Dynamic DNS Support for Cisco IOS Software

The Dynamic DNS Support for Cisco IOS Software feature enables Cisco IOS software devices to perform Dynamic Domain Name System (DDNS) updates to ensure that an IP host DNS name is correctly associated with its IP address.

It provides two mechanisms to generate or perform DDNS: the IETF standard as defined by RFC 2136 and a generic HTTP using various DNS services. With this feature, you can define a list of hostnames and IP addresses that will receive updates, specify an update method, and specify a configuration for Dynamic Host Configuration Protocol (DHCP) triggered updates.

- Finding Feature Information, page 1
- Restrictions for Dynamic DNS Support for Cisco IOS Software, page 1
- Information About Dynamic DNS Support for Cisco IOS Software, page 2
- How to Configure Dynamic DNS Support for Cisco IOS Software, page 3
- Configuration Examples for Dynamic DNS Support for Cisco IOS Software, page 23
- Additional References, page 25
- Feature Information for Dynamic DNS Support for Cisco IOS Software, page 27

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for Dynamic DNS Support for Cisco IOS Software

The performance of the DHCP client can be impacted when the Dynamic DNS Support for Cisco IOS Software feature is enabled, because of sending DDNS update packets and waiting for responses from the server (before sending the ACK to the client REQUEST) and the client (immediately after receiving the ACK and assigning

the address to the interface). The default for the client is two attempts with a 5-second wait time between attempts.

The DHCP server continues to process DHCP client DISCOVER and REQUEST packets while waiting for the DDNS updates to complete. Even if the update is done before sending the ACK to the client, it does not delay processing of other DHCP requests. The DHCP server could be impacted minimally because of the time and memory needed in order to set up the DDNS update and get things started.

Reloading the system may take a little longer in some cases, such as, if there are outstanding DDNS updates that need to complete.

Information About Dynamic DNS Support for Cisco IOS Software

Domain Name System and Dynamic Updates

The DNS was designed to support queries of a statically configured database. The data was expected to change, but minimally. All updates were made as external edits to a zone master file. The domain name identifies a node within the domain name space tree structure. Each node has a set (possibly empty) of Resource Records (RRs). All RRs having the same NAME, CLASS, and TYPE are called a Resource Record Set (RRset).

There are address (A) or forward RRs and pointer (PTR) or reverse RRs. The DDNS update can specify additions or deletions of hostnames and IP addresses. The two mechanisms to update this information are by using HTTP-based protocols such as DynDNS.org or by using the IETF standard.

DDNS Updates for HTTP-Based Protocols

The Dynamic DNS Support for Cisco IOS Software feature provides the capability of a proprietary HTTP-based protocol to generate or perform DDNS updates. The most notable HTTP-based protocol is DynDNS.org, but there are many others.

Since most of these protocols consist of a simple HTTP command that specifies parameters such as hostname and IP address in the URL portion of the command, this feature takes the same generic approach. You can specify the hostname and IP address in a URL. Configuration of a maximum interval between updates is also allowed.

DHCP Support for DDNS Updates

Before the Dynamic DNS Support for Cisco IOS Software feature, a DHCP server assigned IP addresses to DHCP clients and any DNS information was static. In a network that uses a DHCP server, there are many cases in which DNS hostnames should be associated with the IP addresses that are being assigned. There is an existing method for dynamically updating DNS for DHCP by using information in the fully qualified domain name (FQDN) DHCP option (if it is supplied by the client).

The Dynamic DNS Support for Cisco IOS Software feature enables the DHCP server to support a new FQDN DHCP option. In addition, when the address on an interface is configured, the client can pass the new FQDN option to the server so that name-to-address and address-to-name translations can be updated for the DHCP client as well.

Feature Design of Dynamic DNS Support for Cisco IOS Software

The Dynamic DNS Support for Cisco IOS Software feature enables the tracking of the FQDN DHCP option. If dynamic updates are enabled for the DHCP server, the server updates the PTR RR. The PTR RRs are used for reverse mapping (translation of addresses to names). PTRs use official names not aliases. The name in a PTR record is the local IP address portion of the reverse name.

If the client requests the server to update A RRs as well, the server will attempt to do it. The A RR provides the name-to-address mapping for a DNS zone. The server may be configured to override the client suggestion and always update PTR and A RRs.

The DHCP client can specify whether or not it wants to allow dynamic updates (include the FQDN option), instruct the server to allow the client to update both A and PTR RRs (normally only the A RR is updated by the client), and optionally instruct the server not to update any DNS information (either because the client will be updating both or simply because the client does not want the server to do any updates at all).

There are three basic components of the Dynamic DNS Support for Cisco IOS Software feature that are as follows:

- Definition of the hostname list and IP addresses that will receive updates using a new command that specifies a group of hostnames. Each configured list can consist of any number of IPv4 addresses or hostnames. If a hostname is configured, the name is translated to an IPv4 address at the time at which it is used.
- Specification of an update method. The options are HTTP, DDNS, or an internal Cisco IOS name cache. If the HTTP option is specified, the configuration will include a URL. The username and password must be explicitly written into the URL string and the entire “GET” operation must be specified on one line. The specification will be stored in a linked list. If the update method is DDNS, the configuration will include the update of the IP address.

Events that trigger updates can be as follows:

- IP address that is assigned by a DHCP server for an IP device
- IP address assigned to a router using a DHCP client
- Forwarding of the fully qualified domain name (FQDN) of a user or router hostname from the DHCP client to the server
- Point-to-Point Protocol (PPP)/IP Control Protocol (IPCP) obtaining an IP address for a router interface
- Forced update using a timer to verify a router IP address

Associated with each update method is a value specifying the maximum number of seconds between updates. If left unspecified, then the update is performed only when the address is changed. If specified, the update is performed automatically if the specified number of seconds have passed since the last update.

How to Configure Dynamic DNS Support for Cisco IOS Software

Note The internal Cisco IOS name cache does not require any configuration.

Configuring a Host List

Perform this task to configure a host list if you are going to use a host list in your configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip host-list** *host-list-name*
4. **host** [**vrf** *vrf-name*] {*host-ip-address* | *hostname*}
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip host-list** *host-list-name* **Example:** `Router(config)# ip host-list abc`	Specifies a list of hosts and enters host-list configuration mode. The *host-list-name argument*assigns a name to the list of hosts.
Step 4	**host** [**vrf** *vrf-name*] {*host-ip-address* | *hostname*} **Example:** `Router(host-list)# host 10.1.1.1 10.2.2.2 10.3.3.3 a.com b.com 10.4.4.4 10.5.5.5 d.com host 10.6.6.6 f.com host vrf abc a.com b.com c.com host vrf def 10.1.1.1 10.2.2.2 10.3.3.3`	Configures one or more hosts. The arguments and keyword are as follows: • **vrf** *vrf-name* --Associates a hostname with a virtual private network (VPN) routing and forwarding instance (VRF) name. **Note** All hostnames or IP addresses specified after the **vrf** keyword are associated with that VRF. • *host-ip-address* --Specifies an IP address for a host in the host list. You can specify more than one host using this argument by listing the hostname and IP addresses on the same line. • *hostname* --Specifies a hostname.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Router(host-list)# exit`	Exits to global configuration mode.

Examples

The following example shows how to configure several hosts with VRF:

```
ip host-list abc
 host 10.1.1.1 10.2.2.2 10.3.3.3 a.com b.com 10.4.4.4 10.5.5.5 d.com
 host 10.6.6.6 f.com
 host vrf abc a.com b.com c.com
 host vrf def 10.1.1.1 10.2.2.2 10.3.3.3
```

Verifying the Host-List Configuration

To verify the host-list configuration, perform the following steps.

SUMMARY STEPS

1. **show ip host-list**
2. **show running-config | inc host-list**
3. **show running-config | inc host**
4. **debug ip ddns update**

DETAILED STEPS

Step 1 **show ip host-list**

Use this command to verify that the IP addresses and hostnames have been assigned to a host list, for example:

Example:

```
Router# show ip host-list abc
Host list: abc
 ddns.abc
 10.2.3.4
 ddns2.abc
 10.3.4.5
 ddns3.com
 10.3.3.3
 d.org
 e.org
 1.org.2.org
 3.com
 10.2.2.2 (VRF: test)
 10.5.5.5 (VRF: test)
```

```
 a.net (VRF: test)
 b.net (VRF: test)
```

Step 2 **show running-config | inc host-list**

Use this command to verify the configuration of a host list, for example:

Example:

```
Router# show running-config | inc host-list
ip host-list a
ip host-list b
ip host-list c
ip host-list abc
```

Step 3 **show running-config | inc host**

Use this command to verify the configuration of a hostname, for example:

Example:

```
Router# show running-config | inc host
hostname who
ip host who 10.0.0.2
ip host-list a
 host 10.1.1.1 a.com b.com 10.2.2.3 10.2.2.2 c.com. 10.3.3.3 10.4.4.4
 host d.com
 host vrf abc 10.10.10.4 10.10.10.8
 host vrf def 10.2.3.4 10.6.7.8
ip host-list b
 host a.com b.com c.com 10.1.1.1 10.2.2.2 10.3.3.3
 host vrf ppp 10.2.1.0
ip host-list c
 host 10.1.1.1 10.2.2.2 10.3.3.3 a.com b.com 10.4.4.4 10.5.5.5 d.com
 host 10.6.6.6 f.com
 host vrf zero a.com b.com c.om
 host vrf one 10.1.1.1 10.2.2.2 10.3.3.3
ip host-list unit-test
 host ddns.unit.test 10.2.3.4 ddns2.unit.test 10.3.4.5 ddns3.com 10.3.3.3 d.org e.org
 host 1.org.2.org 3.com
 host vrf ZERO 10.2.2.2 10.5.5.5 a.net b.net
 ip ddns update hostname use-this.host.name
 ip ddns update this-method host 10.2.3.4
 ip ddns update this-method host this-host
 ip ddns update this-method host-group this-list
 ip ddns update this-method host 10.3.4.5
 ip ddns update test host 10.19.192.32
 ip ddns update test host 10.19.192.32
 ip ddns update a host-group a
 ip ddns update a host-group ab
 ip ddns update aa host-group ab
 ip ddns update method host 10.33.44.55
```

Step 4 **debug ip ddns update**

Use the **debug ip ddns update** command for the following configuration to verify the configuration of the hosts. Two servers are configured in the host list. A DHCP client is configured for IETF DDNS updating of both A and DNS RRs and requesting the DHCP server to update neither. The DHCP client is configured to include an FQDN DHCP option that instructs the DHCP server not to update either A or PTR Resource Records. This is configured using the interface version of the command. The DHCP server is configured to allow the DHCP client to update whatever RRs it chooses.

Example:

```
!Configure the DHCP Client
ip host-list servers
```

```
 host 10.19.192.32 10.0.0.1
ip ddns update method testing
 ddns
interface Ethernet1
 ip dhcp client update dns server none
 ip ddns update testing host-group servers
 ip address dhcp
end
!Configure the DHCP Server
ip dhcp pool test
 network 10.0.0.0 255.0.0.0
 update dns
!Enable Debugging
debug ip ddns update
!The update to the server 10.0.0.1 fails in this example
00:18:58:%DHCP-6-ADDRESS_ASSIGN: Interface Ethernet1 assigned DHCP address 10.0.0.8, mask 255.0.0.0,
 hostname canada_reserved
00:18:58: DYNDNSUPD: Adding DNS mapping for canada_reserved.hacks <=> 10.0.0.8 server 10.19.192.32
00:18:58: DYNDNSUPD: Sleeping for 3 seconds waiting for interface Ethernet1 configuration to settle
00:19:01: DDNS: Enqueuing new DDNS update 'canada_reserved.hacks' <=> 10.0.0.8 server 10.19.192.32
00:19:01: DYNDNSUPD: Adding DNS mapping for canada_reserved.hacks <=> 10.0.0.8 server 10.0.0.1
00:19:01: DDNS: Enqueuing new DDNS update 'canada_reserved.hacks' <=> 10.0.0.8 server 10.0.0.1
00:19:01: DYNDNSUPD: Adding DNS mapping for canada_reserved.hacks <=> 10.0.0.8 server 10.0.0.1
00:19:01: DDNS: Enqueuing new DDNS update 'canada_reserved.hacks' <=> 10.0.0.8 server 10.0.0.1
00:19:01: DDNS: Zone name for '10.0.0.11.in-addr.arpa.' is '10.in-addr.arpa'
00:19:01: DDNS: Using server 10.19.192.32
00:19:01: DDNS: Dynamic Update 1: (sending to server 10.19.192.32)
00:19:01: DDNS: Zone = 10.in-addr.arpa
00:19:01: DDNS: Prerequisite: 10.0.0.11.in-addr.arpa. not in use
00:19:01: DDNS: Update: add 10.0.0.11.in-addr.arpa. IN PTR canada_reserved.hacks
00:19:01: DDNS: Zone name for '10.0.0.11.in-addr.arpa.' is '10.in-addr.arpa'
00:19:01: DDNS: Using server 10.0.0.1
00:19:01: DDNS: Dynamic Update 1: (sending to server 10.0.0.1)
00:19:01: DDNS: Zone = 10.in-addr.arpa
00:19:01: DDNS: Prerequisite: 10.0.0.11.in-addr.arpa. not in use
00:19:01: DDNS: Update: add 10.0.0.11.in-addr.arpa. IN PTR canada_reserved.hacks
00:19:01: DDNS: Zone name for '10.0.0.11.in-addr.arpa.' is '10.in-addr.arpa'
00:19:01: DDNS: Using server 10.0.0.1
00:19:01: DDNS: Dynamic Update 1: (sending to server 10.0.0.1)
00:19:01: DDNS: Zone = 10.in-addr.arpa
00:19:01: DDNS: Prerequisite: 10.0.0.11.in-addr.arpa. not in use
00:19:01: DDNS: Update: add 10.0.0.11.in-addr.arpa. IN PTR canada_reserved.hacks
00:19:01: DDNS: Dynamic DNS Update 1 (PTR) for host canada_reserved.hacks returned 6 (YXDOMAIN)
00:19:01: DDNS: Dynamic Update 2: (sending to server 10.19.192.32)
00:19:01: DDNS: Zone = 10.in-addr.arpa
00:19:01: DDNS: Update: delete 10.0.0.11.in-addr.arpa. all PTR RRs
00:19:01: DDNS: Update: add 10.0.0.11.in-addr.arpa. IN PTR canada_reserved.hacks
00:19:01: DDNS: Dynamic DNS Update 2 (PTR) for host canada_reserved.hacks returned 0 (NOERROR)
00:19:01: DDNS: Zone name for 'canada_reserved.hacks' is 'hacks'
00:19:01: DDNS: Using server 10.19.192.32
00:19:01: DDNS: Dynamic Update 1: (sending to server 10.19.192.32)
00:19:01: DDNS: Zone = hacks
00:19:01: DDNS: Prerequisite: canada_reserved.hacks not in use
00:19:01: DDNS: Update: add canada_reserved.hacks IN A 10.0.0.8
00:19:01: DDNS: Dynamic DNS Update 1 (A) for host canada_reserved.hacks returned 0 (NOERROR)
00:19:01: DDNS: Update of 'canada_reserved.hacks' <=> 10.0.0.8 finished
00:19:01: DYNDNSUPD: Another update completed (total outstanding=2)
00:19:11: DDNS: Dynamic DNS Update 1 (PTR) for host canada_reserved.hacks returned 0 (NOERROR)
00:19:11: DDNS: Dynamic DNS Update 1 (PTR) for host canada_reserved.hacks returned 0 (NOERROR)
00:19:11: DDNS: Zone name for 'canada_reserved.hacks' is 'hacks'
00:19:11: DDNS: Using server 10.0.0.1
00:19:11: DDNS: Dynamic Update 1: (sending to server 10.0.0.1)
00:19:11: DDNS: Zone = hacks
00:19:11: DDNS: Prerequisite: canada_reserved.hacks not in use
00:19:11: DDNS: Update: add canada_reserved.hacks IN A 10.0.0.8
00:19:11: DDNS: Zone name for 'canada_reserved.hacks' is 'hacks'
00:19:11: DDNS: Using server 10.0.0.1
00:19:11: DDNS: Dynamic Update 1: (sending to server 10.0.0.1)
00:19:11: DDNS: Zone = hacks
00:19:11: DDNS: Prerequisite: canada_reserved.hacks not in use
00:19:11: DDNS: Update: add canada_reserved.hacks IN A 10.0.0.8
```

```
00:19:21: DDNS: Dynamic DNS Update 1 (A) for host canada_reserved.hacks returned 0 (NOERROR)
00:19:21: DDNS: Update of 'canada_reserved.hacks' <=> 10.0.0.8 failed
00:19:21: DYNDNSUPD: Another update completed (total outstanding=1)
00:19:21: DDNS: Dynamic DNS Update 1 (A) for host canada_reserved.hacks returned 0 (NOERROR)
00:19:21: DDNS: Update of 'canada_reserved.hacks' <=> 10.0.0.8 failed
00:19:21: DYNDNSUPD: Another update completed (total outstanding=0)
```

Configuring DHCP Support of DDNS Updates

DDNS updates contain information about A or forward RRs for a particular IP address. The IP address is in dotted decimal form, and there must be at least one A record for each host address. The name specified is the hostname expressed as an FQDN (ns.example.com). The PTR or reverse RRs map a domain name to another domain name and is used for reverse mapping (IP address to domain name).

The updates are performed using messages. In general, you will probably want DDNS updates done by the server after the server has sent the ACK response to the DHCP client. Performing the DDNS updates before sending the ACK response will delay the response to the client. Both methods are supported. The default is to do the updates after sending the response.

When looking for a client hostname to use in the update, the server will take the hostname from the FQDN option, if such exists, first. If there is no FQDN option, the server will look for a HOSTNAME option and take the name from there.

If the FQDN or HOSTNAME option is included in subsequent RENEWAL messages, the server will attempt to perform the DDNS update each time the lease is renewed. This process gives the opportunity for the client to change the name specified after the lease has been granted and have the server do the appropriate updates. Although the server has this capability, the DHCP client will continue to use the same hostname throughout the duration of a lease.

The IP address of the server to update is discovered by sending a DNS query for records associated with the hostname to update. If such a record exists, the hostname of the master DNS server is extracted from this information. If no such record exists, the record, which should be included in the response, is used as the authoritative record for the zone where the hostname exists. In either case, once the master DNS server hostname is found, another query for A RRs is sent in order to discover the IP address of this server. The resulting IP address is used for sending updates.

Perform this task to configure the DDNS updates.

Before You Begin

In order for DDNS updates to discover the DNS server, in cases in which the user did not configure the server, the **ip name-server** command should be configured. This name server should be reachable by the system, and the **ip domain lookup** command should be configured (which is the default anyway). In cases in which the configured hostname does not include a period (is not a fully qualified domain name [FQDN]), an IP domain name should be configured.

Note DHCP server-pool configuration commands and interface configurations have precedence over global configurations.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp update dns** [**both**] [**override**] [**before**]
4. **ip dhcp-client update dns** [**server** {**both** | **none**}]
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp update dns** [**both**] [**override**] [**before**] **Example:** `Router(config)# ip dhcp update dns both override`	Enables DDNS updates of PTR RRs for all address pools except those configured with the per-pool **update dns** command, which overrides global configuration. The keywords are as follows: • **both** --(Optional) Enables the DHCP server to perform DDNS updates for A and PTR RRs, unless the DHCP client has specified in the FQDN option that the server should not perform the updates. • **override** --(Optional) Enables the DHCP server to perform DDNS updates for PTR RRs even if the DHCP client has specified in the FQDN option that the server should not perform the updates. **Note** If you specify the **both** and **override** keywords together, this enables the DHCP server to perform DDNS updates for A and PTR RRs overriding anything the DHCP client specified in the FQDN option to the contrary. • **before** --(Optional) Enables the DHCP server to perform DDNS updates before sending the DHCP ACK back to the client. The default is to perform updates after sending the DHCP ACK.
Step 4	**ip dhcp-client update dns** [**server** {**both**	**none**}] **Example:** `Router(config)# ip dhcp-client update dns server both`	Enables DDNS updates of PTR RRs. The optional **server** keyword enables the server to perform DDNS updates for A and PTR RRs. The keywords are as follows: • **both** --Enables the DHCP server to perform DDNS updates for A and PTR RRs, unless the DHCP client specifies in the FQDN option that the server should not perform the updates.

	Command or Action	Purpose
		• **none** --Enables the DHCP client to perform DDNS updates and the server will not perform any updates. The server can override this action. **Note** The **ip dhcp-client update dns server none** command instructs the server not to perform any updates. If configured to do so, the server can override the client. **Note** The **ip dhcp-client update dns server both** command instructs the server to update both the A and PTR RRs.
Step 5	**exit** **Example:** `Router(config)# exit`	Exits to privileged EXEC mode.

Examples

The following example shows how to configure A and PTR RR updates that are performed by the server only:

```
ip dhcp-client update dns server both
ip dhcp update dns both override
```

Configuring DDNS Update Support on Interfaces

Perform this task to configure your interfaces for DDNS update capability.

Note The interface configuration overrides the global configuration.

Before You Begin

In order for DDNS updates to discover the DNS server, in cases in which the user did not configure the server, the **ip name-server** command should be configured. This name server should be reachable by the system, and the **ip domain lookup** command should be configured (which is the default anyway). In cases in which the configured hostname does not include a period (is not a fully qualified domain name [FQDN]), an IP domain name should be configured.

Note The changes will not take effect until any current lease on the interface is released and a new lease is requested that uses a new DHCP DISCOVER packet. This means configuring the **ip address dhcp** command or using the **release dhcp** EXEC command followed by the **renew dhcp** EXEC command.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-type number*
4. **ip dhcp client update dns** [**server** {**both** | **none**}]
5. **ip address dhcp**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-type number* **Example:** `Router(config)# interface ethernet1`	Specifies an interface type and number and enters interface configuration mode.
Step 4	**ip dhcp client update dns** [**server** {**both** | **none**}] **Example:** `Router(config-if)# ip dhcp client update dns server both`	Configures the DHCP client to include an FQDN option when sending packets to the DHCP server. The keywords are as follows: • **both** --(Optional) Enables the DHCP server to perform DDNS updates for A and PTR RRs, unless the DHCP client specifies in the FQDN option that the server should not perform the updates. • **none** --(Optional) Enables the DHCP client to perform DDNS updates and the server will not perform any updates. The server can override this action. **Note** The **ip dhcp client update dns server none** command instructs the server not to perform any updates. If configured to do so, the server can override the client. **Note** The **ip dhcp client update dns server both** command instructs the server to update both the A and PTR RRs.
Step 5	**ip address dhcp**	Releases any current lease on the interface and enables the configuration.

	Command or Action	Purpose
	Example: `Router(config-if)# ip address dhcp`	**Note** You can also release any lease by using the **release dhcp** EXEC command followed by the **renew dhcp** EXEC command.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Exits to privileged EXEC mode.

Configuring a Pool of DHCP Servers to Support DDNS Updates

There are two parts to the DDNS update configuration on the client side. First, if the **ip ddns update method** command is configured on the client, which specifies the DDNS-style updates, then the client will be trying to generate or perform A updates. If the **ip ddns update method ddns both** command is configured, then the client will be trying to update both A and PTR RRs.

Second, the only way for the client to communicate with the server, with reference to what updates it is generating or expecting the server to generate, is to include an FQDN option when communicating with the server. Whether or not this option is included is controlled on the client side by the **ip dhcp-client update dns** command in global configuration mode or the **ip dhcp client update dns** command in interface configuration mode.

If the FQDN option is included in the DHCP interaction, then the client may instruct the server to update “reverse” (the default), “both”, or “none.” Obviously, if the **ip ddns update method** command is configured with the **ddns** and **both**keywords, then the FQDN option configuration should reflect an IP DHCP client update DNS server none, but you have to configure the system correctly.

Finally, even if the client instructs the server to update both or update none, the server can override the client request and do whatever it was configured to do anyway. If there is an FQDN option in the DHCP interaction as above, then server can communicate to the client that it was overridden, in which case the client will not perform the updates because it knows that the server has done the updates. Even if the server is configured to perform the updates after sending the ACK (the default), it can still use the FQDN option to instruct the client what updates it will be performing and thus the client will not do the same types of updates.

If the server is configured with the **update dns** command with or without any keywords, and if the server does not see an FQDN option in the DHCP interaction, then it will assume that the client does not understand DDNS and will automatically act as though it were configured to update both A and PTR RRs on behalf of the client.

Perform this task to configure a pool of DHCP servers to support DDNS updates.

Before You Begin

In order for DDNS updates to discover the DNS server, in cases in which the user did not configure the server, the **ip name-server** command should be configured. This name server should be reachable by the system, and the **ip domain lookup** command should be configured (which is the default anyway). In cases in which

the configured hostname does not include a period (is not a fully qualified domain name [FQDN]), an IP domain name should be configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *pool-name*
4. **update dns** [**both** | **never**] [**override**] [**before**]
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp pool** *pool-name* **Example:** `Router(config)# ip dhcp pool test`	Assigns a name to a DHCP pool and enters DHCP configuration mode.
Step 4	**update dns** [**both** | **never**] [**override**] [**before**] **Example:** `Router(dhcp-config)# update dns never`	Enables DDNS update capability for a pool of DHCP servers for any addresses assigned from this address pool. If the server is configured using this command with or without any of the other keywords, and if the server does not see an FQDN option in the DHCP interaction, then it will assume that the client does not understand DDNS and act as though it were configured to update both A and PTR records on behalf of the client. The keywords are as follows: • **both** --(Optional) Perform forward and reverse updates. If the **before** optional keyword is specified along with the **both** keyword, the server can perform DDNS updates before sending the ACK back to the client. If the **override** optional keyword is specified with the **both** keyword, the server can override the client and update forward and reverse RRs.

	Command or Action	Purpose
		If the **override** and **before** optional keywords are specified with the **both** keyword, the server can override the client (forward and reverse updates) and perform the updates before sending the ACK. • **never** --(Optional) Never perform updates for this pool. • **override** --(Optional) Override the client FQDN flags. If the **before** optional keyword is specified, the updates will be performed before sending the ACK. • **before** --(Optional) Perform updates before sending the ACK.
Step 5	**exit** **Example:** `Router(dhcp-config)# exit`	Exits to global configuration mode.

Examples

The following example shows how to configure a pool of DHCP servers to perform updates for A and PTR RRs before the ACK is sent:

```
ip dhcp pool test
 update dns both before
```

Configuring the Update Method and Interval

Perform this task to specify the update method and interval maximum.

Before You Begin

In order for DDNS updates to discover the DNS server, in cases in which the user did not configure the server, the **ip name-server** command should be configured. This name server should be reachable by the system, and the **ip domain lookup** command should be configured (which is the default anyway). In cases in which the configured hostname does not include a period (is not a fully qualified domain name [FQDN]), an IP domain name should be configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ddns update method** *method-name*
4. **interval minimum** *days hours minutes seconds*
5. **interval maximum** *days hours minutes seconds*
6. **ddns** [**both**]
7. **internal**
8. **http**
9. **add** *url*
10. **remove** *url*
11. exit
12. **exit**
13. **interface** *interface-type number*
14. **ip ddns update hosthame** *hostname*
15. ip ddns update *name*
16. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if pron
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode
Step 3	**ip ddns update method** *method-name* **Example:** `Router(config)# ip ddns update method myupdate`	Specifies the update method nam
Step 4	**interval minimum** *days hours minutes seconds* **Example:** `Router(DDNS-update-method)# interval minimum 1 0 0 0`	Configures a minimum update int • *days* --Range is from 0 to 3 • *hours* --Range is from 0 to 2 • *minutes* --Range is from 0 t

	Command or Action	Purpose
		• *seconds* --Range is from 0 to 59
Step 5	**interval maximum** *days hours minutes seconds* **Example:** `Router(DDNS-update-method)# interval maximum 1 0 0 0`	Configures a maximum update interv • *days* --Range is from 0 to 365. • *hours* --Range is from 0 to 24. • *minutes* --Range is from 0 to 60 • *seconds* --Range is from 0 to 60
Step 6	**ddns** [**both**] **Example:** `Router(DDNS-update-method)# ddns`	Configures DDNS as the update meth updated. **Note** You can specify DDNS or H must disable it by using the configuration, see Steps 7,8,
Step 7	**internal** **Example:** `Router(DDNS-update-method)# internal`	Specifies that an internal cache will b
Step 8	**http** **Example:** `Router(DDNS-update-method)# http`	Configures HTTP as the update meth

Step 9	**add** *url*
	
	Example:
	`Router(DDNS-HTTP)# add`
	`http://test:test@members.dyndns.org/nic/update?system=dyndns&hostname=<h>&myip=<a>`
		Configures a URL that should be invo
		an IP address. The following exampl
		information using DynDNS.org:
		• http://userid:password@members
		You have to enter the URL string abo
		DynDNS.org website. The special cha
		to update and the IP address with wh
		Note Before entering the question
		together on your keyboard. T
		the ? as a help query.

Step 10	**remove** *url*
	
	Example:
	`Router(DDNS-HTTP)# remove`
	`http://test:test@members.dyndns.org/nic/update?system=dyndns&hostname=<h>&myip=<a>`
		Configures a URL that should be inv
		address. The URL takes the same for

	Command or Action	Purpose
Step 11	exit **Example:** `Router(DDNS-HTTP)# exit`	Exits to update-method configura
Step 12	**exit** **Example:** `Router(DDNS-update-method)# exit`	Exits to global configuration mod
Step 13	**interface** *interface-type number* **Example:** `Router(config)# interface ether1`	Enters interface configuration mo
Step 14	**ip ddns update hostname** *hostname* **Example:** `Router(config-if)# ip ddns update hostname abc.dyndns.org`	Specifies a host to be used for the IP address of the interface. The *ho* (for example, DynDNS.org).
Step 15	ip ddns update *name* **Example:** `Router(config-if) ip ddns update myupdate`	Specifies the name of the update address changes on this interface.
Step 16	**exit** **Example:** `Router(config)# exit`	Exits to privileged EXEC mode.

Examples

The following example shows how to configure the update method, the maximum interval of the updates (globally), and configure the hostname on the interface:

```
ip ddns update method mytest
ddns
 http
!Before entering the question mark (?) character in the add http CLI, press the control
(Ctrl) key and the v key together on your keyboard. This will allow you to enter the ?
without the software interpreting the ? as a help query.

 add http://test:test@members.dyndns.org/nic/update?system=dyndns&hostname=<h>&myip=<a>

 interval maximum 1 0 0 0
```

```
 exit
interface ether1

 ip ddns update hostname abc.dyndns.org

 ip ddns update mytest
```

Verifying DDNS Updates

Use the **debug ip ddns update** command to verify that DDNS updates are being performed. There are several sample configurations and the debug output that would display for that scenario.

Sample Configuration #1

The following scenario has a client configured for IETF DDNS updating of A DNS RRs during which a DHCP server is expected to update the PTR DNS RR. The DHCP client discovers the DNS server to update using an SOA RR lookup since the IP address to the server to update is not specified. The DHCP client is configured to include an FQDN DHCP option and notifies the DHCP server that it will be updating the A RRs.

```
!Configure the DHCP Client
ip ddns update method testing
 ddns
interface Ethernet1
 ip dhcp client update dns
 ip ddns update testing
 ip address dhcp
end
!Configure the DHCP Server
ip dhcp pool test
 network 10.0.0.0 255.0.0.0
 update dns
!Enable Debugging
Router# debug ip ddns update
00:14:39:%DHCP-6-ADDRESS_ASSIGN: Interface Ethernet1 assigned DHCP address 10.0.0.4, mask
255.0.0.0, hostname canada_reserved
00:14:39: DYNDNSUPD: Adding DNS mapping for canada_reserved.hacks <=> 10.0.0.4
00:14:39: DYNDNSUPD: Sleeping for 3 seconds waiting for interface Ethernet1 configuration
to settle
00:14:42: DHCPC: Server performed PTR update
00:14:42: DDNS: Enqueuing new DDNS update 'canada_reserved.hacks' <=> 10.0.0.4
00:14:42: DDNS: Zone name for 'canada_reserved.hacks' is 'hacks'
00:14:42: DDNS: Dynamic Update 1: (sending to server 10.19.192.32)
00:14:42: DDNS:   Zone = hacks
00:14:42: DDNS:   Prerequisite: canada_reserved.hacks not in use
00:14:42: DDNS:   Update: add canada_reserved.hacks IN A 10.0.0.4
00:14:42: DDNS: Dynamic DNS Update 1 (A) for host canada_reserved.hacks returned 0 (NOERROR)
00:14:42: DDNS: Update of 'canada_reserved.hacks' <=> 10.0.0.4 finished
00:14:42: DYNDNSUPD: Another update completed (total outstanding=0)
```

Sample Configuration #2

The following scenario has the client configured for IETF DDNS updating of both A and DNS RRs and requesting that the DHCP server update neither. The DHCP client discovers the DNS server to update using an SOA RR lookup since the IP address to the server to update is not specified. The DHCP client is configured to include an FQDN DHCP option that instructs the DHCP server not to update either A or PTR RRs. This is configured using the global version of the command.

```
!Configure the DHCP Client
ip dhcp-client update dns server none
ip ddns update method testing
 ddns both
interface Ethernet1
 ip ddns update testing
```

```
 ip address dhcp
end
!Configure the DHCP Server
ip dhcp pool test
 network 10.0.0.0 255.0.0.0
 update dns
!Enable Debugging
Router# debug ip ddns update
00:15:33:%DHCP-6-ADDRESS_ASSIGN: Interface Ethernet1 assigned DHCP address 10.0.0.5, mask
255.0.0.0, hostname canada_reserved
00:15:33: DYNDNSUPD: Adding DNS mapping for canada_reserved.hacks <=> 10.0.0.5
00:15:33: DYNDNSUPD: Sleeping for 3 seconds waiting for interface Ethernet1 configuration
to settle
00:15:36: DDNS: Enqueuing new DDNS update 'canada_reserved.hacks' <=> 10.0.0.5
00:15:36: DDNS: Zone name for '10.0.0.11.in-addr.arpa.' is '10.in-addr.arpa'
00:15:36: DDNS: Dynamic Update 1: (sending to server 10.19.192.32)
00:15:36: DDNS:   Zone = 10.in-addr.arpa
00:15:36: DDNS:   Prerequisite: 10.0.0.11.in-addr.arpa. not in use
00:15:36: DDNS:   Update: add 10.0.0.11.in-addr.arpa. IN PTR canada_reserved.hacks
00:15:36: DDNS: Dynamic DNS Update 1 (PTR) for host canada_reserved.hacks returned 0 (NOERROR)
00:15:36: DDNS: Zone name for 'canada_reserved.hacks' is 'hacks'
00:15:36: DDNS: Dynamic Update 1: (sending to server 10.19.192.32)
00:15:36: DDNS:   Zone = hacks
00:15:36: DDNS:   Prerequisite: canada_reserved.hacks not in use
00:15:36: DDNS:   Update: add canada_reserved.hacks IN A 10.0.0.5
00:15:36: DDNS: Dynamic DNS Update 1 (A) for host canada_reserved.hacks returned 0 (NOERROR)
00:15:36: DDNS: Update of 'canada_reserved.hacks' <=> 10.0.0.5 finished
00:15:36: DYNDNSUPD: Another update completed (total outstanding=0)
```

Sample Configuration #3

The following scenario the client is configured for IETF DDNS updating of both A and DNS RRs and requesting that the DHCP server update neither. The DHCP client explicitly specifies the server to update. The DHCP client is configured to include an FQDN DHCP option which instructs the DHCP server not to update either A or PTR RRs. This is configured using the global version of the command. The DHCP server is configured to override the client request and update both A and PTR RR anyway.

```
!Configure the DHCP Client
ip dhcp client update dns server non
ip ddns update method testing
 ddns both
interface Ethernet1
 ip dhcp client update dns server none
 ip ddns update testing
 ip address dhcp
end
!Configure the DHCP Server
ip dhcp pool test
 network 10.0.0.0 255.0.0.0
 update dns both override
!Enable Debugging on the DHCP Client
Router# debug ip ddns update
00:16:30:%DHCP-6-ADDRESS_ASSIGN: Interface Ethernet1 assigned DHCP address 10.0.0.6, mask
255.0.0.0, hostname canada_reserved
00:16:30: DYNDNSUPD: Adding DNS mapping for canada_reserved.hacks <=> 10.0.0.6
00:16:30: DYNDNSUPD: Sleeping for 3 seconds waiting for interface Ethernet1 configuration
to settle
00:16:33: DHCPC: Server performed both updates
```

Sample Configuration #4

In the following scenario the client is configured for IETF DDNS updating of both A and DNS RRs and requesting the DHCP server to update neither. The DHCP client explicitly specifies the server to update. The DHCP client is configured to include an FQDN DHCP option which instructs the DHCP server not to update

either A or PTR RRs. This is configured using the global version of the command. The DHCP server is configured to allow the client to update whatever RR it chooses.

```
!Configure the DHCP Client
ip dhcp client update dns server non
ip ddns update method testing
 ddns both
interface Ethernet1
 ip dhcp client update dns server none
 ip ddns update testing host 172.19.192.32
 ip address dhcp
end
!Configure the DHCP Server
ip dhcp pool test
 network 10.0.0.0 255.0.0.0
 update dns
!Enable Debugging on the DHCP Client
Router# debug ip ddns update
00:17:52:%DHCP-6-ADDRESS_ASSIGN: Interface Ethernet1 assigned DHCP address 10.0.0.7, mask
255.0.0.0, hostname canada_reserved
00:17:52: DYNDNSUPD: Adding DNS mapping for canada_reserved.hacks <=> 10.0.0.6
00:17:52: DYNDNSUPD: Sleeping for 3 seconds waiting for interface Ethernet1 configuration
to settle
00:17:55: DDNS: Enqueuing new DDNS update 'canada_reserved.hacks' <=> 10.0.0.7
00:17:55: DYNDNSUPD: Adding DNS mapping for canada_reserved.hacks <=> 10.0.0.7 server
10.19.192.32
00:17:55: DDNS: Enqueuing new DDNS update 'canada_reserved.hacks' <=> 10.0.0.7 server
10.19.192.32
00:17:55: DDNS: Zone name for '10.0.0.11.in-addr.arpa.' is '11.in-addr.arpa'
00:17:55: DDNS: Dynamic Update 1: (sending to server 10.19.192.32)
00:17:55: DDNS: Zone = 10.in-addr.arpa
00:17:55: DDNS: Prerequisite: 10.0.0.11.in-addr.arpa. not in use
00:17:55: DDNS: Update: add 10.0.0.11.in-addr.arpa. IN PTR canada_reserved.hacks
00:17:55: DDNS: Zone name for '10.0.0.11.in-addr.arpa.' is '10.in-addr.arpa'
00:17:55: DDNS: Using server 10.19.192.32
00:17:55: DDNS: Dynamic Update 1: (sending to server 10.19.192.32)
00:17:55: DDNS: Zone = 10.in-addr.arpa
00:17:55: DDNS: Prerequisite: 10.0.0.11.in-addr.arpa. not in use
00:17:55: DDNS: Update: add 10.0.0.11.in-addr.arpa. IN PTR canada_reserved.hacks
00:17:55: DDNS: Dynamic DNS Update 1 (PTR) for host canada_reserved.hacks returned 0 (NOERROR)
00:17:55: DDNS: Dynamic DNS Update 1 (PTR) for host canada_reserved.hacks returned 6
(YXDOMAIN)
00:17:55: DDNS: Dynamic Update 2: (sending to server 10.19.192.32)
00:17:55: DDNS: Zone = 10.in-addr.arpa
00:17:55: DDNS: Update: delete 10.0.0.11.in-addr.arpa. all PTR RRs
00:17:55: DDNS: Update: add 10.0.0.11.in-addr.arpa. IN PTR canada_reserved.hacks
00:17:55: DDNS: Dynamic DNS Update 2 (PTR) for host canada_reserved.hacks returned 0 (NOERROR)
00:17:55: DDNS: Zone name for 'canada_reserved.hacks' is 'hacks'
00:17:55: DDNS: Dynamic Update 1: (sending to server 10.19.192.32)
00:17:55: DDNS: Zone = hacks
00:17:55: DDNS: Prerequisite: canada_reserved.hacks not in use
00:17:55: DDNS: Update: add canada_reserved.hacks IN A 10.0.0.7
00:17:55: DDNS: Dynamic DNS Update 1 (A) for host canada_reserved.hacks returned 0 (NOERROR)
00:17:55: DDNS: Update of 'canada_reserved.hacks' <=> 10.0.0.7 finished
00:17:55: DYNDNSUPD: Another update completed (total outstanding=1)
00:17:55: DDNS: Zone name for 'canada_reserved.hacks' is 'hacks'
00:17:55: DDNS: Using server 10.19.192.32
00:17:55: DDNS: Dynamic Update 1: (sending to server 10.19.192.32)
00:17:55: DDNS: Zone = hacks
00:17:55: DDNS: Prerequisite: canada_reserved.hacks not in use
00:17:55: DDNS: Update: add canada_reserved.hacks IN A 10.0.0.7
00:17:55: DDNS: Dynamic DNS Update 1 (A) for host canada_reserved.hacks returned 6 (YXDOMAIN)
00:17:55: DDNS: Dynamic Update 2: (sending to server 10.19.192.32)
00:17:55: DDNS: Zone = hacks
00:17:55: DDNS: Update: delete canada_reserved.hacks all A RRs
00:17:55: DDNS: Update: add canada_reserved.hacks IN A 10.0.0.7
00:17:55: DDNS: Dynamic DNS Update 2 (A) for host canada_reserved.hacks returned 0 (NOERROR)
00:17:55: DDNS: Update of 'canada_reserved.hacks' <=> 10.0.0.7 finished
00:17:55: DYNDNSUPD: Another update completed (total outstanding=0)
```

Sample Configuration #5

In the following scenario, the debug output is displaying internal host table updates when the default domain name is “hacks.” The “test” update method specifies that the internal Cisco IOS host table should be updated. Configuring the update method as “test” should be used when the address on the Ethernet 0/0 interface changes. The hostname is configured for the update on this interface.

```
ip domain name hacks
ip ddns update method test
 internal
interface ethernet0/0
 ip ddns update test hostname test2
 ip addr dhcp
!Enable Debugging
Router# debug ip ddns update
*Jun 4 03:11:10.591:%DHCP-6-ADDRESS_ASSIGN: Interface Ethernet0/0 assigned DHCP address
10.0.0.5, mask 255.0.0.0, hostname test2
*Jun 4 03:11:10.591: DYNDNSUPD: Adding DNS mapping for test2.hacks <=> 10.0.0.5
*Jun 4 03:11:10.591: DYNDNSUPD: Adding internal mapping test2.hacks <=> 10.0.0.5
```

Using the **show hosts** command displays the newly added host table entry.

```
Router# show hosts
Default domain is hacks
Name/address lookup uses domain service
Name servers are 255.255.255.255
Codes: UN - unknown, EX - expired, OK - OK,?? - revalidate
       temp - temporary, perm - permanent
       NA - Not Applicable None - Not defined
Host                      Port Flags      Age Type   Address(es)
test2.hacks               None (perm, OK) 0   IP    10.0.0.5
```

Shutting down the interface removes the host table entry.

```
interface ethernet0/0
 shutdown
*Jun 4 03:14:02.107: DYNDNSUPD: Removing DNS mapping for test2.hacks <=> 10.0.0.5
*Jun 4 03:14:02.107: DYNDNSUPD: Removing mapping test2.hacks <=> 10.0.0.5
```

The **show hosts** command output shows the entry has been removed.

```
Router# show hosts
Default domain is hacks
Name/address lookup uses domain service
Name servers are 255.255.255.255
Codes: UN - unknown, EX - expired, OK - OK,?? - revalidate
       temp - temporary, perm - permanent
       NA - Not Applicable None - Not defined
Host                      Port Flags      Age Type   Address(es)
```

Sample Configuration #6

In the following scenario, the debug output shows the HTTP-style DDNS updates. The sample configuration defines a new IP DDNS update method named dyndns that configures a URL to use when adding or changing an address. No URL has been defined for use when removing an address since DynDNS.org does not use such a URL for free accounts. A maximum update interval of 28 days has been configured, so specifying that updates should be sent at least every 28 days. Configuring the new dyndns update method should be used for Ethernet interface .

Note Before entering the question mark (?) character in the “add http” configuration after the **update** keyword, press the control (Ctrl) key and the “v” key together on your keyboard. This will allow you to enter the ? without the software interpreting it as a help query.

```
!Configure the DHCP Client
ip ddns update method dyndns
 http
    add http://test:test@<s>/nic/update?system=dyndns&hostname=<h>&myip=<a>
    interval max 28 0 0 0
interface ethernet1
 ip ddns update hostname test.dyndns.org
 ip ddns update dyndns host members.dyndns.org
 ip addr dhcp
!Enable Debugging
Router# debug ip ddns update
00:04:35:%DHCP-6-ADDRESS_ASSIGN: Interface Ethernet1 assigned DHCP address 10.32.254.187,
mask 255.255.255.240, hostname test.dyndns.org
00:04:35: DYNDNSUPD: Adding DNS mapping for test.dyndns.org <=> 10.32.254.187 server
10.208.196.94
00:04:35: DYNDNSUPD: Sleeping for 3 seconds waiting for interface Ethernet1 configuration
to settle
00:04:38: HTTPDNS: Update add called for test.dyndns.org <=> 10.32.254.187
00:04:38: HTTPDNS: Update called for test.dyndns.org <=> 10.32.254.187
00:04:38: HTTPDNS: init
00:04:38: HTTPDNSUPD: Session ID = 0x7
00:04:38: HTTPDNSUPD: URL =
'http://test:test@10.208.196.94/nic/update?system=dyndns&hostname=test.dyndns.org&myip=10.32.254.187'
00:04:38: HTTPDNSUPD: Sending request
00:04:40: HTTPDNSUPD: Response for update test.dyndns.org <=> 10.32.254.187
00:04:40: HTTPDNSUPD: DATA START
good 10.32.254.187
00:04:40: HTTPDNSUPD: DATA END, Status is Response data received, successfully
00:04:40: HTTPDNSUPD: Call returned SUCCESS for update test.dyndns.org <=> 10.32.254.187
00:04:40: HTTPDNSUPD: Freeing response
00:04:40: DYNDNSUPD: Another update completed (outstanding=0, total=0)
00:04:40: HTTPDNSUPD: Clearing all session 7 info
!28 days later, the automatic update happens.
00:05:39: DYNDNSUPD: Adding DNS mapping for test.dyndns.org <=> 10.32.254.187 server
10.208.196.94
00:05:39: HTTPDNS: Update add called for test.dyndns.org <=> 10.32.254.187
00:05:39: HTTPDNS: Update called for test.dyndns.org <=> 10.32.254.187
00:05:39: HTTPDNS: init
00:05:39: HTTPDNSUPD: Session ID = 0x8
00:05:39: HTTPDNSUPD: URL =
'http://test:test@10.208.196.94/nic/update?system=dyndns&hostname=test.dyndns.org&myip=10.32.254.187'
00:05:39: HTTPDNSUPD: Sending request
00:05:39: HTTPDNSUPD: Response for update test.dyndns.org <=> 10.32.254.187
00:05:39: HTTPDNSUPD: DATA START
nochg 10.32.254.187
00:05:39: HTTPDNSUPD: DATA END, Status is Response data received, successfully
00:05:39: HTTPDNSUPD: Call returned SUCCESS for update test.dyndns.org <=> 10.32.254.187
00:05:39: HTTPDNSUPD: Freeing response
00:05:39: DYNDNSUPD: Another update completed (outstanding=0, total=0)
00:05:39: HTTPDNSUPD: Clearing all session 8 info
```

Configuration Examples for Dynamic DNS Support for Cisco IOS Software

Configuration of the DHCP Client Example

The following example shows that no DDNS updates will be performed for addresses assigned from the address pool “abc.” Addresses allocated from the address pool “def” will have both forward (A) and reverse (PTR) updates performed. This configuration has precedence over the global server configurations.

```
ip dhcp update dns both override
 ip dhcp pool abc
  network 10.1.0.0 255.255.0.0
!
 update dns never
!
 ip dhcp pool def
  network 10.10.0.0 255.255.0.0
```

Configuration of the DHCP Server Example

The following example shows how to configure A and PTR RR updates that are performed by the server only:

```
ip dhcp-client update dns server both

ip dhcp update dns both override
```

Configuration of the HTTP Updates Example

The following example shows how to configure a PPPoE server for HTTP DDNS:

```
!Username and Password for PPP Authentication Configuration
!
username user1 password 0 cisco
!
!DHCP Pool Configuration
ip dhcp pool mypool
 network 10.10.10.0 255.255.255.0
 default-router 10.10.10.1
!
!VPDN configuration for PPPoE
vpdn enable
!
vpdn-group pppoe
accept-dialin
protocol pppoe
virtual-template 1
!
interface Loopback0
ip address 10.10.10.1 255.255.255.0
!
!Port used to connect to the Internet, it can be the same port that is under test, but to
make the test clear and simple these two are separated.
!
interface FastEthernet0/0
 ip address 10.0.58.71 255.255.255.0
```

```
!
!Port under test.
!
interface FastEthernet0/1
 no ip address
 pppoe enable
!
!Virtual template and address pool config for PPPoE.
interface Virtual-Template1
 ip unnumbered Loopback0
 ip mtu 1492
 peer default ip address dhcp-pool mypool
 ppp authentication chap
```

The following example shows how to configure a DHCP client for IETF DDNS:

```
!Default hostname of the router.
hostname mytest
!
!Default domain name on the router.
ip domain name test.com
!
!Port under test.
!
interface FastEthernet0/1
 no ip address (configured to “ip address dhcp”)
```

The following example shows how to configure the method of update and the maximum interval of the updates (globally) and configure the hostname on the interface:

Note Before entering the question mark (?) character in the “add http” configuration after the **update** keyword, press the control (Ctrl) key and the “v” key together on your keyboard. This will allow you to enter the ? without the software interpreting it as a help query.

```
ip ddns update method mytest
ddns
 http

 add http://test:test@members.dyndns.org/nic/update?system=dyndns&hostname=<h>&myip=<a>

 interval maximum 1 0 0 0
 exit
interface ether1

 ip ddns update hostname abc.dyndns.org

 ip ddns update mytest
```

The following are examples of URLs that can be used to update some HTTP DNS update services. These URLs are correct to the best of the knowledge of Cisco but have not been tested in all cases. Where the word “USERNAME:” appears in the URL, the customer account username at the HTTP site should be used.

Where the word “PASSWORD” appears in the URL, the customer password for that account should be used:

Note Before entering the question mark (?) character in the “add http” configuration after the **update** keyword, press the control (Ctrl) key and the “v” key together on your keyboard. This will allow you to enter the ? without the software interpreting it as a help query.

DDNS

```
http://USERNAME:PASSWORD@members.dyndns.org/nic/update?system=dyndns&hostname=<h>&myip=<a>
!Requires “interval max 28 0 0 0" in the update method definition.
```

TZO

```
http://cgi.tzo.com/webclient/signedon.html?TZOName=<h>&Email=USERNAME&TZOKey=PASSWORD&IPAddress=<a>
```

EASYDNS

```
http://USERNAME:PASSWORD@members.easydns.com/dyn/ez-ipupdate.php?action=edit&myip=<a>&host_id=<h>
```

JUSTLINUX

```
http://USERNAME:PASSWORD@www.justlinux.com/bin/controlpanel/dyndns/jlc.pl?direst=1&username=USERNAME&password=PASSWORD&host=<h>&ip=<a>
```

DYNS

```
http://USERNAME:PASSWORD@www.dyns.cx/postscript.php?username=USERNAME&password=PASSWORD&host=<h>&ip=<a>
```

HN

```
http://USERNAME:PASSWORD@dup.hn.org/vanity/update?ver=1&IP=<a>
```

ZONEEDIT

```
http://USERNAME:PASSWORD@www.zoneedit.com/auth/dynamic.html?host=<h>&dnsto=<a>
```

Note Because these services are provided by the respective companies, the URLs may be subject to change or the service could be discontinued at any time. Cisco takes no responsibility for the accuracy or use of any of this information. The URLs were obtained using an application called “ez-ipupdate,” which is available for free on the Internet.

Additional References

The following sections provide references related to the Dynamic DNS Support for Cisco IOS Software feature.

Related Documents

Related Topic	Document Title
DNS Configuration Tasks	“Configuring DNS” module
DNS commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2136	*Dynamic Updates in the Domain Name System (DNS Update)*
RFC 3007	*Secure Domain Name System (DNS) Dynamic Update*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Dynamic DNS Support for Cisco IOS Software

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1: Feature Information for Dynamic DNS Support for Cisco IOS Software

Feature Name	Releases	Feature Information
Dynamic DNS Support for Cisco IOS Software	12.3(8)YA 12.3(14)T	The Dynamic DNS Support for Cisco IOS Software feature enables Cisco IOS software devices to perform Dynamic Domain Name System (DDNS) updates to ensure that an IP host DNS name is correctly associated with its IP address.