NAME

dhcp-options - Dynamic Host Configuration Protocol options

DESCRIPTION

The Dynamic Host Configuration protocol allows the client to receive **options** from the DHCP server describing the network configuration and various services that are available on the network. When configuring dhcpd(8) or dhclient(8), options must often be declared. The syntax for declaring options, and the names and formats of the options that can be declared, are documented here.

REFERENCE: OPTION STATEMENTS

DHCP **option** statements always start with the **option** keyword, followed by an option name, followed by option data. The option names and data formats are described below. It is not necessary to exhaustively specify all DHCP options - only those options which are needed by clients must be specified.

Option data comes in a variety of formats, as defined below:

The *ip-address* data type can be entered either as an explicit IP address (e.g., 239.254.197.10) or as a domain name (e.g., haagen.isc.org). A domain name must resolve to a single IP address.

The *int32* data type specifies a signed 32-bit integer. The *uint32* data type specifies an unsigned 32-bit integer. The *int16* and *uint16* data types specify signed and unsigned 16-bit integers. The *int8* and *uint8* data types specify signed and unsigned 8-bit integers. Unsigned 8-bit integers are also sometimes referred to as octets.

The *string* data type specifies an NVT (Network Virtual Terminal) ASCII string, which must be enclosed in double quotes - for example, to specify a domain-name option, the syntax would be

```
option domain-name "isc.org";
```

The *flag* data type specifies a boolean value. Booleans can be either true or false (or on or off, if that makes more sense to you).

The *data-string* data type specifies either an NVT ASCII string enclosed in double quotes, or a series of octets specified in hexadecimal, separated by colons. For example:

```
option dhcp-client-identifier "CLIENT-FOO";
```

or

```
option dhcp-client-identifier 43:4c:49:45:54:2d:46:4f:4f;
```

The documentation for the various options mentioned below is taken from the IETF draft document on

DHCP options, RFC 2132. Options which are not listed by name may be defined by the name option-*nnn*, where *nnn* is the decimal number of the option code. These options may be followed either by a string, enclosed in quotes, or by a series of octets, expressed as two-digit hexadecimal numbers separated by colons. For example:

```
option option-133 "my-option-133-text";
option option-129 1:54:c9:2b:47;
```

Because dhcpd(8) does not know the format of these undefined option codes, no checking is done to ensure the correctness of the entered data.

The standard options are:

RFC 1497 Vendor Extensions

option subnet-mask *ip-address*;

The **subnet-mask** option specifies the client's subnet mask as per RFC 950. If no subnet-mask option is provided anywhere in scope, as a last resort dhcpd(8) will use the subnet mask from the subnet declaration for the network on which an address is being assigned. However, *any* subnet-mask option declaration that is in scope for the address being assigned will override the subnet mask specified in the subnet declaration.

option time-offset *int32*;

The **time-offset** option specifies the offset of the client's subnet in seconds from Coordinated Universal Time (UTC).

option routers *ip-address* [, *ip-address* ...];

The **routers** option specifies a list of IP addresses for routers on the client's subnet. Routers should be listed in order of preference.

option time-servers *ip-address* [, *ip-address* ...];

The **time-server** option specifies a list of RFC 868 time servers available to the client. Servers should be listed in order of preference.

option ien116-name-servers *ip-address* [, *ip-address* ...];

The **ien116-name-servers** option specifies a list of IEN 116 name servers available to the client. Servers should be listed in order of preference.

option domain-name-servers *ip-address* [, *ip-address* ...];

The **domain-name-servers** option specifies a list of Domain Name System (STD 13, RFC 1035) name servers available to the client. Servers should be listed in order of preference.

option log-servers *ip-address* [, *ip-address* ...];
The **log-servers** option specifies a list of MIT-LCS UDP log servers available to the client. Servers should be listed in order of preference.

option cookie-servers *ip-address* [, *ip-address* ...];
The **cookie-servers** option specifies a list of RFC 865 cookie servers available to the client. Servers should be listed in order of preference.

option lpr-servers *ip-address* [, *ip-address* ...];
The **lpr-servers** option specifies a list of RFC 1179 line printer servers available to the client. Servers should be listed in order of preference.

option impress-servers *ip-address* [, *ip-address* ...];
The **impress-servers** option specifies a list of Imagen Impress servers available to the client. Servers should be listed in order of preference.

option resource-location-servers *ip-address* [, *ip-address* ...];
This option specifies a list of RFC 887 Resource Location servers available to the client. Servers should be listed in order of preference.

option host-name *string*;
This option specifies the name of the client. The name may or may not be qualified with the local domain name (it is preferable to use the **domain-name** option to specify the domain name). See RFC 1035 for character set restrictions.

option boot-size *uint16*;
This option specifies the length in 512-octet blocks of the default boot image for the client.

option merit-dump *string*;
This option specifies the pathname of a file to which the client's core image should be dumped in the event the client crashes. The path is formatted as a character string consisting of characters from the NVT ASCII character set.

option domain-name *string*;
This option specifies the domain name that the client should use when resolving hostnames via the Domain Name System.

option domain-search *string*;
This option specifies a list of domain names that the client should use when resolving hostnames via the Domain Name System. This option is defined in RFC 3397.

option swap-server *ip-address*;
This specifies the IP address of the client's swap server.

option root-path *string*;
This option specifies the pathname that contains the client's root disk. The path is formatted as a character string consisting of characters from the NVT ASCII character set.

IP Layer Parameters per Host

option ip-forwarding *flag*;
This option specifies whether the client should configure its IP layer for packet forwarding. A value of 0 means disable IP forwarding, and a value of 1 means enable IP forwarding.

option non-local-source-routing *flag*;
This option specifies whether the client should configure its IP layer to allow forwarding of datagrams with non-local source routes (see Section 3.3.5 of [4] for a discussion of this topic). A value of 0 means disallow forwarding of such datagrams, and a value of 1 means allow forwarding.

option policy-filter *ip-address ip-address* [, *ip-address ip-address* ...];
This option specifies policy filters for non-local source routing. The filters consist of a list of IP addresses and masks which specify destination/mask pairs with which to filter incoming source routes.

Any source-routed datagram whose next-hop address does not match one of the filters should be discarded by the client.

See STD 3 (RFC 1122) for further information.

option max-dgram-reassembly *uint16*;
This option specifies the maximum size datagram that the client should be prepared to reassemble. The minimum legal value is 576.

option default-ip-ttl *uint8*;
This option specifies the default time-to-live that the client should use on outgoing datagrams.

option path-mtu-aging-timeout *uint32*;
This option specifies the timeout (in seconds) to use when aging Path MTU values discovered by the mechanism defined in RFC 1191.

option path-mtu-plateau-table *uint16* [, *uint16* ...];

This option specifies a table of MTU sizes to use when performing Path MTU Discovery as defined in RFC 1191. The table is formatted as a list of 16-bit unsigned integers, ordered from smallest to largest. The minimum MTU value cannot be smaller than 68.

IP Layer Parameters per Interface

option interface-mtu *uint16*;

This option specifies the MTU to use on this interface. The minimum legal value for the MTU is 68.

option all-subnets-local *flag*;

This option specifies whether or not the client may assume that all subnets of the IP network to which the client is connected use the same MTU as the subnet of that network to which the client is directly connected. A value of 1 indicates that all subnets share the same MTU. A value of 0 means that the client should assume that some subnets of the directly connected network may have smaller MTUs.

option broadcast-address *ip-address*;

This option specifies the broadcast address in use on the client's subnet. Legal values for broadcast addresses are specified in section 3.2.1.3 of STD 3 (RFC 1122).

option perform-mask-discovery *flag*;

This option specifies whether or not the client should perform subnet mask discovery using ICMP. A value of 0 indicates that the client should not perform mask discovery. A value of 1 means that the client should perform mask discovery.

option mask-supplier *flag*;

This option specifies whether or not the client should respond to subnet mask requests using ICMP. A value of 0 indicates that the client should not respond. A value of 1 means that the client should respond.

option router-discovery *flag*;

This option specifies whether or not the client should solicit routers using the Router Discovery mechanism defined in RFC 1256. A value of 0 indicates that the client should not perform router discovery. A value of 1 means that the client should perform router discovery.

option router-solicitation-address *ip-address*;

This option specifies the address to which the client should transmit router solicitation requests.

option static-routes *ip-address ip-address* [, *ip-address ip-address ...*];

This option specifies a list of static routes that the client should install in its routing cache. If

multiple routes to the same destination are specified, they are listed in descending order of priority.

The routes consist of a list of IP address pairs. The first address is the destination address, and the second address is the router for the destination.

The default route (0.0.0.0) is an illegal destination for a static route. To specify the default route, use the **routers** option.

Link Layer Parameters per Interface

option trailer-encapsulation *flag*;

This option specifies whether or not the client should negotiate the use of trailers (RFC 893 [14]) when using the ARP protocol. A value of 0 indicates that the client should not attempt to use trailers. A value of 1 means that the client should attempt to use trailers.

option arp-cache-timeout *uint32*;

This option specifies the timeout in seconds for ARP cache entries.

option ieee802-3-encapsulation *flag*;

This option specifies whether or not the client should use Ethernet Version 2 (RFC 894) or IEEE 802.3 (RFC 1042) encapsulation if the interface is an Ethernet. A value of 0 indicates that the client should use RFC 894 encapsulation. A value of 1 means that the client should use RFC 1042 encapsulation.

TCP Parameters

option default-tcp-ttl *uint8*;

This option specifies the default TTL that the client should use when sending TCP segments. The minimum value is 1.

option tcp-keepalive-interval *uint32*;

This option specifies the interval (in seconds) that the client TCP should wait before sending a keepalive message on a TCP connection. The time is specified as a 32-bit unsigned integer. A value of zero indicates that the client should not generate keepalive messages on connections unless specifically requested by an application.

option tcp-keepalive-garbage *flag*;

This option specifies whether or not the client should send TCP keepalive messages with an octet of garbage for compatibility with older implementations. A value of 0 indicates that a garbage octet should not be sent. A value of 1 indicates that a garbage octet should be sent.

Application and Service Parameters

option nis-domain *string*;

This option specifies the name of the client's NIS (Sun Network Information Services) domain. The domain is formatted as a character string consisting of characters from the NVT ASCII character set.

option nis-servers *ip-address* [, *ip-address* ...];

This option specifies a list of IP addresses indicating NIS servers available to the client. Servers should be listed in order of preference.

option ntp-servers *ip-address* [, *ip-address* ...];

This option specifies a list of IP addresses indicating NTP (RFC 1305) servers available to the client. Servers should be listed in order of preference.

option netbios-name-servers *ip-address* [, *ip-address* ...];

The NetBIOS name server (NBNS) option specifies a list of RFC 1001/1002 NBNS name servers listed in order of preference. NetBIOS Name Service is currently more commonly referred to as WINS. WINS servers can be specified using the **netbios-name-servers** option.

option netbios-dd-server *ip-address* [, *ip-address* ...];

The NetBIOS datagram distribution server (NBDD) option specifies a list of RFC 1001/1002 NBDD servers listed in order of preference.

option netbios-node-type *uint8*;

The NetBIOS node type option allows NetBIOS over TCP/IP clients which are configurable to be configured as described in RFC 1001/1002. The value is specified as a single octet which identifies the client type.

Possible node types are:

1 B-node: Broadcast - no WINS

2 P-node: Peer - WINS only

4 M-node: Mixed - broadcast, then WINS

8 H-node: Hybrid - WINS, then broadcast

option netbios-scope *string*;

The NetBIOS scope option specifies the NetBIOS over TCP/IP scope parameter for the client as

specified in RFC 1001/1002. See RFC 1001, RFC 1002, and RFC 1035 for character-set restrictions.

option font-servers *ip-address* [, *ip-address* ...];
This option specifies a list of X Window System Font servers available to the client. Servers should be listed in order of preference.

option x-display-manager *ip-address* [, *ip-address* ...];
This option specifies a list of systems that are running the X Window System Display Manager and are available to the client. Addresses should be listed in order of preference.

option dhcp-client-identifier *data-string*;
This option can be used to specify a DHCP client identifier in a host declaration, so that dhcpd(8) can find the host record by matching against the client identifier.

option nisplus-domain *string*;
This option specifies the name of the client's NIS+ domain. The domain is formatted as a character string consisting of characters from the NVT ASCII character set.

option nisplus-servers *ip-address* [, *ip-address* ...];
This option specifies a list of IP addresses indicating NIS+ servers available to the client. Servers should be listed in order of preference.

option tftp-server-name *string*;
This option is used to identify a TFTP server and, if supported by the client, should have the same effect as the **server-name** declaration. BOOTP clients are unlikely to support this option. Some DHCP clients will support it, and others actually require it.

option bootfile-name *string*;
This option is used to identify a bootstrap file. If supported by the client, it should have the same effect as the **filename** declaration. BOOTP clients are unlikely to support this option. Some DHCP clients will support it, and others actually require it.

option mobile-ip-home-agent *ip-address* [, *ip-address* ...];
This option specifies a list of IP addresses indicating mobile IP home agents available to the client. Agents should be listed in order of preference, although normally there will be only one such agent.

option smtp-server *ip-address* [, *ip-address* ...];
The **smtp-server** option specifies a list of SMTP servers available to the client. Servers should be

listed in order of preference.

option pop-server *ip-address* [, *ip-address* ...];
The **pop-server** option specifies a list of POP3 servers available to the client. Servers should be listed in order of preference.

option nntp-server *ip-address* [, *ip-address* ...];
The **nntp-server** option specifies a list of NNTP servers available to the client. Servers should be listed in order of preference.

option www-server *ip-address* [, *ip-address* ...];
The **www-server** option specifies a list of WWW servers available to the client. Servers should be listed in order of preference.

option finger-server *ip-address* [, *ip-address* ...];
The **finger-server** option specifies a list of finger(1) servers available to the client. Servers should be listed in order of preference.

option irc-server *ip-address* [, *ip-address* ...];
The **irc-server** option specifies a list of IRC servers available to the client. Servers should be listed in order of preference.

option streettalk-server *ip-address* [, *ip-address* ...];
The **streettalk-server** option specifies a list of StreetTalk servers available to the client. Servers should be listed in order of preference.

option streettalk-directory-assistance-server *ip-address* [, *ip-address* ...];
The StreetTalk Directory Assistance (STDA) server option specifies a list of STDA servers available to the client. Servers should be listed in order of preference.

option url *string*;
This option specifies the URL that the client may use when using UEFI boot from a HTTP server.

SEE ALSO

dhclient.conf(5), dhcpd.conf(5), dhcpd.leases(5), dhclient(8), dhcpd(8)

RFC 2131, RFC 2132, RFC 3769.

AUTHORS

The dhcpd(8) utility was written by Ted Lemon <*mellon@vix.com*> under a contract with Vixie Labs.

The current implementation was reworked by Henning Brauer <*henning@openbsd.org*>.