Windows 2000 domain controller boots up, it performs these steps to register itself in DNS and advertise its services to the network:</p> <ol style="list-style-type: none"> 1. Queries DNS asking for the start of authority (SOA) record for the DNS domain that mostly closely encapsulates its Windows 2000 domain. 2. Identifies the primary DNS server for the DNS zone (from the SOA record) that mostly closely encapsulates its Windows 2000 domain name. 3. Creates a series of SRV records in this zone using the RFC 2136 dynamic DNS update protocol.

Table 15-4 Windows 2000 SRV Records and Dynamic DNS Updates (continued)

Feature	Description
Server boot process log file examples	<p>Under normal operating conditions, the Network Registrar primary DNS server writes these log entries when a Windows 2000 domain controller boots up and creates its SRV records:</p> <pre>data time name/dns/1 Activity Protocol 0 Added type 33 record to name "_ldap._tcp.w2k.example.com", zone "w2k.example.com"</pre> <pre>data time name/dns/1 Activity Protocol 0 Update of zone "w2k.example.com" from address [10.100.200.2] succeeded.</pre> <p>This log shows only one dynamic DNS update for a single SRV record. A Windows 2000 domain controller typically registers 17 of these SRV records when it boots up.</p>

You can configure the Network Registrar DNS server so that Windows 2000 domain controllers can dynamically register their services in DNS and, thereby, advertise themselves to the network. Because this process occurs through RFC-compliant dynamic DNS updates, you do not need to do anything out of the ordinary in Network Registrar.

To configure Network Registrar to accept these dynamic SRV record updates:

-
- Step 1** Determine the IP addresses of the devices in the network that need to advertise services through DNS.
 - Step 2** If they do not exist, create the appropriate forward and reverse zones for the Windows 2000 domains.
 - Step 3** Enable dynamic DNS updates for the forward and reverse zones.
 - Step 4** Set the value of the zone's *update-acl* attribute to define the IP addresses of the hosts to which you want to restrict accepting dynamic updates. These are usually the DHCP servers and any Windows 2000 domain controllers. (The Windows 2000 domain controllers should have static IP addresses.)

If it is impractical or impossible to enter the list of all the IP addresses from which a DNS server must accept updates, you can configure Network Registrar to accept updates from a range of addresses, although Cisco does not recommend this configuration.
 - Step 5** Reload the DNS and DHCP servers.
-

Recommended Windows 2000 Design Practices

Most Windows 2000 deployments are migrations from Windows NT 4.0 environments. Cisco suggests that you preserve the existing Windows NT 4.0 domain structure wherever possible. These rules and recommendations make Windows 2000 deployments more manageable.

Windows 2000 Rules and Suggestions

Consider these rules and suggestions for Windows 2000 environments:

- Windows 2000 domains must not violate naming conventions for DNS host names.
- Statically configure the IP addresses of all Windows 2000 domain controllers.
- Change existing Windows NT 4.0 domain names to support DNS naming conventions.

- Where necessary, create new DNS zones for Windows 2000 domains, such as the w2k.example.com subzone and the local.w2k.example.com domain.
- Do not allow clients to dynamically update their zones. Instead, configure the DHCP server so that it is responsible for all dynamic DNS updates.

Naming Rules for Windows 2000 Domains

Because the Windows 2000 domain space overlaps with the DNS name space, certain naming restrictions apply to Windows 2000 domains. Section 2.3.1 of RFC 1035 (“Domain Names Implementation and Specification”) defines naming conventions for host names and domain names. Because of the dependence on DNS, these naming conventions apply to Windows 2000 domains.

- Host names and domain names in DNS are not case sensitive.
- Host names must:
 - Start with a letter.
 - End with a letter or a digit.
 - Contain internal characters that are letters, digits, or hyphens.

Therefore, you cannot use many characters commonly used in Windows NT 4.0 domain names for Windows 2000 domain names. Characters that you should not use in Windows 2000 domain names include the underscore (_), “at” symbol (@), and ampersand (&), all commonly used in Windows NT 4.0 domain names. Also, mixed-case domain names are no longer useful. For example, Windows 2000 recognizes ExampleDomain as equivalent to exampledomain.

Issues Related to Windows 2000 Environments

[Table 15-5](#) describes the issues concerning interoperability between Windows 2000 and Network Registrar, intended to inform you of possible problems before you encounter them in the field. For some frequently asked questions about Windows 2000 interoperability, see the “[Frequently Asked Questions About Windows 2000 Integration](#)” section on page 15-19.

Table 15-5 Issues Concerning Windows 2000 and Network Registrar Interoperability

Issue	Description
Invisible dynamically created resource records	<p>Network Registrar, when properly configured, accepts dynamic DNS updates from both DHCP and Windows 2000 servers. You can use the CLI to access the dynamic portion of the DNS zone for viewing and deleting records. Enter this command to view all dynamic DNS resource records in a given zone:</p> <pre>nrcmd> zone myzone listRR dynamic myfile</pre> <p>This redirects the output to the myfile file (see Example 15-1 on page 15-18).</p> <p>You can delete dynamically generated records by entering this command:</p> <pre>nrcmd> zone myzone removeDynRR myname [type]</pre> <p>You can also use nslookup to verify their existence, and you can use version 5.x (shipped with Windows 2000) to view dynamic SRV records. In this version, use the set type=SRV command to enable viewing SRV records.</p>

Table 15-5 Issues Concerning Windows 2000 and Network Registrar Interoperability (continued)

Issue	Description
Domain controller registration	<p>A Windows 2000 domain controller has to register itself in DNS using dynamic DNS updates. The DNS RFCs dictate that only a primary DNS server for a particular zone can accept edits to the zone data. Hence, the Windows 2000 domain controller has to discover which DNS server is the primary for the zone that includes its Windows 2000 domain name.</p> <p>The domain controller discovers this by querying the first DNS server in its resolver list (configured in the TCP/IP properties control panel). The initial query is for the SOA record of the zone that includes the domain controller's Windows 2000 domain. The SOA record includes the name of the primary server for the zone. If no zone exists for the domain name, the domain controller keeps removing the left-most label of the domain name and sends queries until it finds an SOA record with a primary server included in that domain. Once the domain controller has the name of the primary DNS server for its domain, it sends it DNS updates to create the necessary SRV records.</p> <p>Ensure that the name of the zone's primary DNS server is in its SOA record.</p>
Failure of A record dynamic DNS updates	<p>When a Windows 2000 domain controller tries to advertise itself to the network, it sends several dynamic DNS update requests to the DNS server of record for its domain. Most of these update requests are for SRV records. However, the domain controller also requests an update for a single A record of the same name as the Windows 2000 domain.</p> <p>If the Network Registrar DNS server is also authoritative for a zone identical to this Windows 2000 domain, it rejects registering its A record, because the dynamic DNS A record update conflicts with the static SOA and NS records. This is to prevent possible security infractions, such as a dynamic host registering itself and spoofing Web traffic to a site.</p> <p>For example, the domain controller might control the w2k.example.com Windows 2000 zone. If a zone with the same name exists on the Network Registrar DNS server, these resource records could be part of that zone:</p> <pre data-bbox="613 1283 1252 1520"> w2k.example.com. 43200 SOA nameserver.example.com. hostmaster.example.com. (98011312 ;serial 3600 ;refresh 3600 ;retry 3600000 ;expire 43200) ;minim w2k.example.com.86400 NS nameserver.example.com </pre> <p>The domain controller would try to add an additional record, such as:</p> <pre data-bbox="613 1604 1076 1629"> w2k.example.com. 86400 A 192.168.2.1 </pre> <p>Network Registrar does not allow a dynamic DNS update to conflict with any statically configured name in the zone, even if the record type associated with that name is different. In the above example, an attempt to add an A record associated with the name w2k.example.com collides with the SOA and NS records.</p>

Table 15-5 Issues Concerning Windows 2000 and Network Registrar Interoperability (continued)

Issue	Description
	<p>When the domain controller boots up, a DNS log file entry such as this appears:</p> <pre data-bbox="656 394 1471 470">08/10/2000 16:35:33 name/dns/1 Info Protocol 0 Error - REFUSED - Update of static name "w2k.example.com", from address [10.100.200.2]</pre> <p>This is how Network Registrar responds to dynamic updates of static DNS data. Additionally, you can ignore this dynamic DNS update failure. Windows clients do not use this A record. Allocation of domain controllers happens through SRV records. Microsoft added the A record to accommodate legacy NT clients that do not support SRV records.</p> <p>Note that failing to register the controller's A record slows down the domain controller's bootup process, affecting the overall login of worker clients. As mentioned earlier, the workaround is to define the Windows 2000 domain as a subdomain of the authoritative zone, or enable the DNS server's <i>simulate-zone-top-dynupdate</i> attribute. If this is not possible, contact the Cisco Technical Assistance Center for help.</p>
Windows 2000 RC1 DHCP clients	<p>Microsoft released Windows 2000 build 2072 (known as RC1) with a broken DHCP client. This client sends a malformed packet that Network Registrar cannot parse. Network Registrar drops the packet and cannot serve the client, logging this error:</p> <pre data-bbox="656 1016 1503 1092">08/10/2000 14:56:23 name/dhcp/1 Activity Protocol 0 10.0.0.15 Lease offered to Host:'My-Computer' CID: 01:00:a0:24:1a:b0:d8 packet'R15' until True, 10 Aug 2000 14:58:23 -0400. 301 ms.</pre> <pre data-bbox="656 1121 1484 1197">08/10/2000 14:56:23 name/dhcp/1 Warning Protocol 0 Unable to find necessary Client information in packet from MAC address:'1,6,00:d0:ba:d3:bd:3b'. Packet dropped!</pre> <p>Network Registrar includes error checking specifically designed to deal with errors such as this improperly built FQDN option. However, if you encounter this problem, install the Microsoft patch to the RC1 client on the DHCP client. You must obtain this patch from Microsoft.</p>
Windows 2000 plug-and-play network interface card (NIC) configuration	<p>If configured to use DHCP, a Windows 2000 system tries to obtain a DHCP lease on startup. If no DHCP server is available, Windows 2000 may automatically configure the computer's interface with a plug-and-play IP address. This address is not an address the network administrator or DHCP server configured or selected.</p> <p>These plug-and-play addresses are in the range 169.254.0.0/16. If you see devices in this address range on a network, it means that Windows 2000 autoconfigured the interfaces because it could not obtain a lease from a DHCP server.</p> <p>This can cause significant network and troubleshooting problems. The Windows 2000 system no longer informs the user that the DHCP client could not obtain a lease. Everything appears to function normally, but the client cannot route packets off its local subnet. Additionally, you may see the DHCP client trying to operate on the network with an address from the 169.254.0.0/16 network. This may lead you to think that the Network Registrar DHCP server is broken and handing out the wrong addresses.</p>

Table 15-5 Issues Concerning Windows 2000 and Network Registrar Interoperability (continued)

Issue	Description
	<p>If this problem occurs, perform these steps:</p> <ol style="list-style-type: none"> 1. Ensure that the DHCP client has an active network port and a properly configured NIC. 2. Ensure that the network between the client and the DHCP server(s) are properly configured. Ensure that all router interfaces are configured with the correct IPHelper address. 3. Reboot the DHCP client. 4. If necessary, look at the DHCP log file. If the DHCP client can successfully route packets to the server, this logs a DHCPDISCOVER, even if Network Registrar does not answer the packet. <p>If the network is correctly configured, and if the DHCP client is not broken, Network Registrar should receive the packet and log it. If there is no log entry for a packet receipt, there is a problem somewhere else in the network.</p>
Scavenging Windows 2000 client address records	<p>Windows 2000 clients do not clean after themselves, potentially causing their dynamic record registration to remain indefinitely. This leaves stale address records on the DNS server. To ensure that these stale records are periodically removed, you must enable scavenging for the zone (see the “Scavenging Dynamic Records” section on page 15-7).</p>

Example 15-1 Output Showing Invisible Dynamically Created Resource Records

```

Dynamic Resource Records
_ldap._tcp.test-lab._sites 600 IN SRV 0 100 389 CNR-MKT-1.w2k.example.com.
_ldap._tcp.test-lab._sites.gc._msdcs 600 IN SRV 0 100 3268 CNR-MKT-1.w2k.example.com.
_kerberos._tcp.test-lab._sites.dc._msdcs 600 IN SRV 0 100 88 CNR-MKT-1.w2k.example.com.
_ldap._tcp.test-lab._sites.dc._msdcs 600 IN SRV 0 100 389 CNR-MKT-1.w2k.example.com.
_ldap._tcp 600 IN SRV 0 100 389 CNR-MKT-1.w2k.example.com.
_kerberos._tcp.test-lab._sites 600 IN SRV 0 100 88 CNR-MKT-1.w2k.example.com.
_ldap._tcp.pdc._msdcs 600 IN SRV 0 100 389 CNR-MKT-1.w2k.example.com.
_ldap._tcp.gc._msdcs 600 IN SRV 0 100 3268 CNR-MKT-1.w2k.example.com.
_ldap._tcp.1ca176bc-86bf-46f1-8a0f-235ab891bcd2.domains._msdcs 600 IN SRV 0 100 389
    CNR-MKT-1.w2k.example.com.
e5b0e667-27c8-44f7-bd76-6b8385c74bd7._msdcs 600 IN CNAME CNR-MKT-1.w2k.example.com.
_kerberos._tcp.dc._msdcs 600 IN SRV 0 100 88 CNR-MKT-1.w2k.example.com.
_ldap._tcp.dc._msdcs 600 IN SRV 0 100 389 CNR-MKT-1.w2k.example.com.
_kerberos._tcp 600 IN SRV 0 100 88 CNR-MKT-1.w2k.example.com.
_gc._tcp 600 IN SRV 0 100 3268 CNR-MKT-1.w2k.example.com.
_kerberos._udp 600 IN SRV 0 100 88 CNR-MKT-1.w2k.example.com.
_kpasswd._tcp 600 IN SRV 0 100 464 CNR-MKT-1.w2k.example.com.
_kpasswd._udp 600 IN SRV 0 100 464 CNR-MKT-1.w2k.example.com.
gc._msdcs 600 IN A 10.100.200.2
_gc._tcp.test-lab._sites 600 IN SRV 0 100 3268 CNR-MKT-1.w2k.example.com.

```

Frequently Asked Questions About Windows 2000 Integration

These questions are frequently asked about integrating Network Registrar DNS services with Windows 2000:

Q. **What happens if both Windows 2000 clients and the DHCP server are allowed to update the same zone? Can this create the potential for stale DNS records being left in a zone? If so, what can be done about it?**

A. The recommendation is not to allow Windows 2000 clients to update their zones. Instead, the DHCP server should manage all the client dynamic RR records. When configured to perform dynamic DNS updates, the DHCP server accurately manages all the resource records associated with the clients that it served leases to. In contrast, Windows 2000 client machines blindly send a daily dynamic DNS update to the server, and when removed from the network, leave a stale DNS entry behind.

Any zone being updated by dynamic DNS update clients should have DNS scavenging enabled to shorten the longevity of stale resource records left by transient Windows 2000 clients. If the DHCP server and Windows 2000 clients are both updating the same zone, three things are required in Network Registrar:

a. Enable scavenging for the zone.

b. Configure the DHCP server to refresh its dynamic DNS update entries as each client renews its lease. By default, Network Registrar does not re-update the DNS record between its creation and its final deletion. A dynamic DNS update record that Network Registrar creates lives from the start of the lease until the lease expires. You can change this behavior using a DHCP server attribute, *force-dns-updates*. For example:

```
nrcmd> dhcp enable force-dns-updates
100 Ok
force-dns-updates=true
```

c. If scavenging is enabled on a particular zone, then the lease time associated with clients that the DHCP server updates that zone on behalf of must be less than the sum of the *no-refresh-interval* and *refresh-interval* scavenging settings. Both of these settings default to seven days. You can set the lease time to 14 days or less if you do not change these default values.

Q. **What needs to be done to integrate a Windows 2000 domain with a pre-existing DNS domain naming structure if it was decided not to have overlapping DNS and Windows 2000 domains? For example, if there is a pre-existing DNS domain called example.com and a Windows 2000 domain is created that is called w2k.example.com, what needs to be done to integrate the Windows 2000 domain with the DNS domain?**

A. In the example, a tree in the Windows 2000 domain forest would have a root of w2k.example.com. There would be a DNS domain named example.com. This DNS domain would be represented by a zone named example.com. There may be additional DNS subdomains represented in this zone, but no subdomains are ever delegated out of this zone into their own zones. All the subdomains will always reside in the example.com. zone.

Q. *In this case, how are dynamic DNS updates from the domain controllers dealt with?*

A. To deal with the SRV record updates from the Windows 2000 domain controllers, limit dynamic DNS updates to the example.com. zone to the domain controllers by IP address only. (Later, you will also add the IP address of the DHCP server to the list.) Enable scavenging on the zone. The controllers will update SRV and A records for the w2k.example.com subdomain in the example.com

zone. There is no special configuration required to deal with the A record update from each domain controller, because an A record for w2k.example.com does not conflict with the SOA, NS, or any other static record in the example.com zone.

The example.com zone then might include these records:

```
example.com. 43200 SOA ns.example.com. hostmaster.example.com. (
    98011312 ;serial
    3600 ;refresh
    3600 ;retry
    3600000 ;expire
    43200 ) ;minimum
example.com.86400 NS ns.example.com
ns.example.com. 86400 A 10.0.0.10
_ldap._tcp.w2k.example.com. IN SRV 0 0 389 dc1.w2k.example.com
w2k.example.com 86400 A 10.0.0.25
...
```

- Q. *In this case, how are zone updates from individual Windows 2000 client machines dealt with?*
- A. In this scenario, the clients could potentially try to update the example.com. zone with updates to the w2k.example.com domain. The way to avoid this is to close down the zone to updates except from trusted sources. Before Network Registrar 6.0, the DNS server should be configured to accept updates from the DHCP server by IP address only. With Network Registrar 6.0, you can use transaction signatures (TSIG) between the DHCP server and the primary DNS server for the example.com zone.
- Configure the DHCP server to do dynamic DNS updates to the example.com zone and the appropriate reverse zone for each client, and use option 81 to prevent the clients from doing dynamic DNS updates themselves.
- Q. *Has security been addressed in this case?*
- A. By configuring the forward and reverse zone to accept only updates from trusted IP addresses, you close the zone to updates from any other device on the network. Security by IP is not the most ideal solution, as it would not prevent a malicious attack from a spoofed IP address source. You can secure updates from the DHCP server by configuring TSIG between the DHCP server and the DNS server.
- Q. *Is scavenging required in this case?*
- A. No. Updates are only accepted from the domain controllers and the DHCP server. The DHCP server accurately maintains the life cycle of the records that they add and do not require scavenging. You can manage the domain controller dynamic entries manually by using the Network Registrar single-record dynamic resource record removal feature.
- Q. **What needs to be done to integrate a Windows 2000 domain that shares its namespace with a DNS domain? For example, if there is a pre-existing DNS zone called example.com and a Windows 2000 Active Directory domain called example.com needs to be deployed, how can it be done?**
- A. In this example, a tree in the Windows 2000 domain forest would have a root of example.com. There is a pre-existing domain that is also named example.com that is represented by a zone named example.com.
- Q. *In this case, how are dynamic DNS updates from individual Windows 2000 client machines dealt with?*

- A. To deal with the SRV record updates, create subzones for:

```
_tcp.example.com.  
_sites.example.com.  
_msdcs.example.com.  
_msdcs.example.com.  
_udp.example.com.
```

Limit dynamic DNS updates to those zones to the domain controllers by IP address only. Enable scavenging on these zones.

To deal with the A record update from each domain controller, enable a DNS server attribute, *simulate-zone-top-dynupdate*:

```
nrcmd> dns enable simulate-zone-top-dynupdate
```

It is not required, but if desired, manually add an A record for the domain controllers to the example.com zone.

- Q. *In this case, how are zone updates from individual Windows 2000 client machines dealt with?*
- A. In this scenario, the clients could potentially try to update the example.com zone. The way to avoid this is to close down the zone to updates except from trusted sources. Before Network Registrar 6.0, the DNS server should be configured to accept updates from the DHCP server by IP address only. As of Network Registrar 6.0, you can use transaction signatures (TSIG) between the DHCP server and the primary DNS server for the example.com zone.

Configure the DHCP server to do dynamic DNS updates to the example.com zone and the appropriate reverse zone for each client, and use option 81 to prevent the clients from doing dynamic DNS updates themselves.

- Q. *Has security been addressed in this case?*
- A. By configuring the forward and reverse zone to accept only updates from trusted IP addresses, you close the zone to updates from any other device on the network. Security by IP is not the most ideal solution, as it would not prevent a malicious attack from a spoofed IP address source. Updates from the DHCP server are more secure when TSIG is configured between the DHCP server and the DNS server.
- Q. *Has scavenging been addressed in this case?*
- A. Yes. The subzones _tcp.example.com, _sites.example.com, _msdcs.example.com, _msdcs.example.com, and _udp.example.com zones accept updates only from the domain controllers and scavenging was turned on for these zones. The example.com zone accepts dynamic DNS updates only from the DHCP server.

