[MS-DHCPE]:

Dynamic Host Configuration Protocol (DHCP) Extensions

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights**. This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.
- **Patents**. Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft Open Specification Promise or the Community Promise. If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **Trademarks**. The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names**. The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications do not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them. Certain Open Specifications are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

Revision Summary

Date	Revision History	Revision Class	Comments
12/18/2006	0.1		Version 0.1 release
3/2/2007	1.0		Version 1.0 release
4/3/2007	1.1		Version 1.1 release
5/11/2007	1.2		Version 1.2 release
6/1/2007	2.0	Major	Updated and revised the technical content.
7/3/2007	3.0	Major	Updated and revised the technical content.
7/20/2007	4.0	Major	Updated and revised the technical content.
8/10/2007	5.0	Major	Updated and revised the technical content.
9/28/2007	6.0	Major	Updated and revised the technical content.
10/23/2007	7.0	Major	Updated and revised the technical content.
11/30/2007	7.0.1	Editorial	Changed language and formatting in the technical content.
1/25/2008	7.0.2	Editorial	Changed language and formatting in the technical content.
3/14/2008	7.0.3	Editorial	Changed language and formatting in the technical content.
5/16/2008	7.0.4	Editorial	Changed language and formatting in the technical content.
6/20/2008	7.1	Minor	Clarified the meaning of the technical content.
7/25/2008	7.2	Minor	Clarified the meaning of the technical content.
8/29/2008	7.2.1	Editorial	Changed language and formatting in the technical content.
10/24/2008	7.2.2	Editorial	Changed language and formatting in the technical content.
12/5/2008	8.0	Major	Updated and revised the technical content.
1/16/2009	8.1	Minor	Clarified the meaning of the technical content.
2/27/2009	8.1.1	Editorial	Changed language and formatting in the technical content.
4/10/2009	8.2	Minor	Clarified the meaning of the technical content.
5/22/2009	8.2.1	Editorial	Changed language and formatting in the technical content.
7/2/2009	8.3	Minor	Clarified the meaning of the technical content.
8/14/2009	8.4	Minor	Clarified the meaning of the technical content.
9/25/2009	8.5	Minor	Clarified the meaning of the technical content.
11/6/2009	9.0	Major	Updated and revised the technical content.
12/18/2009	10.0	Major	Updated and revised the technical content.
1/29/2010	11.0	Major	Updated and revised the technical content.
3/12/2010	11.0.1	Editorial	Changed language and formatting in the technical content.

Copyright © 2015 Microsoft Corporation

Date	Revision History	Revision Class	Comments
4/23/2010	11.0.2	Editorial	Changed language and formatting in the technical content.
6/4/2010	12.0	Major	Updated and revised the technical content.
7/16/2010	13.0	Major	Updated and revised the technical content.
8/27/2010	13.0	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2010	13.0	None	No changes to the meaning, language, or formatting of the technical content.
11/19/2010	14.0	Major	Updated and revised the technical content.
1/7/2011	15.0	Major	Updated and revised the technical content.
2/11/2011	16.0	Major	Updated and revised the technical content.
3/25/2011	16.1	Minor	Clarified the meaning of the technical content.
5/6/2011	16.1	None	No changes to the meaning, language, or formatting of the technical content.
6/17/2011	16.2	Minor	Clarified the meaning of the technical content.
9/23/2011	16.2	None	No changes to the meaning, language, or formatting of the technical content.
12/16/2011	17.0	Major	Updated and revised the technical content.
3/30/2012	17.1	Minor	Clarified the meaning of the technical content.
7/12/2012	18.0	Major	Updated and revised the technical content.
10/25/2012	19.0	Major	Updated and revised the technical content.
1/31/2013	19.0	None	No changes to the meaning, language, or formatting of the technical content.
8/8/2013	20.0	Major	Updated and revised the technical content.
11/14/2013	21.0	Major	Updated and revised the technical content.
2/13/2014	21.0	None	No changes to the meaning, language, or formatting of the technical content.
5/15/2014	21.0	None	No changes to the meaning, language, or formatting of the technical content.
6/30/2015	22.0	Major	Significantly changed the technical content.
10/16/2015	22.0	No Change	No changes to the meaning, language, or formatting of the technical content.

Copyright © 2015 Microsoft Corporation

Table of Contents

Copyright © 2015 Microsoft Corporation

Copyright © 2015 Microsoft Corporation

Copyright © 2015 Microsoft Corporation

1 Introduction

The Dynamic Host Configuration Protocol (DHCP) is an Internet Engineering Task Force (IETF) standard protocol designed to provide a framework for passing configuration information to hosts on a TCP/IP network. See [RFC2131] section 1 for an introduction to this protocol.

This document specifies a set of vendor-specific options, nonstandard options for DHCP, and a set of vendor-specific options, which can be used to authorize a **DHCP server**.

Sections 1.8, 2, and 3 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in [RFC2119]. Sections 1.5 and 1.9 are also normative but do not contain those terms. All other sections and examples in this specification are informative.

1.1 Glossary

The following terms are specific to this document:

Active Directory: A general-purpose network directory service. **Active Directory** also refers to the Windows implementation of a directory service. **Active Directory** stores information about a variety of objects in the network. Importantly, user accounts, computer accounts, groups, and all related credential information used by the Windows implementation of Kerberos are stored in **Active Directory**. **Active Directory** is either deployed as **Active Directory Domain Services (AD DS)** or Active Directory Lightweight Directory Services (AD LDS). [MS-ADTS] describes both forms. For more information, see [MS-AUTHSOD] section 1.1.1.5.2, **Lightweight Directory Access Protocol (LDAP)** versions 2 and 3, Kerberos, and **DNS**.

Active Directory Domain Services (AD DS): A directory service (DS) implemented by a domain controller (DC). The DS provides a data store for objects that is distributed across multiple DCs. The DCs interoperate as peers to ensure that a local change to an object replicates correctly across DCs. For more information, see [MS-AUTHSOD] section 1.1.1.5.2 and [MS-ADTS]. For information about product versions, see [MS-ADTS] section 1. See also **Active Directory**.

Administratively Authorized Server: A **DHCP server** that has been explicitly authorized by an administrator.

ADsPath: An LDAP string representation of distinguished names.

canonical IDNA: A domain name string is said to be encoded in **canonical IDNA** form when the Unicode string is first encoded in canonical form as described in [RFC1035] section 3 and then the resulting string is converted using IDNA.

Classless Static Route: A DHCP option that provides a subnet mask for each entry so that the subnet mask can be other than what would be determined by using the algorithm specified in Internet Protocol STD 5 [RFC791]and Internet Standard Subnetting Procedure STD 5 [RFC950].

client: A computer on which the remote procedure call (RPC) client is executing.

code page: An ordered set of characters of a specific script in which a numerical index (code-point value) is associated with each character. Code pages are a means of providing support for character sets (1) and keyboard layouts used in different countries. Devices such as the display and keyboard can be configured to use a specific code page and to switch from one code page (such as the United States) to another (such as Portugal) at the user's request.

DHCP client: The remote procedure call (RPC) **clients** that use the Dynamic Host Configuration Protocol Server Management Protocol (DHCPM) to configure, manage, and monitor the Dynamic Host Configuration Protocol (DHCP) **server**.

Copyright © 2015 Microsoft Corporation

DHCPv4: A Dynamic Host Configuration Protocol (DHCP) client that runs over the Internet Protocol version 4 (IPv4).

DHCPv6: DHCP over IPv6 protocol.

Domain Name System (DNS): A hierarchical, distributed database that contains mappings of domain names (1) to various types of data, such as IP addresses. DNS enables the location of computers and services by user-friendly names, and it also enables the discovery of other information stored in the database.

Dynamic Host Configuration Protocol (DHCP): A protocol that provides a framework for passing configuration information to hosts on a TCP/IP network, as described in [RFC2131].

Dynamic Host Configuration Protocol (DHCP) client: An Internet host using DHCP to obtain configuration parameters such as network addresses.

Dynamic Host Configuration Protocol (DHCP) server: A computer running a DHCP service that offers dynamic configuration of IP addresses and related information to DHCP-enabled **clients**.

Internationalized Domain Names for Applications (IDNA): An encoding process that transforms a string of Unicode characters into a smaller, restricted character set. **IDNA** encoding is commonly used for creating domain names that can be represented in the ASCII character set that is supported in the **Domain Name System (DNS)** of the Internet. **IDNA** uses the Punycode algorithm [RFC3492] and ACE (ASCII-compatible encoding) prefix [RFC5890] for the transformation.

Internet Protocol version 4 (IPv4): An Internet protocol that has 32-bit source and destination addresses. IPv4 is the predecessor of IPv6.

Internet Protocol version 6 (IPv6): A revised version of the Internet Protocol (IP) designed to address growth on the Internet. Improvements include a 128-bit IP address size, expanded routing capabilities, and support for authentication (2) and privacy.

Lightweight Directory Access Protocol (LDAP): The primary access protocol for **Active Directory**. Lightweight Directory Access Protocol (LDAP) is an industry-standard protocol, established by the Internet Engineering Task Force (IETF), which allows users to query and update information in a directory service (DS), as described in [MS-ADTS]. The Lightweight Directory Access Protocol can be either version 2 [RFC1777] or version 3 [RFC3377].

Network Access Protection (NAP): A feature of an operating system that provides a platform for system health-validated access to private networks. **NAP** provides a way of detecting the health state of a network client that is attempting to connect to or communicate on a network, and limiting the access of the network client until the health policy requirements have been met. **NAP** is implemented through quarantines and health checks, as specified in [TNC-IF-TNCCSPBSoH].

network byte order: The order in which the bytes of a multiple-byte number are transmitted on a network, most significant byte first (in big-endian storage). This may or may not match the order in which numbers are normally stored in memory for a particular processor.

OEM code page: See **original equipment manufacturer (OEM) code page**.

original equipment manufacturer (OEM) code page: A code page used to translate between non-Unicode encoded strings and UTF-16 encoded strings.

Rogue Authorized Server: A **DHCP server** that has been authorized using **Rogue Detection**.

Rogue Aware Server: A **DHCP server** that implements **Rogue Detection**.

Rogue Detection: A mechanism that can be used by a **DHCP server** to validate whether or not it is authorized to lease out addresses to **DHCP** clients.

Copyright © 2015 Microsoft Corporation

server: A computer on which the remote procedure call (RPC) server is executing.

Transmission Control Protocol (TCP): A protocol used with the Internet Protocol (IP) to send data in the form of message units between computers over the Internet. TCP handles keeping track of the individual units of data (called packets) that a message is divided into for efficient routing through the Internet.

Unauthorized Server: A **DHCP server** that is not authorized either administratively or using **Rogue Detection**. Unauthorized servers do not respond to either DHCPv4 or DHCPv6 messages.

User Datagram Protocol (UDP): The connectionless protocol within TCP/IP that corresponds to the transport layer in the ISO/OSI reference model.

UTF-8: A byte-oriented standard for encoding Unicode characters, defined in the Unicode standard. Unless specified otherwise, this term refers to the UTF-8 encoding form specified in [UNICODE5.0.0/2007] section 3.9.

Validating Server: A **Rogue Aware Server** that is attempting to validate its authorization using **Rogue Detection**.

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as defined in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the Errata.

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information.

[IANA-ENT] Internet Assigned Numbers Authority, "Private Enterprise Numbers", January 2007, http://www.iana.org/assignments/enterprise-numbers

[MS-ADA1] Microsoft Corporation, "Active Directory Schema Attributes A-L".

[MS-ADSC] Microsoft Corporation, "Active Directory Schema Classes".

[MS-ADTS] Microsoft Corporation, "Active Directory Technical Specification".

[MS-DHCPM] Microsoft Corporation, "Microsoft Dynamic Host Configuration Protocol (DHCP) Server Management Protocol".

[RFC1534] Droms, R., "Interoperation Between DHCP and BOOTP", RFC 1534, October 1993, http://www.ietf.org/rfc/rfc1534.txt

[RFC1812] Baker, F., Ed., "Requirements for IP Version 4 Routers", RFC 1812, June 1995, http://www.ietf.org/rfc/rfc1812.txt

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.rfc-editor.org/rfc/rfc2119.txt

Copyright © 2015 Microsoft Corporation

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997, http://www.ietf.org/rfc/rfc2131.txt

[RFC2132] Alexander, S., and Droms, R., "DHCP Options and BOOTP Vendor Extensions", RFC 2132, March 1997, http://www.ietf.org/rfc/rfc2132.txt

[RFC2730] Hanna, S., Patel, B., and Shah, M., "Multicast Address Dynamic Client Allocation Protocol (MADCAP)", RFC 2730, December 1999, http://www.ietf.org/rfc/rfc2730.txt

[RFC3004] Stump, G., Droms, R., Gu, Y., Vyaghrapuri, R., Demirtjis, A., Beser, B., and Privat, J., "The User Class Option for DHCP", RFC 3004, June 2000, http://www.ietf.org/rfc/rfc3004.txt

[RFC3315] Droms, R., Bound, J., Volz, B., et al., "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003, http://www.ietf.org/rfc/rfc3315.txt

[RFC3442] Lemon, T., Cheshire, S., and Volz, B., "The Classless Static Route Option for Dynamic Host Configuration Protocol (DHCP) Version 4", RFC 3442, December 2002, http://www.ietf.org/rfc/rfc3442.txt

[RFC3925] Littlefield, J., "Vendor-Identifying Vendor Options for Dynamic Host Configuration Protocol Version 4 (DHCPv4)", RFC 3925, October 2004, http://www.ietf.org/rfc/rfc3925.txt

[RFC4702] Stapp, M., Volz, B., and Rekhter, Y., "The Dynamic Host Configuration Protocol (DHCP) Client Fully Qualified Domain Name (FQDN) Option", RFC 4702, October 2006, http://www.ietf.org/rfc/rfc4702.txt

[RFC951] Croft, B., and Gilmore, J., "BOOTSTRAP Protocol (BOOTP)", RFC 951, September 1985, http://www.ietf.org/rfc/rfc951.txt

1.2.2 Informative References

[MS-DHCPN] Microsoft Corporation, "Dynamic Host Configuration Protocol (DHCP) Extensions for Network Access Protection (NAP)".

[MS-HNDS] Microsoft Corporation, "Host Name Data Structure Extension".

[MSDN-NAP] Microsoft Corporation, "Network Access Protection", http://msdn.microsoft.com/en-us/library/aa369712(VS.85).aspx

[RFC1001] Network Working Group, "Protocol Standard for a NetBIOS Service on a TCP/UDP Transport: Concepts and Methods", RFC 1001, March 1987, http://www.ietf.org/rfc/rfc1001.txt

[RFC1002] Network Working Group, "Protocol Standard for a NetBIOS Service on a TCP/UDP Transport: Detailed Specifications", STD 19, RFC 1002, March 1987, http://www.rfc-editor.org/rfc/rfc1002.txt

[RFC1035] Mockapetris, P., "Domain Names - Implementation and Specification", STD 13, RFC 1035, November 1987, http://www.ietf.org/rfc/rfc1035.txt

[RFC3396] Lemon, T., and Cheshire, S., "Encoding Long Options in the Dynamic Host Configuration Protocol (DHCPv4)", RFC 3396, November 2002, http://www.ietf.org/rfc/rfc3396.txt

[TNC-IF-TNCCSPBSoH] TCG, "TNC IF-TNCCS: Protocol Bindings for SoH", version 1.0, May 2007, http://www.trustedcomputinggroup.org/resources/tnc_iftnccs_protocol_bindings_for_soh_version_10/

Copyright © 2015 Microsoft Corporation

1.3 Overview

DHCP uses the following basic steps to automatically configure a network address and configuration information on a **DHCP client**. The application of DHCP discussed here is an illustrative example of an **IPv4** network, where the DHCP client and the DHCP server are on the same subnet and the client machine has no prior IP address configured on the network interface. DHCP may also be used by a client to obtain configuration parameters (other than the IP address) from the DHCP server. For further details, see section 3.4 of [RFC2131].

1. When the **TCP**/IP protocol initializes and **DHCPv4** has been enabled on any of the client machine's interfaces, the DHCPv4 client sends a **DHCPDISCOVER** message to find the DHCPv4 servers on the network and to obtain a valid IPv4 address configuration. The DHCP client includes a Vendor Class Identifier Option that contains "vendor-class identifier" information about the host, such as the operating system version.

2. All DHCPv4 servers that receive the **DHCPDISCOVER** message and have been configured with valid IPv4 address configuration for the client send a **DHCPOFFER** message back to the DHCPv4 client. The DHCPv4 servers optionally include other configuration information for the client in the **DHCPOFFER** message, in case the client wants to select the specific configuration information that it desires. If configuration information is included, then based on the vendor class identifier that the client included in the message, the DHCPv4 servers also include any specific standard options or vendor-specific options appropriate to hosts running that operating system version. If no specific standard options or vendor-specific options are defined for hosts running that operating system version, the server ignores the Vendor Class Identifier Option sent by the client.

3. The DHCPv4 client selects an IPv4 address configuration to use from the **DHCPOFFER** messages that it receives. The DHCPv4 client then sends a **DHCPREQUEST** message to the selected DHCPv4 server by using the **Server ID** option, requesting the use of the selected configuration. The client again includes its vendor class identifier in the message.

4. The **DHCPREQUEST** message identifies the server that sent the offer that the DHCPv4 client selected. The DHCPv4 servers for which the Server Identifier sent by the client in the **DHCPREQUEST** does not match the **Server ID** put the offered IPv4 address back into the available pool of addresses. The selected DHCPv4 server assigns the IPv4 address configuration to the DHCPv4 client and sends a **DHCPACK** (acknowledgment) message to the DHCPv4 client. The DHCPv4 server includes configuration information, including any specific standard options or vendor-specific options based on the vendor class identifier sent by the client in the **DHCPREQUEST** message.

The DHCPv4 client computer completes the TCP/IP initialization. It then repeats the preceding steps for other interfaces, if present, for which DHCP is enabled. (See the following figure.) After this is complete, the client can use all TCP/IP services and applications for normal network communications and connectivity to other IPv4 hosts.

Copyright © 2015 Microsoft Corporation

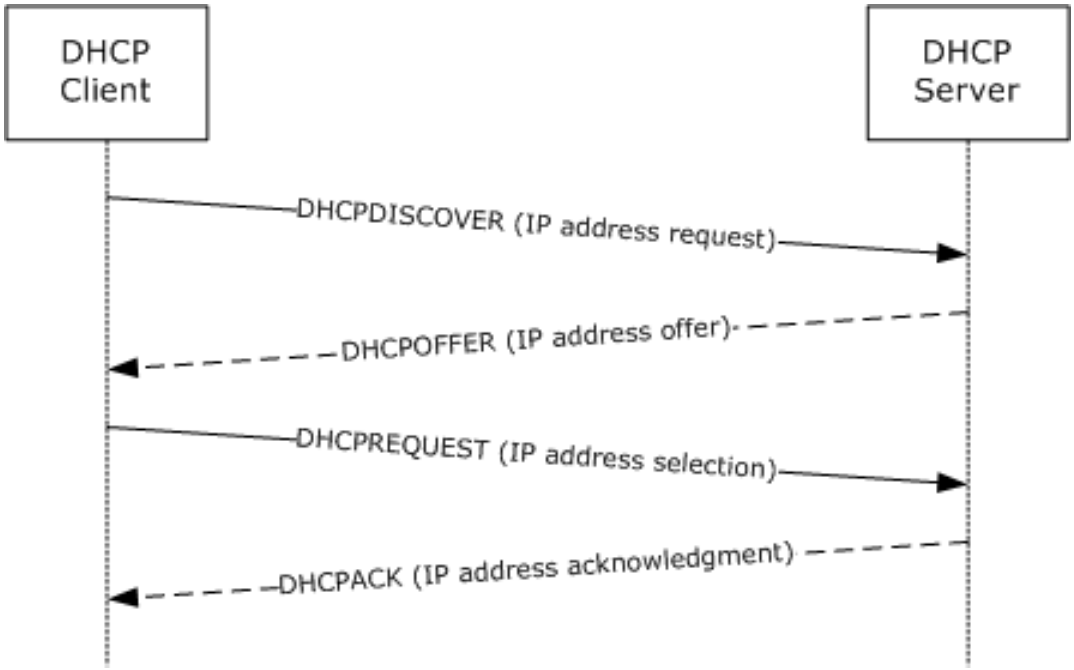


Figure 1: Basic DHCP process

The DHCPv4 client may decline an offer from a DHCPv4 server if it determines that the IP address included in the **DHCPOFFER** message sent by the server is already in use on that network. If so, the DHCPv4 client sends a **DHCPDECLINE** message and restarts the configuration process by sending a **DHCPDISCOVER** message again.

The DHCPv4 server may send a **DHCPNAK** message in response to the client's **DHCPREQUEST** message if one or more of the desired configuration options sent by the client in that message are unacceptable. In this case, the DHCPv4 client restarts the configuration process by sending a **DHCPDISCOVER** message again.

The DHCPv4 client may relinquish its lease on the IP address by sending a **DHCPRELEASE** message to the server.

In some cases, the DHCPv4 client may remember and want to reuse an IP address that was previously allocated by the DHCPv4 server to it. In this case, the client may begin the initialization process by sending a **DHCPREQUEST** message to the server containing that network address as the "requested IP address". The DHCPv4 server sends a **DHCPACK** message to the client if it chooses to allow the client to continue to use that IP address. Otherwise, the DHCPv4 server may send a **DHCPNAK** message to the client.

For further details on the DHCP protocol overview, refer to section 3 of [RFC2131].

DHCPv6 uses the following basic steps to automatically configure a network address on a DHCPv6 client. The application of DHCPv6 discussed here is an illustrative example of an **IPv6** network. The DHCPv6 client and the DHCPv6 server are on the same subnet, and the client machine has no prior IPv6 address configured on the network interface. DHCPv6 may also be used by a client to obtain configuration parameters (other than the IP address) from the DHCPv6 server. Details are as specified in [RFC3315] sections 1, 18.1.5, and 18.2.5.

1. When the TCP/IP protocol initializes and DHCPv6 has been enabled on any of the client machine's interfaces, the DHCPv6 client sends a DHCPv6 **Solicit** message to the **All_DHCP_Relay_Agents_and_Servers** multicast address specified in [RFC3315] to discover

Copyright © 2015 Microsoft Corporation

the available DHCPv6 servers. The DHCPv6 client includes a Vendor Class Option that contains information about the host, such as the operating system version.

2. All DHCPv6 servers that receive the DHCPv6 **Solicit** message from the client and have been configured with valid IPv6 address configuration information for the client send a DHCPv6 **Advertise** message in response to the DHCPv6 client. The DHCPv6 servers optionally include other configuration information for the client in the DHCPv6 **Advertise** message, in case the client wants to select the specific configuration information it requires. If configuration information is included, then based on the vendor class information that the client included in the message, the DHCPv6 servers also include any specific standard options or vendor-specific options appropriate to hosts running that operating system version. If no specific standard options or vendor-specific options are defined for hosts running that operating system version, the DHCPv6 servers ignore the Vendor Class Option sent by the client.

3. The DHCPv6 client selects an IPv6 address configuration to use from the DHCPv6 **Advertise** messages that it receives. The DHCPv6 client then sends a DHCP **Request** message to the selected DHCPv6 server by using the Server Identifier option, requesting the use of the selected configuration. The client again includes a Vendor Class Option in the message.

4. The DHCPv6 **Request** message identifies the server that sent the offer that the DHCPv6 client selected. The DHCPv6 servers for which the **Server Identifier** sent by the client in DHCPv6 **Request** does not match the **Server Identifier** put the offered IPv6 address back into the available pool of addresses. The selected DHCPv6 server assigns the IPv6 address configuration to the DHCPv6 client and sends a DHCPv6 **Reply** message with no Status Code option to the DHCPv6 client. The DHCPv6 servers include the configuration information, including any specific standard options or vendor-specific options based on the vendor class information sent by the client in the DHCPv6 **Request** message.

The presence of a Status Code option with any value other than Success in a DHCPv6 message from the server to the client is construed as a failure, and the DHCPv6 client then restarts the initialization process by sending the DHCPv6 **Solicit** message again.

The DHCPv6 client computer completes the TCP/IP initialization as described in the preceding steps. It then repeats the preceding steps for other interfaces, if present, for which DHCPv6 is enabled. (See the following figure.) After this is complete, the client can use all TCP/IP services and applications for normal network communications and connectivity to other IPv6 hosts.

Copyright © 2015 Microsoft Corporation

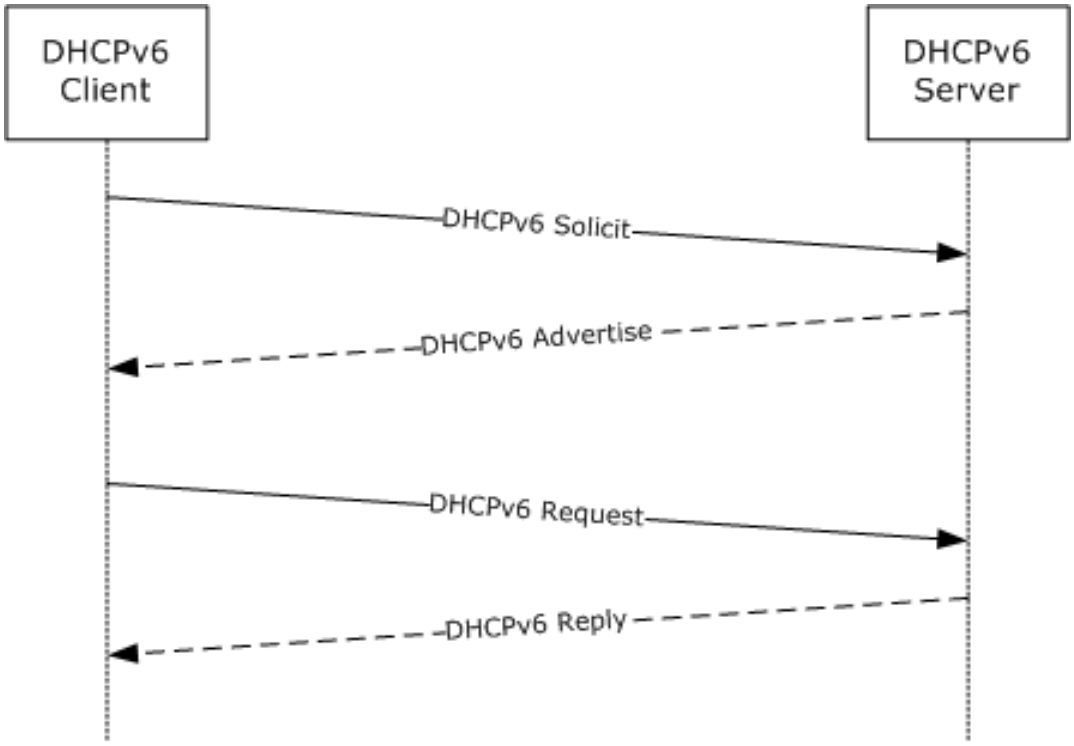


Figure 2: Basic DHCPv6 process

The DHCPv6 client may decline an offer from a DHCPv6 server if it finds that the IPv6 address included in the DHCPv6 **Advertise** message sent by the server is already in use on that network. If so, the DHCPv6 client sends a DHCPv6 **Decline** message and restarts the configuration process by sending a DHCPv6 **Solicit** message again.

The DHCPv6 server may send a DHCPv6 **Reply** message with a Status Code option with a value other than Success in response to the client's DHCPv6 **Request** message if one or more of the desired configuration options sent by the client in that message are unacceptable. In this case, the DHCPv6 client restarts the configuration process by sending a DHCPv6 **Solicit** message again.

The DHCPv6 client may relinquish its lease on the IP address by sending a DHCPv6 **Release** message to the server.

In some cases, the DHCPv6 client may remember and want to reuse an IP address that was previously allocated by the DHCPv6 server to it. In this case, the client may begin the initialization process by sending a DHCPv6 **Renew** or **Rebind** message to the server containing that network address as the "requested IP address". The DHCPv6 server sends a DHCPv6 **Reply** message with no Status Code option to the client if it chooses to allow the client to continue to use that IPv6 address. Otherwise, the DHCPv6 server may send a DHCP **Reply** message to the client with a value other than Success.

For further details of the DHCPv6 protocol overview, see section 3 of [RFC3315].

Accidental configuration of multiple DHCP servers on a network may cause misconfiguration of **DHCP clients**. DHCP servers can implement **Rogue Detection** to prevent such accidental configurations. A **Rogue Aware Server** periodically checks whether it is authorized.

Rogue Detection can be implemented using **DHCPINFORM** and/or DHCPv6 Information-Request messages. The following scenarios are valid for Rogue Detection:

Authorization of a DHCPv4 server using DHCPINFORM.

Copyright © 2015 Microsoft Corporation

1. A **Validating Server** sends a broadcast **DHCPINFORM** message containing a designated vendor-specific option requesting other Rogue Aware Servers on the network to respond.

2. A Rogue Aware **Administratively Authorized Server** replies to the **DHCPINFORM** message by sending a **DHCPACK** message with the corresponding vendor-specific option with a NULL-terminated string in the option data.

3. A **Rogue Authorized Server** on the network replies to the **DHCPINFORM** message by sending a **DHCPACK** message containing the corresponding vendor-specific option with a NULL-terminated string in the option data.

4. Any DHCP servers that are not Rogue Aware may reply to the **DHCPINFORM** message by sending a **DHCPACK** message, which does not contain the corresponding vendor-specific option.

5. A Rogue Aware **Unauthorized Server** on the network will not reply to the **DHCPINFORM** message.

6. If a Validating Server receives a **DHCPACK** message from an Administratively Authorized Server, it will consider itself unauthorized.

7. If a Validating Server does not receive a **DHCPACK** message from an Administratively Authorized Server within a stipulated time, it retries sending the **DHCPINFORM** message.

8. If a Validating Server receives a **DHCPACK** message from a Rogue Authorized Server, or a server which is not Rogue Aware, it retries sending the **DHCPINFORM** message.

9. The maximum number of retries is implementation specific. After all retry attempts are exhausted, the Validating Server may consider itself authorized or continue validation using the DHCPv6 Information-Request message.

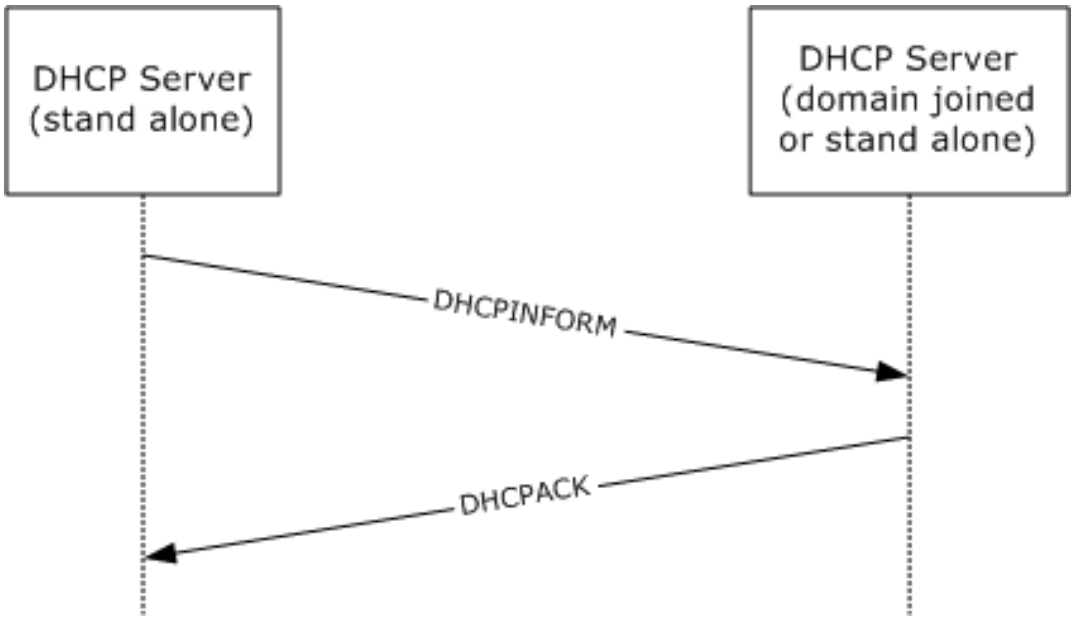


Figure 3: DHCPv4 Server Authorization messages

Authorization of a DHCP server using Information-Request.

1. A Validating Server sends a DHCPv6 Information-Request message containing a designated vendor specific option requesting other Rogue Aware Servers on the network to respond.

Copyright © 2015 Microsoft Corporation

2. A Rogue Aware Administratively Authorized Server replies to the DHCPv6 Information-Request message by sending a DHCPv6 **Reply** message containing the corresponding vendor-specific option with a NULL-terminated string in the option data.

3. A Rogue Authorized Server on the network replies to the DHCPv6 Information-Request message by sending a DHCPv6 **Reply** message containing the corresponding vendor-specific option with a NULL-terminated string in the option data.

4. Any DHCP server that is not Rogue Aware may reply to the DHCPv6 Information-Request message by sending a DHCPv6 **Reply** message that does not contain the corresponding vendor-specific option.

5. A Rogue Aware Unauthorized Server on the network will not reply to the DHCPv6 Information-Request message.

6. If a Validating Server receives a DHCPv6 **Reply** message from an Administratively Authorized Server, it will consider itself unauthorized.

7. If a Validating Server does not receive a DHCPv6 **Reply** message from an Administratively Authorized Server within a stipulated time, it retries sending the DHCPv6 Information-Request message.

8. If a Validating Server receives a DHCPv6 **Reply** message from a Rogue Authorized Server, or a server which is not Rogue Aware, it retries sending the DHCPv6 Information-Request message.

9. The maximum number of retries is implementation-specific. After all retry attempts are exhausted, the Validating Server will consider itself authorized.

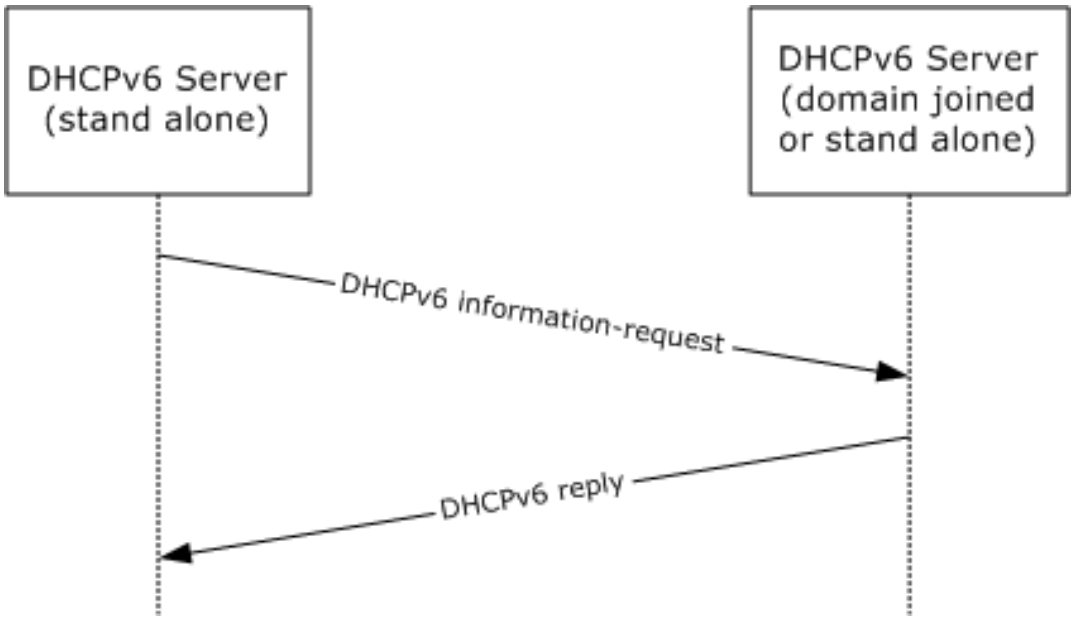


Figure 4: DHCPv6 Server Authorization messages

1.4 Relationship to Other Protocols

DHCPv4 (as specified in [RFC2131]) is based on the Bootstrap Protocol (BOOTP), as specified in [RFC951]. The format of the DHCPv4 messages is based on the format of the BOOTP messages. The relationship between these two protocols is defined in [RFC1534].

The vendor-specific options specified in this document rely on and are transported within DHCPv4.

Copyright © 2015 Microsoft Corporation

DHCPv4 can be used as one of the enforcement mechanisms defined for Network Access Protection (NAP), as described in [MSDN-NAP]. The vendor-specific options used for DHCPv4-based enforcement of NAP are defined in [MS-DHCPN] section 1. [MS-DHCPN] affects the contents of DHCPv4 messages when NAP is used.

The NetBIOS over TCP/IP protocol is defined in [RFC1001] and [RFC1002]. The vendor-specific option defined in this specification provides the capability to enable or disable the use of the NetBIOS over TCP/IP protocol on client and server machines.

[MS-DHCPM] affects the contents of DHCPv4 messages extended by this protocol by setting or modifying DHCP server configurations.

The following diagram illustrates the layering of the protocol in this section with other protocols in its stack.

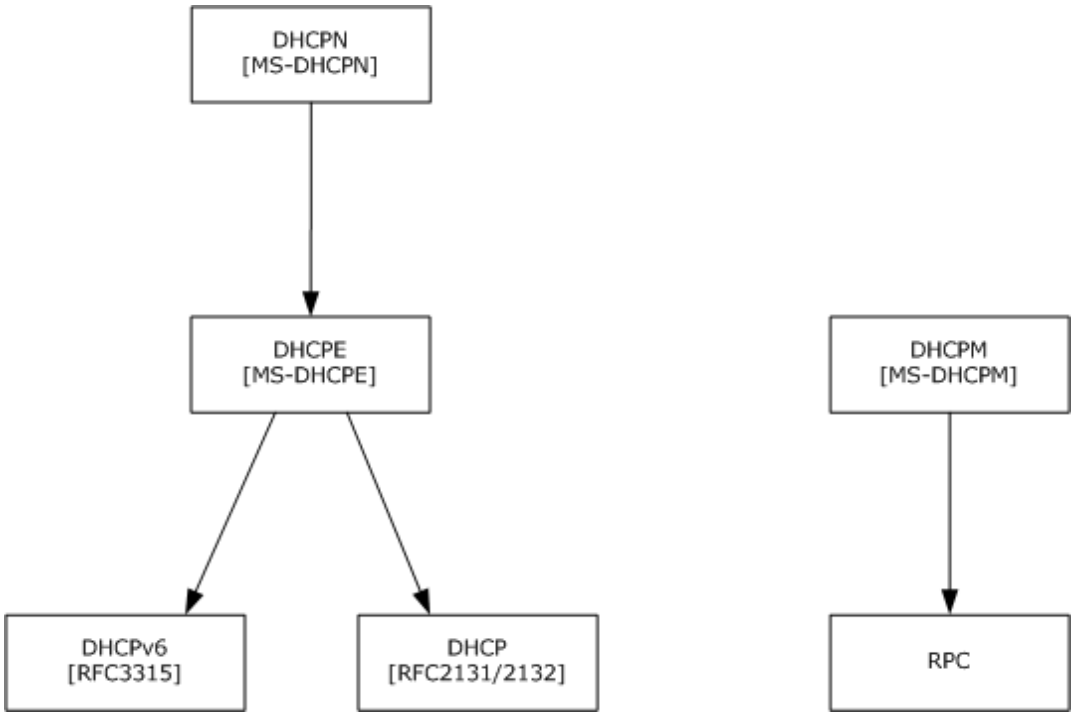


Figure 5: Protocol layering diagram

The following data flow diagram illustrates the interaction of the server implementation of this protocol with those of other protocols in its stack.

Copyright © 2015 Microsoft Corporation

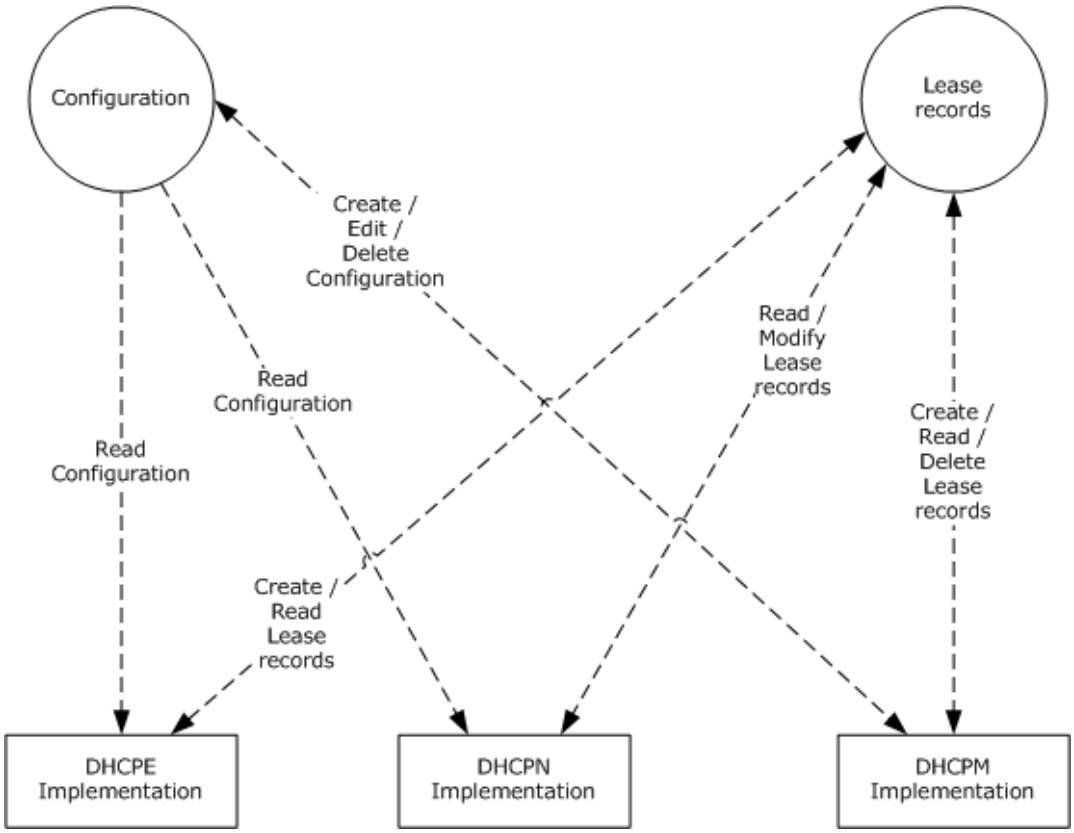


Figure 6: Server-side interaction with related protocols

The following is the relationship between [MS-DHCPM] ADM elements and the elements defined by [RFC2131] and [RFC3315] which are extended by this extension.

1. The subnet ([RFC2131] section 2) is represented by the **DHCPv4Scope** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.2). The DHCPv4 server will process an incoming DHCPv4 client message only if there exists a **DHCPv4Scope** object in its configuration that matches either the IP address of the network interface on which it received the message or the IP address of the relay agent in the client message (as specified in [RFC2131] section 4.3.1).

2. The **DHCPv4IpRange**, a shared ADM element (see [MS-DHCPM] section 3.1.1.4), restricts the range of available network addresses ([RFC2131] section 3.1 point 2) for allocation within a **DHCPv4Scope**. Once a subnet is selected, the DHCPv4 server identifies a **DHCPv4IpRange** object (we allow only up to 1 object to be configured) in the **DHCPv4Scope** which has available addresses in it. If no range is configured or the range is full, the DHCPv4 server will not respond to the client message. Otherwise the IP address to be assigned will be decided based on the available address in the range.

3. The **DHCPv4ExclusionRange**, a shared ADM element (see [MS-DHCPM] section 3.1.1.5), marks a range of address within a subnet as excluded from allocation. The IP addresses within the **DHCPv4ExclusionRange** will not be counted as available network addresses ([RFC2131] section 3.1 point 2). The DHCPv4 server will also check for the existence of **DHCPv4ExclusionRange** objects (these can be multiple). IP addresses will not be assigned from these ranges.

4. Manual allocation ([RFC2131] section 1) is achieved by the **DHCPv4Reservation** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.6). The DHCPv4 server also checks for the existence of a **DHCPv4Reservation** object that corresponds to the hardware address in the client

Copyright © 2015 Microsoft Corporation

message. If a matching reservation exists, the corresponding IP address will be assigned to the client even if it lies outside of the **DHCPv4IpRange** or within a **DHCPv4ExclusionRange**.

5. The database of allocated addresses and leases ([RFC2131] section 4) is represented by the **DHCPv4Client** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.7). Whenever a client accepts the IP address assigned to it by the DHCPv4 server, the latter will create a **DHCPv4Client** object and add it to the subnet's client list.

6. The **DHCPv4Filter** elements, shared ADM elements (see [MS-DHCPM] section 3.1.1.30), implement DHCPv4 server administrative controls ([RFC2131] section 4.2). The **DHCPv4FiltersList** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1), defines global allow/deny lists which determine which clients the server should allocate addresses to. The **DHCPv4FilterStatus** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1), can be used by the administrator to enable/disable enforcement of the allow/deny lists. The enforcement works in the following way:

    1. If neither DHCPv4FilterStatus.EnforceAllowList nor DHCPv4FilterStatus.EnforceDenyList is set to TRUE, the client message is processed further for the DHCPv4 protocol and no further checking for a DHCPv4 filter element is done.

    2. If the incoming client message has the client hardware address ([RFC2131] section 2) that matches a **DHCPv4Filter** entry in the **DHCPv4FiltersList** with ListType Deny and the DHCPv4FilterStatus.EnforceDenyList is set to TRUE, then the client message is not processed further or responded to.

    3. If the incoming client message has the client hardware address ([RFC2131] section 2) that matches a **DHCPv4Filter** entry in the **DHCPv4FiltersList** with ListType Allow and the **DHCPv4FilterStatus.EnforceAllowList** is set to TRUE, then the client message is processed further for the DHCPv4 protocol and no further checking for a DHCPv4 filter element is done.

    4. If the **DHCPv4FilterStatus.EnforceAllowList** is set to true and the client hardware address ([RFC2131] section 2) does not match any **DHCPv4Filter** entry in the **DHCPv4FiltersList** with ListType Allow, then the client message is not processed further or responded to.

7. The **DHCPv4SuperScope** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.3), allows configuration of network architectures with more than one IP subnets assigned to a physical network segment ([RFC2131] section 4.3.1). If the subnet that would be normally chosen by the DHCPv4 server according to the relay agent IP address has exhausted all addresses and happens to have a non-zero **DHCPv4Scope.SuperScopeId**, a shared ADM element (see [MS-DHCPM] section 3.1.1.23.1.1.2), then the server may choose to allocate an address from any other subnet configured with the same **DHCPv4Scope.SuperScopeId**.

8. The **DHCPv4ServerOptValueList**, a shared ADM element (see [MS-DHCPM] section 3.1.1.1), the **DHCPv4Scope.DHCPv4ScopeOptValuesList**, a shared ADM element (see [MS-DHCPM] section 3.1.1.2), and the **DHCPv4Reservation. DHCPv4ResvOptValuesList**, a shared ADM element (see [MS-DHCPM] section 3.1.1.6), allow explicit configuration of a default value for parameters requested by the client ([RFC2131] section 4.3.1). The order of selecting a configured default value is:

    1. **DHCPv4OptionValue** configured in the **DHCPv4Reservation**. **DHCPv4ResvOptValuesList** for a **DHCPv4Reservation** matching the client hardware address ([RFC2131] section 2) / client identifier ([RFC2132] section 9.14).

    2. **DHCPv4OptionValue** configured in the **DHCPv4Scope.DHCPv4ScopeOptValuesList** for a **DHCPv4Scope** selected as outlined above.

    3. **DHCPv4OptionValue** configured in the **DHCPv4ServerOptValueList**

9. Wherever the client message contains a user class option ([RFC3004]) and there exists a **DHCPv4ClassDef** object, a shared ADM element (see [MS-DHCPM] section 3.1.1.8), whose

Copyright © 2015 Microsoft Corporation

DHCPv4ClassDef.ClassData and **DHCPv4ClassDef.ClassDataLength** match the user class option data then any parameter values configured in **DHCPv4Reservation**. **DHCPv4ResvOptValuesList**, **DHCPv4Scope.DHCPv4ScopeOptValuesList** or **DHCPv4ServerOptValueList** with the corresponding **DHCPv4ClassDef.ClassName** in the **DHCPv4OptionValue.UserClass**, a shared ADM element (see [MS-DHCPM] section 3.1.1.11), will be selected in preference to parameters configured without a ClassName in any list. The overall order of selecting a configured default value is:

1. **DHCPv4OptionValue** with matching ClassName configured in the DHCPv4Reservation. DHCPv4ResvOptValuesList for a **DHCPv4Reservation** matching the client hardware address ([RFC2131] section 2) / client identifier ([RFC2132] section 9.14).

2. **DHCPv4OptionValue** with matching ClassName configured in the DHCPv4Scope.DHCPv4ScopeOptValuesList for a **DHCPv4Scope** selected as outlined above.

3. **DHCPv4OptionValue** with matching ClassName configured in the **DHCPv4ServerOptValueList**

4. **DHCPv4OptionValue** with no ClassName configured in the **DHCPv4Reservation**. **DHCPv4ResvOptValuesList** for a **DHCPv4Reservation** matching the client hardware address ([RFC2131] section 2) / client identifier ([RFC2132] section 9.14).

5. **DHCPv4OptionValue** with no ClassName configured in the DHCPv4Scope.DHCPv4ScopeOptValuesList for a **DHCPv4Scope** selected as outlined above.

6. **DHCPv4OptionValue** with no ClassName configured in the **DHCPv4ServerOptValueList**

10. The **DHCPv4ServerMibInfo** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1), is updated by the server with the counts of various DHCPv4 messages ([RFC2131] section 3.1) processed or sent by it. Specifically, **DHCPv4ServerMibInfo.Discovers**, **DHCPv4ServerMibInfo.Offers**, **DHCPv4ServerMibInfo.Requests**, **DHCPv4ServerMibInfo.Declines** and **DHCPv4ServerMibInfo.Releases** are updated with the counts of **DHCPDISCOVER**, **DHCPOFFER**, **DHCPREQUEST**, **DHCPDECLINE** and **DHCPRELEASE** messages processed by the server respectively. DHCPv4ServerMibInfo.Acks and DHCPv4ServerMibInfo.Naks are updated with the counts of **DHCPACK** and **DHCPNAK** messages sent by the server respectively.

11. IPv6 prefixes ([RFC3315] section 4.1) are configured on the server as **DHCPv4Scope** elements, shared ADM elements (see [MS-DHCPM] section 3.1.1.14). IP addresses are selected for assignment to an IA ([RFC3315] section 11) based on the existence in configuration of a prefix corresponding to the address of the interface over which a direct message was received or the address of the forwarding relay agent in the case of relay forwarded messages.

12. The **DHCPv6ExclusionRange** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.15), marks a range of address within a subnet as excluded from allocation. While selecting addresses for assignment to an IA ([RFC3315] section 11) the server will not select addresses so excluded from allocation.

13. The **DHCPv6Reservation** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.16), implements a manual allocation scheme on par with the one outlined for DHCPv4 processing above.

14. The **DHCPv6ClientInfo** element, a shared ADM element ([MS-DHCPM] section 3.1.1.18), represents a DHCPv6 binding which contains information about the identity association ([RFC3315] section 4.2). Whenever a client accepts the IP address assigned to it by the DHCPv6 server, the latter will create a **DHCPv6ClientInfo** object and add it to the DHCPv6Scope. DHCPv6ClientInfoList.

15. The **DHCPv6ServerClassedOptValueList**, a shared ADM element (see [MS-DHCPM] section 3.1.1.1), **DHCPv6Scope.DHCPv6ScopeClassedOptValueList**, a shared ADM element (see [MS-

Copyright © 2015 Microsoft Corporation

DHCPM] section 3.1.1.14), and **DHCPv6Reservation.DHCPv6ResvClassedOptValueList**, a shared ADM element (see [MS-DHCPM] section 3.1.1.16), allow the server to be configured to return options to the client as described in ([RFC3315] sections 17.2.2 and 18.2). The order of selecting a configured option is:

1. **DHCPv6OptionValue** configured in the DHCPv6Reservation.DHCPv6ResvClassedOptValueList for a **DHCPv6Reservation** matching the client identifier and IAID specified in the client message.

2. **DHCPv6OptionValue** configured in the DHCPv6Scope.DHCPv6ScopeClassedOptValueList for a **DHCPv6Scope** which corresponds to the prefix used in address selection as outlined above.

3. **DHCPv6OptionValue** configured in the **DHCPv6ServerClassedOptValueList**

16. Wherever the client message contains a user class option ([RFC3315] section 22.15) and there exists a **DHCPv6ClassDef** object, a shared ADM element (see [MS-DHCPM] section 3.1.1.19), whose DHCPv6ClassDef.ClassData and DHCPv6ClassDef.ClassDataLength match the user class option data then any parameter values configured in DHCPv6Reservation.DHCPv6ResvClassedOptValueList, DHCPv6Scope.DHCPv6ScopeClassedOptValueList or DHCPv6ServerClassedOptValueList with the corresponding DHCPv6ClassDef.ClassName in the DHCPv6OptionValue.UserClass, a shared ADM element (see [MS-DHCPM] section 3.1.1.21), will be selected in preference to a parameter configured without a ClassName in the corresponding list. The overall order of selecting a configured default value is:

    1. **DHCPv6OptionValue** with matching ClassName configured in the DHCPv6Reservation.DHCPv6ResvClassedOptValueList for a **DHCPv6Reservation** matching the client identifier and IAID specified in the client message.

    2. **DHCPv6OptionValue** with no ClassName configured in the DHCPv6Reservation.DHCPv6ResvClassedOptValueList for a **DHCPv6Reservation** matching the client identifier and IAID specified in the client message.

    3. **DHCPv6OptionValue** with matching ClassName configured in the DHCPv6Scope.DHCPv6ScopeClassedOptValueList for a **DHCPv6Scope** selected as outlined above.

    4. **DHCPv6OptionValue** with no ClassName configured in the DHCPv6Scope.DHCPv6ScopeClassedOptValueList for a **DHCPv6Scope** selected as outlined above.

    5. **DHCPv6OptionValue** with matching ClassName configured in the **DHCPv6ServerClassedOptValueList**

    6. **DHCPv6OptionValue** with no ClassName configured in the **DHCPv6ServerClassedOptValueList**

17. The **DHCPv6ServerMibInfo** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1), is updated by the server with the counts of various DHCPv6 messages ([RFC2131] section 3.1) processed or sent by it. Specifically, DHCPv6ServerMibInfo.Solicits, DHCPv6ServerMibInfo.Requests, DHCPv6ServerMibInfo.Renews, DHCPv6ServerMibInfo.Rebinds, DHCPv6ServerMibInfo.Confirms and DHCPv6ServerMibInfo.Declines, DHCPv6ServerMibInfo.Releases, DHCPv6ServerMibInfo.Informs are updated with the counts of DHCPv6 Solicit, Request, Renew, Rebind, Confirm, Decline, Release and Inform messages processed by the server respectively. DHCPv6ServerMibInfo.Advertises and DHCPv6ServerMibInfo.Replies are updated with the counts of DHCPv6 Advertise and Reply messages sent by the server respectively.

Copyright © 2015 Microsoft Corporation

18. The **DHCPv4ServerMcastMibInfo** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1), is updated by the server with the counts of various MADCAP messages ([RFC2730] section 2.2) processed or sent by it.

1.5 Prerequisites/Preconditions

None.

1.6 Applicability Statement

The use of these DHCP vendor-specific options is applicable in environments where DHCP or DHCPv6 is used.

The optional internationalization extensions specified in section 2.2.7 are only applicable in homogenous environments where either all machines use ASCII or have the same implementation-specific behavior.

1.7 Versioning and Capability Negotiation

The guidelines noted in section 8.4 of [RFC2132] to identify the vendor for the vendor-specific options are applicable to DHCPv4.

The Vendor Class Identifier Option defined in [RFC2132] section 9.13 and the Vendor Class Option defined in [RFC3315] section 22.16 contain values used to negotiate which vendor-specific options defined herein are to be sent to the DHCPv6 client.

1.8 Vendor-Extensible Fields

DHCPv4 (as specified in [RFC2131]) and DHCPv6 (as specified in [RFC3315]) have a provision for vendor-extensible options. These vendor-specific options are used as specified in [RFC2132] and [RFC3315]. The vendor-extensible fields described in this document comply with the provisions defined therein. The vendor-extensible options used by DHCPv4 clients and servers are specified in section 2.2.2.

1.9 Standards Assignments

Parameter	Value	Reference
Private Enterprise Number	311	[IANA-ENT]

Copyright © 2015 Microsoft Corporation

2 Messages

2.1 Transport

All DHCP attributes are transported within DHCP, which is transported over the UDP protocol, as specified in [RFC2131] section 4.1 for DHCPv4 and [RFC3315] section 5.2 for DHCPv6.

Parameter	Value	Reference
DHCPv4 server listens for DHCPv4 messages on UDP port.	0x0043	[RFC2131]
DHCPv4 client listens for DHCPv4 messages on UDP port.	0x0044	[RFC2131]
DHCPv6 server listens for DHCPv6 messages on UDP port.	0x0223	[RFC3315] section 5.2
DHCPv6 clients listen for DHCPv6 messages on UDP port.	0x0222	[RFC3315] section 5.2

2.2 Message Syntax

The following DHCP extensions use the message format for vendor-specific options, as specified in [RFC2132] section 8.4 and in [RFC3925] section 3.

All option fields and values described in this document are sent in **network byte order** unless indicated otherwise.

2.2.1 DHCPv4 Option Code 12 (0xC) - Host Name Option

This option, as defined in [RFC2132] section 3.14, specifies the name of the client. As per [RFC2132] section 3.14, the name MUST follow character set restrictions as specified in [RFC1035]. [RFC1035] section 2.3.1 specifies a "preferred" name syntax that uses letters, digits, and hyphens, but does not state whether this is mandatory (for more information, see [MS-HNDS]). In addition, [RFC1035] does not explain how to ASCII-encode a name if the client has a non-ASCII name, and hence the contents of this option are implementation-specific.<1> It was this ambiguity that led to it being rendered obsolete by DHCPv4 Option Code 81 (section 2.2.7).

2.2.2 DHCPv4 Option Code 43 (0x2B) - Vendor-Specific Information Option

DHCPv4 clients request vendor-specific options from the DHCPv4 server by including option code 43 in the Parameter Request List, as specified in [RFC2132] section 8.4.

DHCPv4 clients implementing this specification MUST also include a vendor-class identifier as described in section 2.2.3 in the DHCPv4. A Validating Server MAY include a vendor-class identifier when authorizing itself using Rogue Detection.

When a DHCPv4 message includes a Vendor Class Identifier with one of the values defined in section 2.2.3, the Vendor-Specific Information Option is defined to use the "Encapsulated vendor-specific options" format specified in [RFC2132] section 8.4. This specification defines the following encapsulated vendor-specific option codes.

Value	Meaning
0x01	Microsoft Disable NetBIOS Option (section 2.2.2.1)
0x02	Microsoft Release DHCP Lease on Shutdown Option (section 2.2.2.2)

Copyright © 2015 Microsoft Corporation

Value	Meaning
0x03	Microsoft Default Router Metric Base Option (section 2.2.2.3)
0x5E	Rogue Detection Request Option (section 2.2.2.4)
0x5F	Rogue Detection Option (section 2.2.2.5)

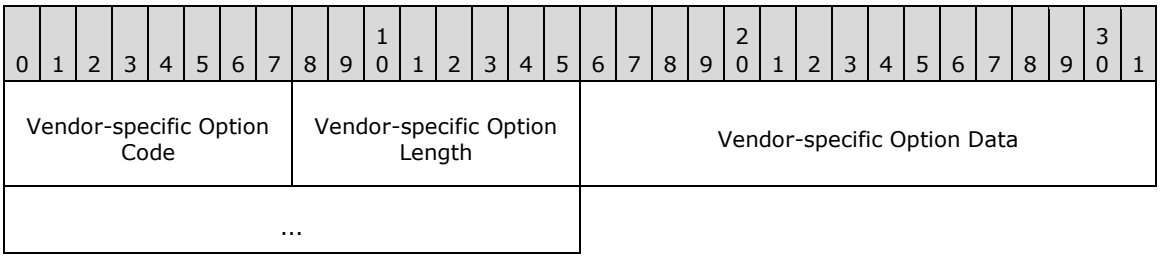
In addition, DHCPv4 clients that support **NAP** also use DHCPv4 vendor-specific options for exchanging NAP-specific information. For an overview of NAP and for more information, see [MS-DHCPN] and [TNC-IF-TNCCSPBSoH]. NAP utilizes vendor-specific options as defined in [MS-DHCPN] section 2.2.1.

For information about the format of DHCP Vendor Extensions, see [RFC2132] section 2 and [RFC3315] section 22.

2.2.2.1 Vendor-Specific Option Code 0x01 - Microsoft Disable NetBIOS Option

This option is sent by a DHCPv4 server to a DHCPv4 client in a **DHCPOFFER** or a **DHCPACK** message.<2> It has no effect on subsequent options in that message or on the **DHCPREQUEST** message sent by the client to the server.

This option can be used to enable or disable the use of NetBIOS over TCP/IP on the network interface for which the DHCPv4 message was received. DHCPv4 clients SHOULD<3> support this option. If the use of NetBIOS over TCP/IP is disabled on the interface, no NetBIOS over TCP/IP packets can be sent from or received on that interface. If any NetBIOS over TCP/IP packets are sent to the client on that interface, they are silently discarded. This option has no effect on NetBIOS over NetBEUI.



Vendor-specific Option Code (1 byte): This MUST be 0x01.

Vendor-specific Option Length (1 byte): This MUST be 0x04.

Vendor-specific Option Data (4 bytes): Values are as follows.

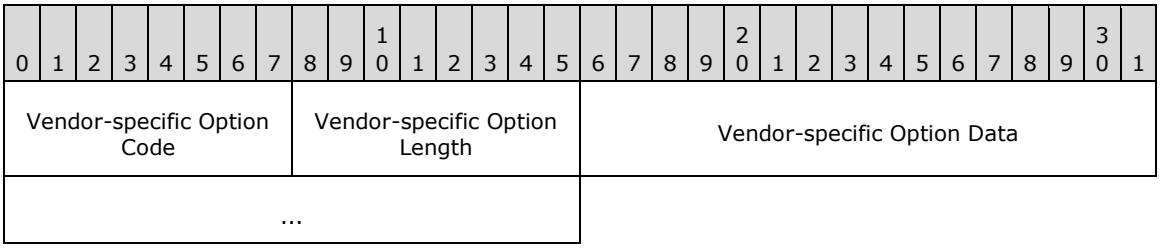
Value	Meaning
0x00000000	Enables NetBIOS over TCP/IP (Default Value) for that network interface.
0x00000001	Ignored (existing behavior unchanged).
0x00000002	Disables NetBIOS over TCP/IP for that network interface.
0x00000003 — 0xFFFFFFFF	Ignored (existing behavior unchanged).

Copyright © 2015 Microsoft Corporation

2.2.2.2 Vendor-Specific Option Code 0x02 - Microsoft Release DHCP Lease on Shutdown Option

This option is sent by a DHCPv4 server to a DHCPv4 client in a **DHCPOFFER** or a **DHCPACK** message. It has no effect on subsequent options in that message or on the **DHCPREQUEST** message sent by the client to the server.

This option is used in DHCPv4 messages by the DHCPv4 server for directing the clients to send a **DHCPRELEASE** on that network interface when the operating system on the client is shutting down. DHCPv4 clients SHOULD<4> support this option.



Vendor-specific Option Code (1 byte): This MUST be 0x02.

Vendor-specific Option Length (1 byte): This MUST be 0x04.

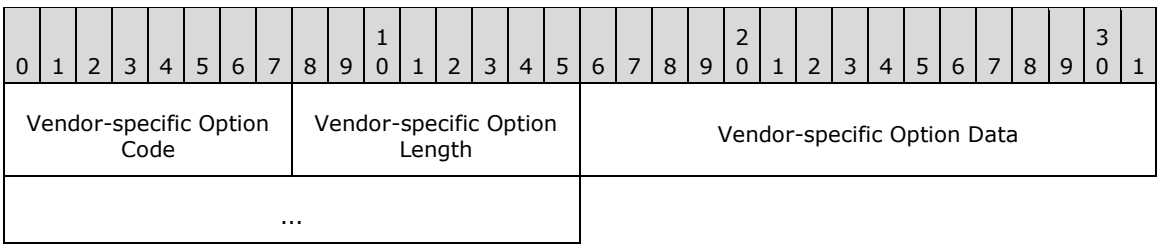
Vendor-specific Option Data (4 bytes): Values are as follows.

Value	Meaning
0x00000000	Disables client behavior of sending DHCPRELEASE message on operating system shutdown.
0x00000001	Enables client behavior of sending DHCPRELEASE message on operating system shutdown.
0x00000002 — 0xFFFFFFFF	Existing behavior of client is unchanged.

2.2.2.3 Vendor-Specific Option Code 0x03 - Microsoft Default Router Metric Base Option

This option is sent by the DHCPv4 server to the DHCPv4 client in a **DHCPOFFER** or a **DHCPACK** message. It has no effect on subsequent options in that message or on the **DHCPREQUEST** message sent by the client to the server.

This option is used to set the default route metric (as specified in [RFC1812] section 5.2.4.3) for all automatically computed network routes for the network interface on which the DHCPv4 message was received. DHCPv4 clients SHOULD<5> support this option.



Copyright © 2015 Microsoft Corporation

Vendor-specific Option Code (1 byte): This MUST be 0x03.

Vendor-specific Option Length (1 byte): This MUST be 0x04.

Vendor-specific Option Data (4 bytes): If zero, clients are to compute a route metric based on link speed. Otherwise, this value overrides the automatically calculated metric for the default route for that network interface.

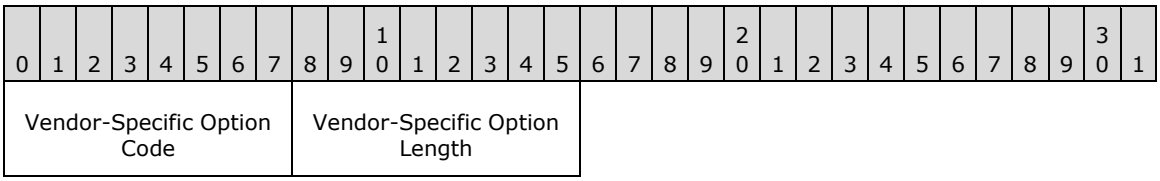
The automatically calculated metric for the default route for DHCPv4 client computers SHOULD<6> be one of the following values.

Value	Meaning
0x0000000A	Greater than 200 Mbps.
0x00000014	Greater than 80 Mbps, and less than or equal to 200 Mbps.
0x00000019	Greater than 20 Mbps, and less than or equal to 80 Mbps.
0x0000001E	Greater than 4 Mbps, and less than or equal to 20 Mbps.
0x00000028	Greater than 500 Kbps, and less than or equal to 4 Mbps.
0x00000032	Less than or equal to 500 Kbps.

2.2.2.4 Vendor-Specific Option Code 0x5E - Rogue Detection Request Option

This option is sent by a Validating Server to DHCPv4 servers on the network in a DHCPINFORM message. It is sent as an encapsulated vendor-specific option in option 43 (section 2.2.2).

The DHCPINFORM message does not contain the vendor-class identifier option (section 2.2.2).

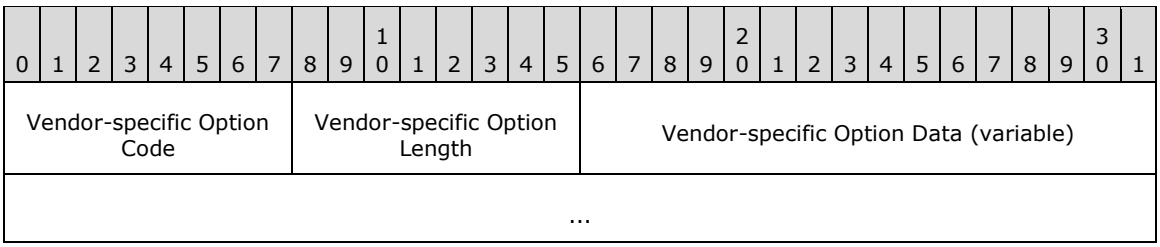


Vendor-Specific Option Code (1 byte): This MUST be 94 (0x5E).

Vendor-Specific Option Length (1 byte): This MUST be 0x00.

2.2.2.5 Vendor-Specific Option Code 0x5F – Rogue Detection Reply Option

This option is sent by a Rogue Aware Server to a Validating Server in a DHCPACK message. It is sent in response to an authorization message (see section 2.2.2.4) received in a DHCPINFORM message. It is sent as an encapsulated option in option 43 (section 2.2.2.4).



Copyright © 2015 Microsoft Corporation

Vendor-specific Option Code (1 byte): This MUST be 0x5F.

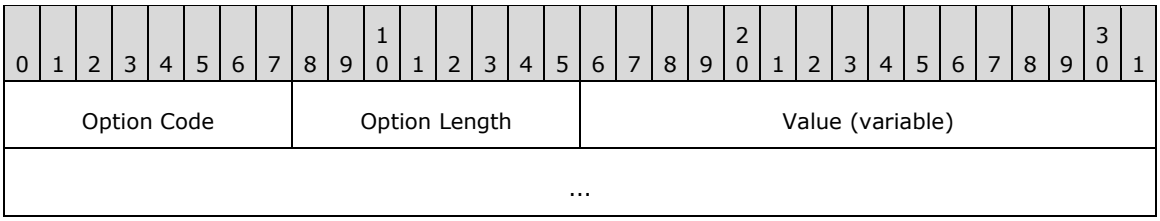
Vendor-specific Option Length (1 byte): The unsigned length, in bytes, of the **Vendor-specific Option Data** field. The maximum length is 255 bytes.

Vendor-specific Option Data (variable): This is a null-terminated string of length specified in **Vendor-specific Option Length**.

2.2.3 DHCPv4 Option Code 60 (0x3C) - Vendor Class Identifier Option

A DHCPv4 client sends vendor information in all DHCPv4 packets that it sends to the DHCPv4 server to indicate the vendor or the version of the operating system running on the client. This information is sent in the form of a Vendor Class Identifier Option, as specified in [RFC2132] section 9.13.

The DHCPv4 servers implementing this specification use the information contained in this option to determine whether a client implements this specification and whether the options defined in this specification should be sent to it. For semantics on the usage of vendor class identifiers, refer to [RFC2132] sections 8.4 and 9.13.



Option Code (1 byte): This MUST be 60 (0x3C) (as specified in [RFC2132] section 9.13) to indicate the Vendor Class Identifier Option.

Option Length (1 byte): The unsigned length in bytes of the **Value** field.

Value (variable): This MUST be set to one of the following values, where the value shown is encoded as a non-NULL-terminated ASCII string.

Value	Meaning
"MSFT 98"	The client implements this specification but does not understand any encapsulated vendor-specific options.
"MSFT 5.0"	The client implements this specification and understands all encapsulated vendor-specific options defined herein.

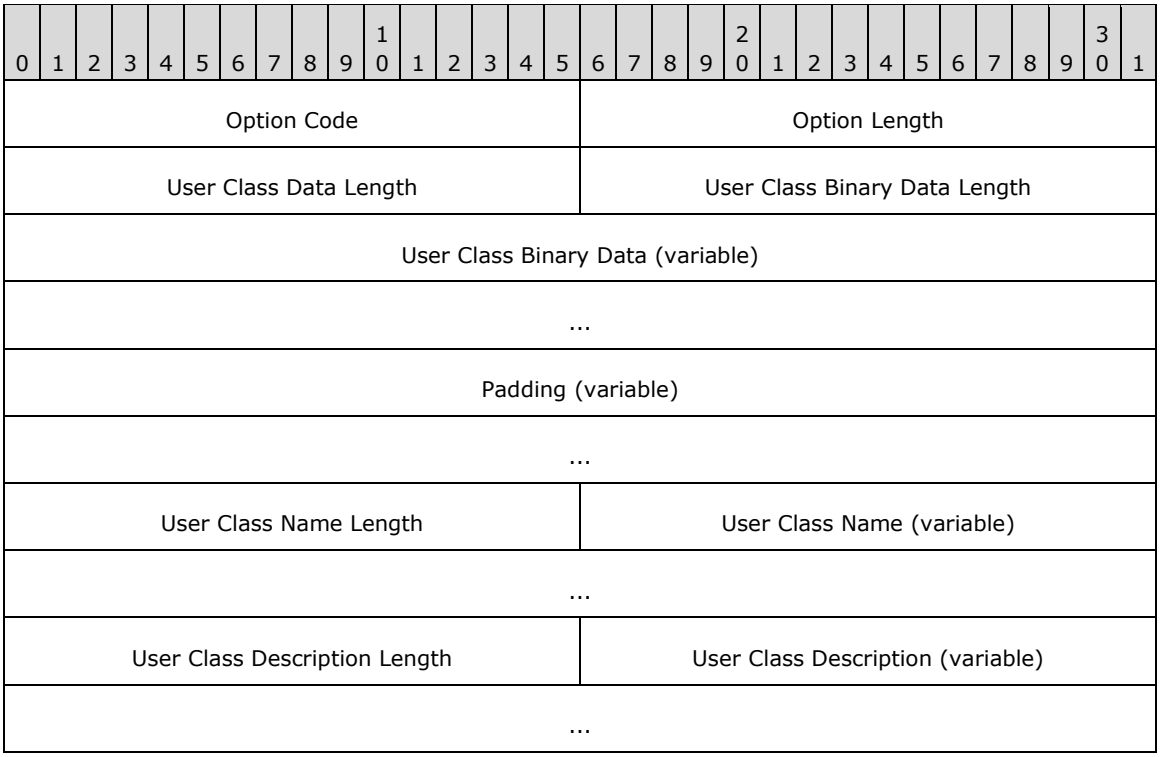
2.2.4 DHCPv6 Option Code 15 (0x000F) - User Class Option

DHCPv6 clients implementing this specification MUST use the message format and semantics specified in [RFC3315] when sending a User Class Option to a DHCPv6 server. The DHCPv6 client MUST send a maximum of only one User Class Option in a DHCPv6 message. This section describes the message format of User Class Option sent by DHCPv6 servers that implement this specification in response to an Option Request from the DHCPv6 client. The format of this option varies from the implementation described in [RFC3315].

DHCPv6 clients MAY<7> request the user classes configured on the DHCPv6 server by sending an Information-request message containing OPTION_ORO (Option 6) with OPTION_USER_CLASS (Option 15) as the ONLY requested option. On receiving the message, the DHCPv6 server SHOULD<8> send a **Reply** message to the DHCPv6 client containing one or more User Class Options (one for each user

Copyright © 2015 Microsoft Corporation

class configured on the DHCPv6 server) in the format shown as follows. This behavior is an extension of the DHCPv6 protocol defined in [RFC3315]. The DHCPv6 server MUST send the **OPTION_USER_CLASS** with a format described in [RFC3315] when the DHCPv6 server sends the option in response to an OPTION_USER_CLASS received from a DHCPv6 client as described in [RFC3315].



Option Code (2 bytes): This MUST be 15 (0x000F) to indicate User Class Option.

Option Length (2 bytes): The unsigned length, in bytes, of the User Class Option, not including the **Option Code** and **Option Length** fields.

User Class Data Length (2 bytes): Value is variable, depending on the size of **User Class Binary Data**, **User Class Name** and **User Class Description**.

User Class Binary Data Length (2 bytes): Size of **User Class Binary Data** in octets.

User Class Binary Data (variable): Binary data of a User Class set on the DHCPv6 server.

Padding (variable): Padding to align User Class Binary data to 4-byte boundary.

User Class Name Length (2 bytes): Size of **User Class Name** in octets.

User Class Name (variable): Name of a User Class set on the DHCPv6 server.

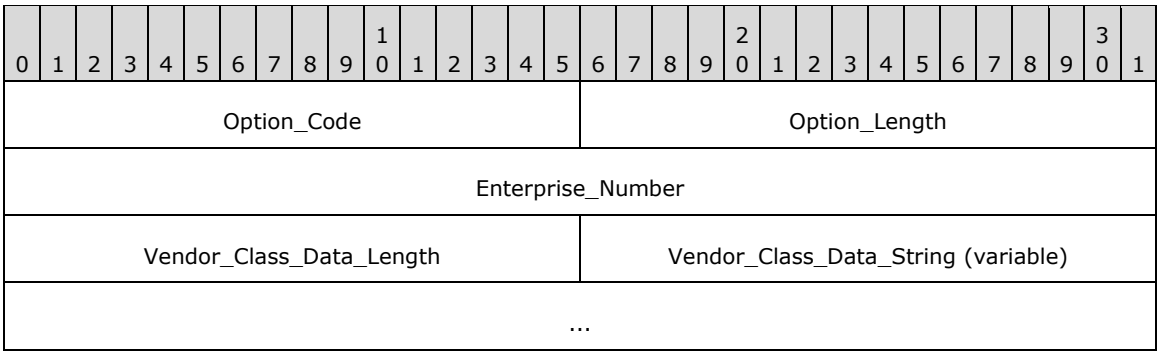
User Class Description Length (2 bytes): Size of **User Class Description** in octets.

User Class Description (variable): Description of a User Class defined on the DHCPv6 server.

2.2.5 DHCPv6 Option Code 16 (0x0010) - Vendor Class Option

A DHCPv6 client sends vendor information in all DHCPv6 packets to the DHCPv6 server. This information is sent in the form of a vendor class option, as specified in [RFC3315] section 22.16. An implementation that supports DHCPv6 MUST support this option.<9>

Copyright © 2015 Microsoft Corporation



Option_Code (2 bytes): As specified in [RFC3315] section 22, this is used to indicate the Vendor Class Option and MUST be 0x0010.

Option_Length (2 bytes): MUST be set to 0x000E (4 + 2 + the size of **Vendor_Class_Data_String**).

Enterprise_Number (4 bytes): MUST be set to 0x00000137 (decimal 311), the Internet Assigned Numbers Authority (IANA)–assigned Microsoft Enterprise number [IANA-ENT].

Vendor_Class_Data_Length (2 bytes): The length of the **Vendor Class Data String** field MUST be set to 0x0008.

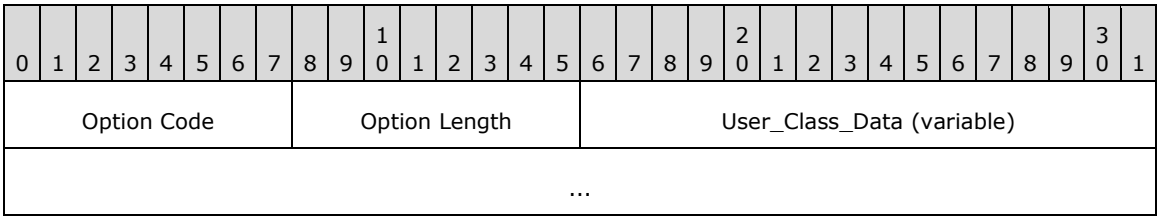
Vendor_Class_Data_String (variable): MUST be set to "MSFT 5.0".

2.2.6 DHCPv4 Option Code 77 (0x4D) - User Class Option

This section describes the message format of the User Class Option sent by DHCPv4 clients and DHCPv4 servers, and the values for this option that are predefined on DHCPv4 servers that implement this specification. The format of this option varies from the implementation described in [RFC3004] in that the User Class Data field format is changed. The use of this alternate format is indicated by the presence of a Vendor Class Identifier Option (section 2.2.3), which can occur anywhere in the same message.

2.2.6.1 User Class Option Sent by DHCPv4 Client to DHCPv4 Server

DHCPv4 clients MAY<10> send a User Class Option in all DHCPv4 messages sent by the client with any string configured by the administrator or a string from the User Class Data field. DHCPv4 clients support only one User Class option. For semantics of the usage of DHCPv4 user classes, refer to [RFC3004] section 4.



Option Code (1 byte): Must be 77 (0x4D) to indicate the User Class Option for DHCP.

Option Length (1 byte): Length in octets of the **User Class Data** field.

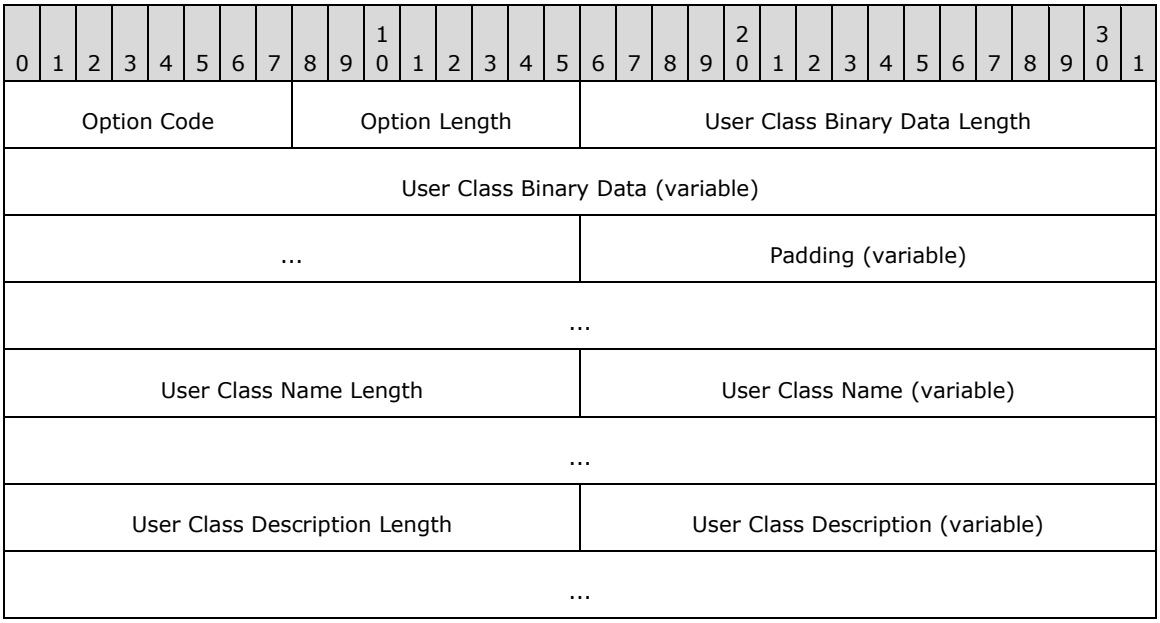
User_Class_Data (variable): The following User Class name is predefined in the DHCPv4 server and is sent by the DHCPv4 client in the specified scenario:

Copyright © 2015 Microsoft Corporation

Value	Meaning
"RRAS.Microsoft"	This value MUST be used if the DHCPv4 client is sending a User Class Option in a message on a dial-up or VPN network interface. This string is otherwise known as the Default Routing and Remote Access Class.

2.2.6.2 User Class Option Sent by DHCPv4 Server to DHCPv4 Client

DHCPv4 clients MAY<11> request the user classes configured on the DHCPv4 server by sending a **DHCPINFORM** message containing OPTION_PARAMETER_REQUEST_LIST (Option 55) with OPTION_USER_CLASS (Option 77) as one of the requested options. On receiving the message, the DHCPv4 server SHOULD<12> send a **DHCPACK** message to the DHCPv4 client containing one or more User Class Options (one for each user class configured on the DHCPv4 server) in the format shown as follows.



Option Code (1 byte): This MUST be 77 (0x4D) to indicate the User Class option for DHCP.

Option Length (1 byte): The unsigned length, in bytes, of the User Class Option, not including the **Option Code** and **Option Length** fields.

User Class Binary Data Length (2 bytes): Size of the User Class binary data in octets.

User Class Binary Data (variable): Binary data of a User Class set on the DHCPv4 server.

Padding (variable): Padding to align the User Class binary data to 4-byte boundary.

User Class Name Length (2 bytes): Size of the **User Class Name** in octets.

User Class Name (variable): Name of a User Class set on the DHCPv4 server.

User Class Description Length (2 bytes): Size of the **User Class Description** in octets.

User Class Description (variable): Description of a User Class set on the DHCPv4 server.

Copyright © 2015 Microsoft Corporation

2.2.7 DHCPv4 Option Code 81 (0x51) - Client FQDN Option

The client FQDN option is specified in [RFC4702] section 2. [RFC4702] section 2.3.1 states that setting the E bit to 0 indicates that the name is ASCII-encoded but does not explain how to ASCII encode a name if the client has a non-ASCII name. It then explains that client software may send data intended to be in other character sets but that support for other character sets is not required. This document clarifies that a client with a non-ASCII name MAY<13> set the E bit to 1 but that a client MAY<14> send its host name in an implementation-specific character set. [RFC4702] section 4 states that DHCPv4 servers SHOULD ignore the client FQDN option if the client's E bit is set to 0 and the servers do not support ASCII encoding. However, [RFC4702] section 2.3.1 states that client software may send data intended to be in other character sets, but that support for other character sets is not required. This specification clarifies that a DHCPv4 server MAY<15> accept other implementation-dependent character sets when the E bit is set to 0.

2.2.8 DHCPv4 Option Code 249 (0xF9) - Microsoft Classless Static Route Option

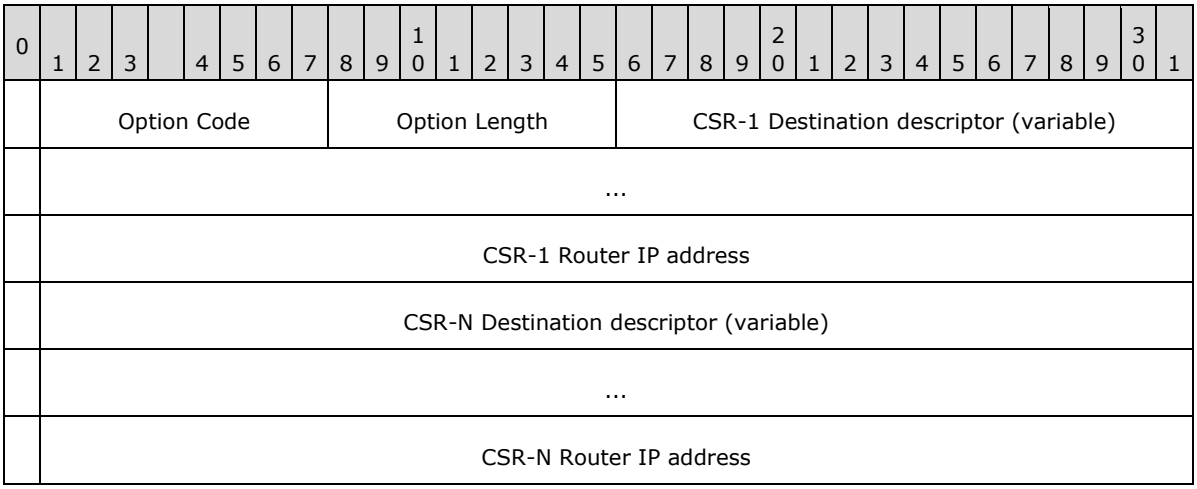
DHCPv4 clients and DHCPv4 servers that implement this specification use some nonstandard options in their implementation.

The length and the data format for the Microsoft **Classless Static Route** Option are exactly the same as those specified for the Classless Static Route Option in [RFC3442]; the only difference is that Option Code 249 SHOULD<16> be used instead of or in addition to Option Code 121.

Multiple routes can be sent using the option. Each classless route consists of the Destination descriptor and Router IP address elements. The number of routes included in the option can be determined by processing the option data.

Note that the router IP address is of length 4 bytes, whereas the destination descriptor length is between 1 byte and 5 bytes, depending on the subnet mask. This is described in detail as follows.

This option is sent by the DHCPv4 server to the DHCPv4 client in the DHCPOFFER or the DHCPACK message. It has no effect on subsequent options in that message or in any of the messages sent by the **client** to the **server**.



Option Code (1 byte): This MUST be 249 (0xF9).

Option Length (1 byte): The unsigned length, in bytes, of the option, not including the **Option Code** and **Option Length** fields.

CSR-1 Destination descriptor (variable): This is determined in exactly the same way described on the destination descriptor specified on page 4 of [RFC3442].

Copyright © 2015 Microsoft Corporation

CSR-1 Router IP address (4 bytes): The IPv4 address of the next-hop router that can be used to reach the destination.

CSR-N Destination descriptor (variable): This is determined in exactly the same way as the destination descriptor specified on page 4 of [RFC3442].

CSR-N Router IP address (4 bytes): The IPv4 address of the next-hop router that can be used to reach the destination.

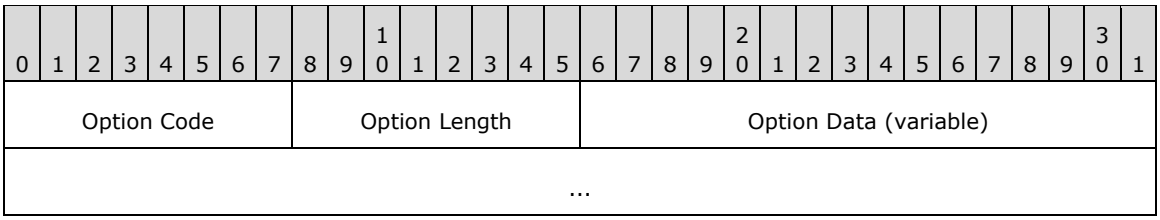
2.2.9 DHCPv4 Option Code 250 (0xFA) - Microsoft Encoding Long Options Packet

DHCPv4 standard options are constrained to be of maximum size 255 bytes due to the length of the 8-bit option-length field that the protocol defines.

In the case where the option data for any of the DHCPv4 option values exceeds 255 bytes in length, implementations of this specification do not follow [RFC3396] section 5. Instead, Option Code 250 is used to encode the excess data over 255 bytes. Option Code 250 repeats immediately after the option being encoded as many times as necessary to encode the remaining data.

For instance, if the option data for a given Option Code X is 600 bytes, the DHCPv4 client or DHCPv4 server sends Option X with 255 bytes of data, immediately followed by Option 250 with another 255 bytes of data, and then again Option 250 with the remaining 90 (600 – 255 – 255) bytes of data.

Option 250 is encoded by DHCPv4 clients and DHCPv4 servers in the same format as the following standard DHCPv4 options.



Option Code (1 byte): This MUST be 250 (0xFA).

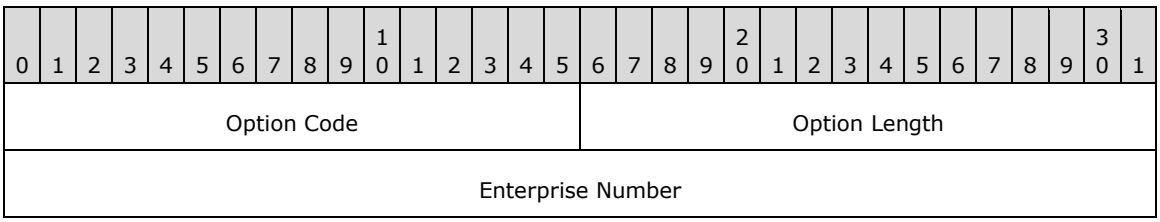
Option Length (1 byte): The unsigned length, in bytes, of the option, not including the **Option Code** and **Option Length** fields.

Option Data (variable): This field contains the continuation of the data of the previous option, which was too long to be contained in that option.

2.2.10 DHCPv6 Option Code 17 (0x0011) - Vendor Specific Information Option

A DHCPv6 client and server exchange vendor-specific information between themselves. This information is sent in the form of a vendor-specific information option, as specified in [RFC3315] section 22.17.

A Validating Server does not need to include a vendor class option (section 2.2.5) when authorizing itself using Rogue Detection.



Copyright © 2015 Microsoft Corporation

Option Data (variable)
...

Option Code (2 bytes): As specified in [RFC3315] section 22.17, this is used to indicate the Vendor-specific information Option and MUST be 0x0011.

Option Length (2 bytes): Set to the size of **Option Data**, in bytes, plus 4.

Enterprise Number (4 bytes): MUST be set to 0x00000137 (decimal 311), the Internet Assigned Numbers Authority (IANA)–assigned Microsoft Enterprise number [IANA-ENT].

Option Data (variable): This is set to the vendor-specific option data.

When a DHCPv6 message includes a Vendor Class Identifier with the value defined in section 2.2.5, the Vendor-Specific Information Option is defined to use the "Encapsulated vendor-specific options" format specified in [RFC3315] section 22.17. This specification defines the following encapsulated vendor-specific option codes.

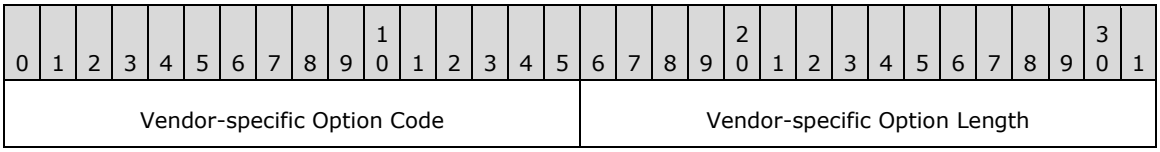
Value	Meaning
0x5E	Rogue Detection Request Option
0x5F	Rogue Detection Reply Option

2.2.10.1 Vendor-Specific Option Code 0x5E – Rogue Detection Request Option

This option is sent by a Validating Server to DHCPv4 servers on the network in a DHCPv6 Information-Request message. It is sent as an encapsulated option in option 17 (section 2.2.10).

The Validating Server MAY include the vendor class option (section 2.2.5) in its DHCPv6 Information-Request message.

The DHCPv6 server requests option 94 (0x005E) in the DHCPv6 Information-Request message in the following format.



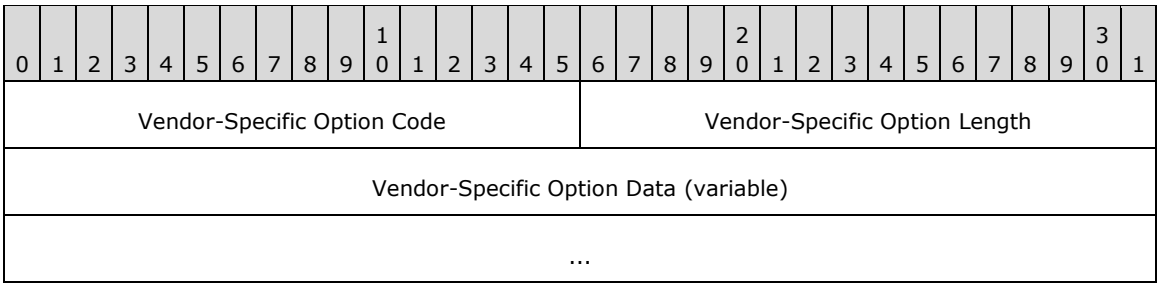
Vendor-specific Option Code (2 bytes): This MUST be 94 (0x5E).

Vendor-specific Option Length (2 bytes): This MUST be 0x00.

2.2.10.2 Vendor-Specific Option Code 0x5F – Rogue Detection Reply Option

This option is sent by a Rogue Aware Server to a Validating Server in a DHCPv6 **Reply** message. It is sent in response to option 0x5E (section 2.2.10.1) received in a DHCPv6 Information-Request message. It is sent as an encapsulated option in option 17 (section 2.2.10).

Copyright © 2015 Microsoft Corporation



Vendor-Specific Option Code (2 bytes): This MUST be 95 (0x5F).

Vendor-Specific Option Length (2 bytes): The unsigned length in bytes of the **Vendor-Specific Option Data** field. The maximum length is 255 bytes.

Vendor-Specific Option Data (variable): This is a null-terminated string of length specified by **Vendor-Specific Option Length**.

2.2.11 DHCPv4 Option Code 15 (0x000f) - Domain Name Option

DHCPv4 Option Code 15 is defined in [RFC2132] section 3.17 which specifies the domain name that the client SHOULD use when resolving host names by using the **DNS**. [RFC1035] provides the specification for domain names, but does not explain how to ASCII-encode a name if the client has a non-ASCII name, and hence the contents of this option are implementation-specific.<17>

Copyright © 2015 Microsoft Corporation

3 Protocol Details

3.1 Client Details

3.1.1 Abstract Data Model

These DHCP extensions comply with the data store (as defined in [RFC2131] section 2.1). The state machine and data model for DHCP are defined in [RFC2131] section 4.4. The data model for DHCPv6 is similar and is defined by [RFC3315].

In addition, DHCP clients also maintain the following state per network interface:

Release DHCP Lease on Shutdown Flag: This flag indicates whether the client will send a **DHCPRELEASE** when it shuts down.

Enable NetBIOS Flag: This flag indicates whether the host has NetBIOS enabled or disabled on the interface.

3.1.2 Timers

None, except the timers in [RFC2131] and [RFC3315].

The DHCPv4 client MUST follow an exponential backoff model for **DHCPDISCOVER** retransmission, as recommended in [RFC2131] section 4.1. However, [RFC2131] does not specify the actual values, so they are specified here. When the DHCPv4 client attempts to allocate a network address as described in [RFC2131] section 3.1, the DHCPv4 client MUST wait for an initial timeout value of 4 seconds for the server to respond. If the server does not respond within the timeout value and the client attempts a retransmission, then for every subsequent retransmission, the client MUST wait double the previous timeout value. For example, on the first transmission the client waits 4 seconds, on the second transmission, 8 seconds, and on the third, 16 seconds. The maximum timeout value allowed for the server to respond SHOULD be 32 seconds. If no response is received to the fourth **DHCPDISCOVER**, the DHCPv4 client MUST wait 5 minutes before repeating the preceding cycle.

DHCPv6 clients implementing this specification adhere to [RFC3315].

3.1.3 Initialization

DHCPv4 client initialization (as specified in [RFC2131]) and DHCPv6 client initialization (as specified in [RFC3315]) are unchanged by the extensions specified in this document.

3.1.4 Higher-Layer Triggered Events

None.

3.1.4.1 Sending a DHCPDISCOVER, DHCPREQUEST, or DHCPINFORM Message

When sending a **DHCPDISCOVER**, **DHCPREQUEST**, or **DHCPINFORM** message, DHCPv4 clients implementing this specification SHOULD<18> include a Vendor Class Identifier Option formatted as in section 2.2.3 and MAY<19> include a User Class Option formatted as in section 2.2.8. Because this specification supports only one user class value in this packet, the client MUST conform to the guidelines for the user class data defined in section 2.2.8.

When sending a **DHCPINFORM** message, DHCPv4 clients implementing this specification MAY<20> include an OPTION_PARAMETER_REQUEST_LIST (Option 55) with OPTION_USER_CLASS (Option 77) as one of the requested options.

Copyright © 2015 Microsoft Corporation

3.1.4.2 Sending a DHCPv6 Solicit, Request, or Information-Request Message

When sending a DHCPv6 **Solicit**, **Request**, or **Information-Request** message, DHCPv6 clients implementing this specification SHOULD<21> include a Vendor Class Option formatted as in section 2.2.5.

When sending an Information-Request message, DHCPv6 clients implementing this specification MAY<22> include an OPTION_ORO (Option 6) with OPTION_USER_CLASS (Option 15) as the only requested option.

3.1.4.3 Sending a DHCPv4 Release or DHCPv6 Release Message

The behavior is same as specified in [RFC2131] section 3.1 for the DHCPv4 **Release** message and [RFC3315] section 18.1.6 for the DHCPv6 **Release** message.

3.1.5 Message Processing Events and Sequencing Rules

DHCPv4 clients process DHCPv4 messages as specified in [RFC2131] sections 3 and 4, with additional behavior as specified in this section.

If the length or the data of the field of any of the options in a DHCPv4 message received by clients implementing this specification is inconsistent, the DHCPv4 client MUST silently discard the DHCPv4 message and restart the initialization process.

3.1.5.1 Receiving a DHCPOFFER

If the **DHCPOFFER** contains any of the options defined in this specification, these options SHOULD be ignored; the client MAY<23> instead use the options to choose among offers in any implementation-specific manner.

When sending a **DHCPREQUEST** in response, the DHCPv4 client SHOULD<24> include a Vendor Class Identifier Option formatted as in section 2.2.3 and MAY<25> include a User Class Option formatted as in section 2.2.6. Because this specification supports only one user class value in this packet, the client MUST conform to the guidelines for the User Class Data defined in section 2.2.6. The DHCPv4 client SHOULD<26> include both options 121 and 249 in the parameter request list in this message.

3.1.5.2 Receiving a DHCPACK

When a DHCPv4 client implementing this specification receives a **DHCPACK** that contains a Vendor-Specific Information Option, it MUST be processed as follows.

If it contains a Microsoft Disable NetBIOS Option, the DHCPv4 client MUST update its NetBIOS Enabled Flag for the interface over which the **DHCPACK** was received, as indicated in section 2.2.2.1.

If it contains a Microsoft Release DHCP Lease on Shutdown Option, the DHCPv4 client MUST update its Release DHCP Lease on Shutdown Flag for the interface over which the **DHCPACK** was received, as indicated in section 2.2.2.2.

If it contains a Microsoft Router Base Metric Option, the value for this option from the **DHCPACK** message MUST be applied by the client for the default routes on that interface.

If it contains a Microsoft Classless Static Route Option, the client MUST first check whether the option conforms to the syntax specified in section 2.2.8. If any of the parameters in this DHCPv4 option are invalid or incomplete, the DHCPv4 client MUST silently discard the complete DHCPv4 message and start the initialization process again. Otherwise, the specified routes MUST be inserted into the routing table in the TCP/IP stack.

Copyright © 2015 Microsoft Corporation

If it contains one or more User Class options (Option 77), the client MUST first check whether each option conforms to the syntax specified in section 2.2.6. If the option does not conform to the syntax, the DHCPv4 client MUST silently discard the complete DHCPv4 message and start the initialization process again. Otherwise, the client uses the information in an implementation-specific manner.

3.1.5.3 Receiving a DHCPv6 Advertise Message

If the DHCPv6 **Advertise** contains any of the V6 options defined in this specification, the client MAY<27> use the options to choose among advertises in any implementation-specific manner.

3.1.5.4 Receiving a DHCPv6 Reply Message

When a DHCPv6 client implementing this specification receives a **Reply** message that contains one or more User Class Options (Option 15), it MUST first check whether the option conforms to the syntax specified in section 2.2.4. If the option does not conform to the syntax, the DHCPv6 client MUST silently discard the option. Otherwise, the DHCPv6 client uses the information in an implementation-specific manner.

3.1.6 Timer Events

DHCPv4 extensions defined in this specification adhere to the RFC standards (as specified in [RFC2131] section 4.4 and in [RFC2132]) for timer events.

The DHCPv6 extensions defined in this specification adhere to the RFC standards (as specified in [RFC3315] section 14 and section 18) for timer events.

3.1.7 Other Local Events

On system shutdown, if its Release DHCP Lease on Shutdown Flag is set, the DHCPv4 client MUST send a **DHCPRELEASE** message for all IP addresses obtained through DHCPv4.

3.1.7.1 DhcpAppendVendorSpecificOption

The higher-level protocol implementations use this method to append the vendor-specific options in the DHCPv4 message packets created in Sending a DHCPDISCOVER, DHCPREQUEST, or DHCPINFORM Message (section 3.1.4.1). When the cumulative size of all the vendor-specific options sent in a message exceeds 255 bytes, then the Microsoft Encoding Long Options Packet, specified in section 2.2.9, MUST be used. The method takes option ID, option data length, and option data as input.

3.1.7.2 DhcpExtractVendorSpecificOption

The higher-level protocol implementations use this method to extract the vendor-specific options in the DHCPv4 message packets received through Receiving a DHCPOFFER (section 3.1.5.1) and Receiving a DHCPACK (section 3.1.5.2). When the cumulative size of all the vendor-specific options sent in a message exceeds 255 bytes, then the Microsoft Encoding Long Options Packet, specified in section 2.2.9, MUST be used.

3.2 Server Details

3.2.1 Abstract Data Model

These DHCP extensions comply with the data store (as defined in [RFC2131] section 2.1). The state machine and data model for DHCPv4 are defined in [RFC2131] section 4.4. The data model for DHCPv6 is similar and is defined by [RFC3315]. The extensions defined in this specification do not require any change to the state machine or the data model of DHCPv4 or DHCPv6.

Copyright © 2015 Microsoft Corporation

This protocol includes the following ADM elements, which are directly accessed from [MS-DHCPM] as specified in [MS-DHCPM] section 3.1.1:

- DHCPv4ClassDef
- DHCPv4Client
- DHCPv4ExclusionRange
- DHCPv4Filter
- DHCPv4FiltersList
- DHCPv4FilterStatus
- DHCPv4IpRange
- DHCPv4OptionValue
- DHCPv4Reservation
- DHCPv4ResvOptValuesList
- DHCPv4Scope
- DHCPv4ServerAttributes
- DHCPv4ServerMibInfo
- DHCPv4ServerOptValueList
- DHCPv4SuperScope
- DHCPv6ClassDef
- DHCPv6ClientInfo
- DHCPv6ExclusionRange
- DHCPv6OptionValue
- DHCPv6Reservation
- DHCPv6ResvClassedOptValueList
- DHCPv6Scope
- DHCPv6ScopeClassedOptValueList
- DHCPv6ServerClassedOptValueList
- DHCPv6ServerMibInfo
- DHCPv6UserClass

3.2.2 Timers

None beyond those in [RFC2131] and [RFC3315].

Copyright © 2015 Microsoft Corporation

3.2.3 Initialization

DHCPv4/DHCPv6 server initialization (as specified in [RFC2131] and [RFC3315]) is unchanged by extensions specified in this document.

3.2.4 Higher-Layer Triggered Events

None.

3.2.5 Message Processing Events and Sequencing Rules

DHCPv4 servers process DHCPv4 messages as specified in [RFC2131] sections 3 and 4. The DHCPv6 server processes messages as specified in [RFC3315]. Additional behavior of DHCPv4 and DHCPv6 servers is specified in this section.

A DHCPv4 or DHCPv6 server is considered unauthorized if the **DHCPv4ServerAttributes.IsRogue** element, a shared element (see [MS-DHCPM] section 3.1.1.26), is set to TRUE. An unauthorized DHCPv4 or DHCPv6 server does not process or respond to any of the messages documented in this section.

3.2.5.1 Receiving a DHCPDISCOVER Message

Increment the **DHCPv4ServerMibInfo.Discovers** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1), as described in section 1.4, point 10.

Evaluate and apply administrative controls as described in section 1.4, point 6.<28>

If the **DHCPDISCOVER** message contains a Vendor Class Identifier Option (section 2.2.3) with a value defined in section 2.2.3, the DHCPv4 server SHOULD ignore the Vendor Class Identifier Option and process the message as if the option were not present. The server MAY<29> instead include any standard option or vendor-specific option defined in this specification in its response (if configured to do so by the administrator) in the **DHCPOFFER** message sent to the clients.

If the **DHCPDISCOVER** contains a Client Identifier Option (Option 61) with the field Client-Identifier containing the first four bytes as "RAS", the DHCPv4 server MUST ignore any FQDN Option (Option 81) in the DHCPv4 message and MUST NOT perform DNS registration of A and PTR records on behalf of the DHCPv4 client. For more information, see [RFC4702].

If processing the **DHCPDISCOVER** message results in the server sending a **DHCPOFFER** message to the client, then increment the **DHCPv4ServerMibInfo.Offers** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1), as described in section 1.4, point 10.

3.2.5.2 Receiving a DHCPREQUEST Message

Evaluate and apply administrative controls as described in section 1.4, point 6.<30>

Increment the **DHCPv4ServerMibInfo.Requests** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1), as described in section 1.4, point 10.

If the **DHCPREQUEST** message contains a Vendor Class Identifier Option (section 2.2.3) with a value defined in section 2.2.3, the following points illustrate the behavior of the DHCPv4 server:

- The DHCPv4 server MUST include the vendor-specific options defined in section 2.2.2 (if configured to do so by the administrator) in the **DHCPACK** message sent to the clients.
- The DHCPv4 server MUST interpret the User Class Option if it exists<31> in the **DHCPREQUEST** message to contain a single value as defined in section 2.2.6.

Copyright © 2015 Microsoft Corporation

When the DHCPv4 server receives a request for Option 249 but not for Option 121 in the Parameter Request List, the Classless Static Route information MUST be returned to the DHCPv4 client in Option 249. If the Parameter Request List contains a request for both Options 121 and 249, the Classless Static Route information SHOULD<32> be returned to the DHCPv4 client in Option 121 only.

The DHCPv4 server MUST format any option values that are longer than 255 bytes, as defined in section 2.2.9.

If processing of the **DHCPREQUEST** message results in the server sending a **DHCPACK** message to the client, then increment the **DHCPv4ServerMibInfo.Acks** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1), as described in section 1.4, point 10.

If processing of the DHCPREQUEST message results in the server sending a **DHCPNACK** message to the client, then increment the **DHCPv4ServerMibInfo.Nacks** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1), as described in section 1.4, point 10.

The remainder of the **DHCPREQUEST** message MUST be processed as specified by [RFC2131] and [RFC2132].

If the **DHCPREQUEST** contains a Client Identifier Option (Option 61) with the field Client-Identifier containing the first four bytes as "RAS ", the DHCPv4 server MUST ignore any FQDN Option (Option 81) in the DHCPv4 message and MUST NOT perform DNS registration of A and PTR records on behalf of the DHCPv4 client. For more information, see [RFC4702].

3.2.5.3 Receiving a DHCPv6 Message with a Vendor Class Option

DHCPv6 servers implementing this specification MAY<33> simply ignore the Vendor Class Option sent in the DHCPv6 messages by the client; the server SHOULD instead return the relevant options configured for clients with the specified vendor class information as specified by [RFC3315].

3.2.5.4 Receiving a DHCPINFORM Message

If the **DHCPINFORM** message does not contain Rogue Detection Request Option (section 2.2.2.4), evaluate and apply administrative controls as described in section 1.4, point 6.<34>

When a DHCPv4 server implementing this specification receives a **DHCPINFORM** message containing an OPTION_PARAMETER_REQUEST_LIST (Option 55) with OPTION_USER_CLASS (Option 77) as one of the requested options, the DHCPv4 server SHOULD<35> send a **DHCPACK** message to the DHCPv4 client containing one or more User Class Options (one for each user class configured on the DHCPv4 Server) in the format specified in section 2.2.6. If the option request is received in a message other than **DHCPINFORM** message, the request is silently ignored.

If the **DHCPINFORM** message contains the Rogue Detection Request option (section 2.2.2.4):

- An Administratively Authorized Server<36> that is Rogue Aware SHOULD<37> reply to the **DHCPINFORM** message by sending a **DHCPACK** message containing the Rogue Detection Reply Option (section 2.2.2.5) with a non-empty NULL-terminated string in the option data.

A Rogue Authorized Server on the network SHOULD<38> reply to the **DHCPINFORM** message by sending a **DHCPACK** message containing the Rogue Detection Reply Option (section 2.2.2.5) with an empty NULL-terminated string in the option data.

3.2.5.5 Receiving an Information-Request Message

Increment the **DHCPv6ServerMibInfo.Informs** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1), as described in section 1.4, point 17.

Copyright © 2015 Microsoft Corporation

When a DHCPv6 server implementing this specification receives an Information-Request message containing OPTION_ORO (Option 6) with OPTION_USER_CLASS (Option 15) as the only requested option, the DHCPv6 server SHOULD<39> send a **DHCPACK** message to the DHCPv6 client containing one or more User Class Options (one for each User Class configured on the DHCPv6 server) in the format specified in section 2.2.4. If other options are also requested in the message, the DHCPv6 server MAY<40>respond to the message. If the message is not an Information-Request message, the option request MUST be silently discarded.

If the DHCPv6 **Information-Request** message contains the Rogue Detection Request option (section 2.2.10.1):

- An Administratively Authorized Server<41> that is Rogue Aware SHOULD<42> reply to the DHCPv6 **Information-Request** message by sending a DHCPv6 **Reply** message containing the Rogue Detection Reply option (section 2.2.10.2) with a non-empty NULL-terminated string in the option data.
- A Rogue Authorized Server on the network SHOULD<43> reply to the DHCPv6 **Information-Request** message by sending a DHCPv6 **Reply** message containing the Rogue Detection Reply option (section 2.2.10.2) with an empty NULL-terminated string in the option data.

If processing of the Information-Request message results in the server sending a DHCPv6 Reply message to the client, then increment the **DHCPv6ServerMibInfo.Replies** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1), as described in section 1.4, point 17.

3.2.5.6 Receiving a DHCP Message with a User Class Option

If the option length or any of the values in the User Class option are inconsistent with the data sent, the DHCP servers implementing this specification MUST silently discard the DHCP message.

3.2.5.7 Receiving a DHCPv4 RELEASE Message

Increment the **DHCPv4ServerMibInfo.Releases** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1), as described in section 1.4, point 10.

Subsequent processing is as specified in [RFC2131] section 4.3.4.

3.2.5.8 Receiving a DHCPv6 Release Message

Increment the **DHCPv6ServerMibInfo.Releases** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1), as described in section 1.4, point 17.

The subsequent processing is as specified in [RFC3315] section 18.2.6.

If processing of the Information-Request message results in the server sending a DHCPv6 Reply message to the client, then increment the **DHCPv6ServerMibInfo.Replies** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1), as described in section 1.4, point 17.

3.2.5.9 Receiving a DHCPDECLINE Message

Increment the **DHCPv4ServerMibInfo.Declines** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1), as described in section 1.4, point 10.

The subsequent processing is as specified in [RFC2131] section 4.3.3.

If processing of the Information-Request message results in the server sending a DHCPv6 Reply message to the client, then increment the **DHCPv6ServerMibInfo.Replies** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1), as described in section 1.4, point 17.

Copyright © 2015 Microsoft Corporation

3.2.5.10 Receiving a DHCPv6 Solicit Message

Increment the **DHCPv6ServerMibInfo.Solicits** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1), as described in section 1.4, point 17.

The subsequent processing is as specified in [RFC3315] section 17.2.1.

If processing of the DHCPv6 Solicit message results in the server sending a DHCPv6 Advertise message to the client, then increment the **DHCPv6ServerMibInfo.Offers** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1), as described in section 1.4, point 17.

3.2.5.11 Receiving a DHCPv6 Request Message

Increment the **DHCPv6ServerMibInfo.Requests** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1), as described in section 1.4, point 17.

The subsequent processing is as specified in [RFC3315] section 18.2.1.

If processing of the Information-Request message results in the server sending a DHCPv6 Reply message to the client, then increment the **DHCPv6ServerMibInfo.Replies** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1), as described in section 1.4, point 17.

3.2.5.12 Receiving a DHCPv6 Confirm Message

Increment the **DHCPv6ServerMibInfo.Confirms** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1), as described in section 1.4, point 17.

The subsequent processing is as specified in [RFC3315] section 18.2.2.

If processing of the Information-Request message results in the server sending a DHCPv6 Reply message to the client, then increment the **DHCPv6ServerMibInfo.Replies** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1), as described in section 1.4, point 17.

3.2.5.13 Receiving a DHCPv6 Renew Message

Increment the **DHCPv6ServerMibInfo.Renews** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1), as described in section 1.4, point 17.

The subsequent processing is as specified in [RFC3315] section 18.2.3.

If processing of the Information-Request message results in the server sending a DHCPv6 Reply message to the client, then increment the **DHCPv6ServerMibInfo.Replies** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1), as described in section 1.4, point 17.

3.2.5.14 Receiving a DHCPv6 Rebind Message

Increment the **DHCPv6ServerMibInfo.Rebinds** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1), as described in section 1.4, point 17.

The subsequent processing is as specified in [RFC3315] section 18.2.4.

If processing of the Information-Request message results in the server sending a DHCPv6 Reply message to the client, then increment the **DHCPv6ServerMibInfo.Replies** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1), as described in section 1.4, point 17.

Copyright © 2015 Microsoft Corporation

3.2.5.15 Receiving a DHCPv6 Decline Message

Increment the **DHCPv6ServerMibInfo.Declines** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1), as described in section 1.4, point 17.

The subsequent processing is as specified in [RFC3315] section 18.2.7.

If processing the Information-Request message results in the server sending a DHCPv6 Reply message to the client, then increment the **DHCPv6ServerMibInfo.Replies** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1), as described in section 1.4, point 17.

3.2.5.16 Receiving a MADCAP DISCOVER Message

Increment the **DHCPv6ServerMibInfo.Declines** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1), as described in section 1.4, item 17.

The subsequent processing is as specified in [RFC3315] section 18.2.7.

If processing the Information-Request message results in the server sending a DHCPv6 Reply message to the client, then increment the **DHCPv6ServerMibInfo.Replies** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1), as described in 1.4, item 17.

3.2.5.17 Receiving a MADCAP REQUEST Message

Increment the **DHCPv4ServerMcastMibInfo.Requests** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1).

The subsequent processing is as specified in [RFC2730] section 2.2.4.

If processing the REQUEST message results in the server sending a MADCAP ACK message to the client, then increment the **DHCPv4ServerMcastMibInfo.Acks** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1).

3.2.5.18 Receiving a MADCAP RENEW Message

Increment the **DHCPv4ServerMcastMibInfo.Renews** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1).

The subsequent processing is as specified in [RFC2730] section 2.2.7.

If processing the RENEW message results in the server sending a MADCAP ACK message to the client, then increment the DHCPv4ServerMcastMibInfo.Acks element, a shared ADM (see [MS-DHCPM] section 3.1.1.1).

If processing the RELEASE message results in the server sending a MADCAP NAK message to the client, then increment the **DHCPv4ServerMcastMibInfo.Naks** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1).

3.2.5.19 Receiving a MADCAP RELEASE Message

Increment the **DHCPv4ServerMcastMibInfo.Releases** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1).

The subsequent processing is as specified in [RFC2730] section 2.2.8.If processing the RELEASE message results in the server sending a MADCAP ACK message to the client, then increment the DHCPv4ServerMcastMibInfo.Acks element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1).

Copyright © 2015 Microsoft Corporation

If processing the RELEASE message results in the server sending a MADCAP NAK message to the client, then increment the **DHCPv4ServerMcastMibInfo.Naks** element, a shared ADM (see [MS-DHCPM] section 3.1.1.1).

3.2.5.20 Receiving a MADCAP GETINFO Message

Increment the **DHCPv4ServerMcastMibInfo.Informs** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1).

The subsequent processing is as specified in [RFC2730] section 2.2.1.

If processing the GETINFO message results in the server sending a MADCAP ACK message to the client, then increment the **DHCPv4ServerMcastMibInfo.Acks** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1).

If processing the GETINFO message results in the server sending a MADCAP NAK message to the client, then increment the **DHCPv4ServerMcastMibInfo.Naks** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.1).

3.2.6 Timer Events

The DHCPv4 extensions adhere to the RFC standards ([RFC2131] section 4.4 and in [RFC2132]) for timer events.

The DHCPv6 extensions adhere to the RFC standard ([RFC3315] section 14 and 18) for timer events.

3.2.7 Other Local Events

None.

3.2.7.1 DhcpAppendVendorSpecificOption

The higher-level protocol implementations use this method to append the vendor-specific options in the DHCPv4 message packets (**DHCPOFFER** and **DHCPACK**) created in Receiving a DHCPDISCOVER Message (section 3.2.5.1), Receiving a DHCPREQUEST Message (section 3.2.5.2), and Receiving a DHCPINFORM Message (section 3.2.5.4). When the cumulative size of all the vendor-specific options being sent in a message exceeds 255 bytes, the Microsoft Encoding Long Options Packet, specified in section 2.2.9, MUST be used. The method takes option ID, option data length, and option data as input.

3.2.7.2 DhcpAppendCSROption

The higher-level protocol implementations use this method to append the Classless Static Route Option as specified in section 2.2.8. This method takes option length and a list of CSR values as input.

3.2.7.3 DhcpExtractVendorSpecificOption

The higher-level protocol implementations use this method to extract the vendor-specific options in the DHCPv4 message packets received through Receiving a DHCPDISCOVER Message (section 3.2.5.1), Receiving a DHCPREQUEST Message (section 3.2.5.2), and Receiving a DHCPINFORM Message (section 3.2.5.4). When the cumulative size of all the vendor-specific options being sent in a message exceeds 255 bytes, the Microsoft Encoding Long Options Packet (section 2.2.9) MUST be used.

Copyright © 2015 Microsoft Corporation

3.3 Validating Server Details

A DHCP server MAY<44> implement the Rogue Detection mechanism. A Rogue Aware Server periodically checks whether it is authorized.<45> This check is done irrespective of the DHCP server authorization state. The time interval between checks MAY<46> vary based on the authorization state.

3.3.1 Abstract Data Model

This protocol includes the following ADM elements, which are shared with and directly accessed by the Dynamic Host Configuration Protocol as specified in [MS-DHCPM] section 3.1.1:

- **DHCPv4ServerAttributes.IsRogue**

3.3.2 Timers

DHCPv4 Authorization Retransmission Timer: This timer is initialized or reset whenever a **DHCPINFORM** message with the Rogue Detection Request Option (section 2.2.2.4) is broadcast by a Validating Server. This is a periodic timer with a default value of 2 seconds. Unless the timer is stopped, it will expire periodically as specified by the timer interval. This timer MUST be stopped when a **DHCPACK** message is received by the Validating Server.

DHCPv6 Authorization Retransmission Timer: This timer is initialized or reset whenever a DHCPv6 Information-Request message with the Rogue Detection Request Option (section 2.2.10.1) is multicast by a Validating Server. This is a periodic timer with a default value of 2 seconds. Unless the timer is stopped, it will expire periodically as specified by the timer interval. This timer MUST be stopped when a DHCPv6 **Reply** message is received by the Validating Server.

Rogue Authorization Recheck Timer: This timer is initialized or reset whenever a rogue detection check is completed irrespective of the DHCP server authorization state resulting from it. This is a periodic timer with a default value of 1 hour. Unless the timer is stopped, it will expire periodically as specified by the timer interval.

3.3.3 Initialization

A Validating Server MUST initialize the **DHCPv4ServerAttributes.IsRogue** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.26), to TRUE because it has not yet performed the authorization process.

3.3.4 Higher-Layer Triggered Events

The higher-layer triggered events relating to server validation are specified in the following sections.

3.3.4.1 Sending a DHCPINFORM Message

A Validating Server MUST open **UDP** port 68 on the interfaces bound to the DHCPv4 server. If the server does not contain bound interfaces, the server cannot validate itself.

A Validating Server MUST send a broadcast **DHCPINFORM** message to address 255.255.255.255 with the Rogue Detection Request Option (section 2.2.2.4) on all the interfaces bound to the DHCPv4 server.

3.3.4.2 Sending a DHCPv6 Information-Request Message

A Validating Server MUST open UDP port 546 on the interfaces bound to the DHCPv6 server. If the Validating Server does not contain bound interfaces, it cannot validate itself.

Copyright © 2015 Microsoft Corporation

A Validating Server MUST send a DHCPv6 **Information-Request** message with the Rogue Detection Request Option (section 2.2.10.1) on all the interfaces bound to the DHCPv6 server.

3.3.5 Message Processing Events and Sequencing Rules

The message processing events and sequencing rules for the messages relating to server validation are specified in the following sections.

3.3.5.1 Receiving a DHCPACK Message

If a Validating Server receives a **DHCPACK** message containing a non-empty Rogue Detection Reply option (section 2.2.2.5), it MUST consider itself unauthorized and set the **DHCPv4ServerAttributes.IsRogue** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.26), to TRUE.

If a Validating Server receives a **DHCPACK** message that either does not contain a Rogue Detection Reply option (section 2.2.2.5) or contains an empty one, it retries sending the **DHCPINFORM** message.

The maximum number of retries is implementation-specific.<47> When all attempts are exhausted, the Validating Server MAY<48> consider itself authorized and set the **DHCPv4ServerAttributes.IsRogue** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.26), to FALSE or continue validation using the DHCPv6 **Information-Request** message.

3.3.5.2 Receiving a DHCPv6 Reply Message

If a Validating Server receives a DHCPv6 **Reply** message containing a non-empty Rogue Detection Reply option (section 2.2.10.2), it will consider itself unauthorized and set the **DHCPv4ServerAttributes.IsRogue** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.26), to TRUE.

If a Validating Server receives a DHCPv6 **Reply** message that either does not contain a Rogue Detection Reply option (section 2.2.10.2) or contains an empty one, it retries sending the DHCPv6 **Information-Request** message.

The maximum number of retries is implementation-specific.<49> After all retry attempts are exhausted, the Validating Server MUST consider itself authorized and set the **DHCPv4ServerAttributes.IsRogue** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.26), to FALSE.

3.3.6 Timer Events

DHCPv4 Authorization Retransmission Timer: The expiration of this timer denotes a period during which the Validating Server has broadcast a **DHCPINFORM** message with the Rogue Detection Request Option (section 2.2.2.4) but has not received a corresponding **DHCPACK** message. The following are the tasks to be performed when this timer expires:

- Retry the message up to an implementation-specific number of attempts.<50>
- If the maximum number of retry attempts has not been reached, retransmit the **DHCPINFORM** message with the Rogue Detection Request Option (section 2.2.2.4).<51>
- When all attempts are exhausted, the Validating Server MAY<52> consider itself authorized and set the **DHCPv4ServerAttributes.IsRogue** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.26), to FALSE or continue validation using the DHCPv6 **Information-Request** message.

Copyright © 2015 Microsoft Corporation

DHCPv6 Authorization Retransmission Timer: The expiration of this timer denotes a period during which the Validating Server has broadcast a DHCPv6 **Information-Request** message with the Rogue Detection Request Option (section 2.2.10.1) but has not received a corresponding DHCPv6 **Reply** message. The following are the tasks to be performed when this timer expires:

- Retry the message up to an implementation-specific number of attempts.<53>
- If the maximum number of retry attempts has not been reached, retransmit the DHCPv6 **Information-Request** message with the Rogue Detection Request Option (section 2.2.10.1).<54>
- When all attempts are exhausted, the Validating Server MAY consider itself authorized and set the **DHCPv4ServerAttributes.IsRogue** element, a shared ADM element (see [MS-DHCPM] section 3.1.1.26), to FALSE.

Rogue Authorization Recheck Timer: The expiration of this timer denotes that the rogue authorization state of the DHCP server needs to be reestablished. The following are the tasks to be performed when this timer expires:

- Initiate rogue detection using DHCPINFORM messages as specified in section 3.3.5.1.

3.3.7 Other Local Events

None.

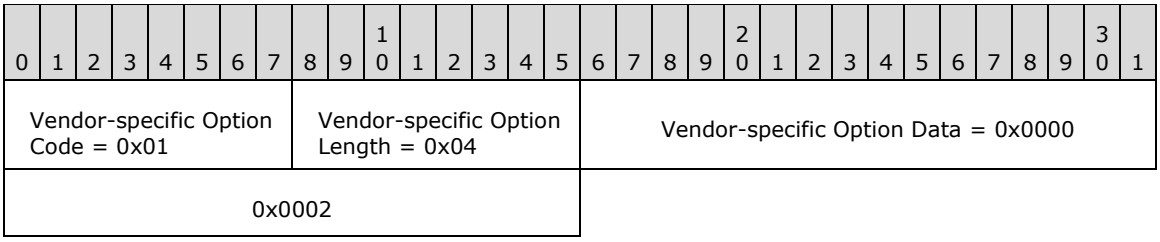
Copyright © 2015 Microsoft Corporation

4 Protocol Examples

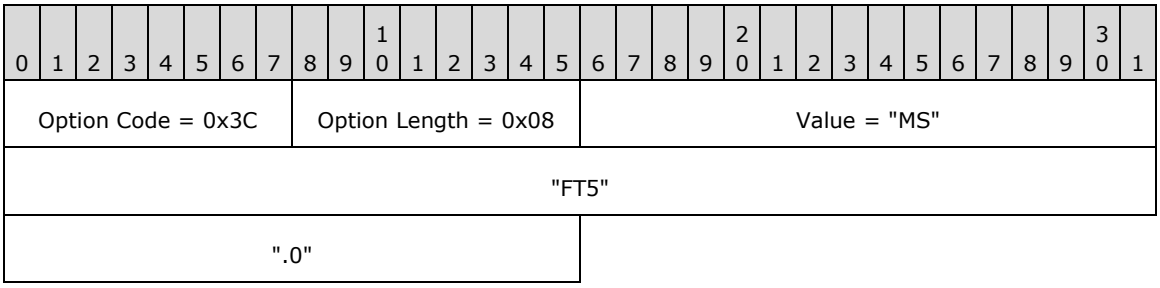
The message exchanges described for DHCPv4 are specified in [RFC2131] section 3. Message exchanges for DHCPv6 are specified in [RFC3315]. The message sequences and the operation of DHCPv4 (as specified in [RFC2131] section 4.1) and DHCPv6 (as specified in [RFC3315]) are unchanged by this extension.

In this example, an administrator wants to prevent clients from using NetBIOS over TCP/IP on the local network.

1. The administrator configures the DHCPv4 server to send Vendor-specific Option number 1 with option value as 2, as specified in section 2.2.2.1, to the DHCPv4 clients.



2. The DHCPv4 client joins the network and sends a **DHCPDISCOVER** message that includes a Vendor Class Identifier Option. For instance, the DHCPv4 client sends the vendor class as "MSFT 5.0".



3. The DHCPv4 server ignores the Vendor Class Identifier Option and responds with a **DHCPOFFER** message. It does not include any option defined in this specification in the **DHCPOFFER** message. The DHCPv4 client accepts the offer by sending a **DHCPREQUEST** message again that includes the Vendor Class Identifier Option as before.

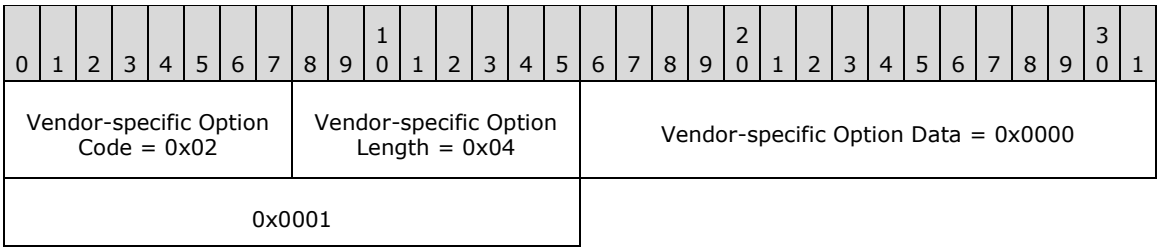
4. The DHCPv4 server recognizes the value in the Vendor Class Identifier Option in the DHCPv4 message from the client and sends a **DHCPACK** message that includes Vendor-specific Option number 1 as previously shown.

5. The DHCPv4 client will receive this Vendor-specific Option from the DHCPv4 server and disable the use of NetBIOS over TCP/IP.

In another example, an administrator wants DHCPv4 clients on the local network to release the DHCPv4 address lease when the machine is shut down.

1. The administrator can configure the DHCPv4 server to send Vendor-specific Option number 2 with value 1, as described in section 2.2.2.2, to the DHCPv4 clients.

Copyright © 2015 Microsoft Corporation



2. The DHCPv4 client joins the network and sends a **DHCPDISCOVER** message that includes a Vendor Class Identifier Option. For instance, the DHCPv4 client may send the vendor class as "MSFT 5.0".

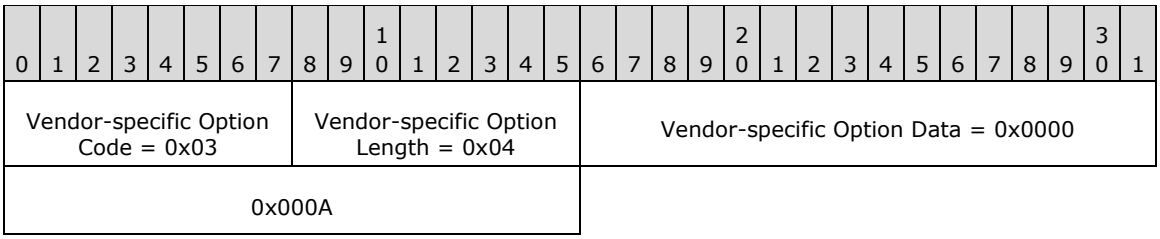
3. The DHCPv4 server ignores the Vendor Class Identifier Option and responds with a **DHCPOFFER** message. It does not include any option defined in this specification in the **DHCPOFFER** message. The DHCPv4 client accepts the offer by sending a **DHCPREQUEST** message again that includes a Vendor Class Identifier Option as before.

4. The DHCPv4 server recognizes the value in the Vendor Class Identifier Option in the DHCPv4 message from the client and sends a **DHCPACK** message that includes Vendor-specific Option number 2 as previously shown.

5. The DHCPv4 client will receive this Vendor-specific Option from the DHCPv4 server and release its DHCPv4 address lease when the machine is shut down.

In another example, an administrator wants to change the router metric used by DHCPv4 clients connecting to the local network.

1. The administrator configures the DHCPv4 server to send Vendor-specific Option number 3 with the desired routing metric (say 10), as specified in section 2.2.2.3, to the DHCPv4 clients.



2. The DHCPv4 client joins the network and sends a **DHCPDISCOVER** message that includes a Vendor Class Identifier Option with, for example, the value "MSFT 5.0".

3. The DHCPv4 server ignores the Vendor Class Identifier Option and responds with a **DHCPOFFER** message. It does not include any option defined in this specification in the **DHCPOFFER** message. The DHCPv4 client accepts the offer by sending a **DHCPREQUEST** message again that includes a Vendor Class Identifier Option as before.

4. The DHCPv4 server recognizes the Vendor Class Identifier Option in the DHCPv4 message from the client and sends a **DHCPACK** message that includes Vendor-specific Option number 3 as shown previously.

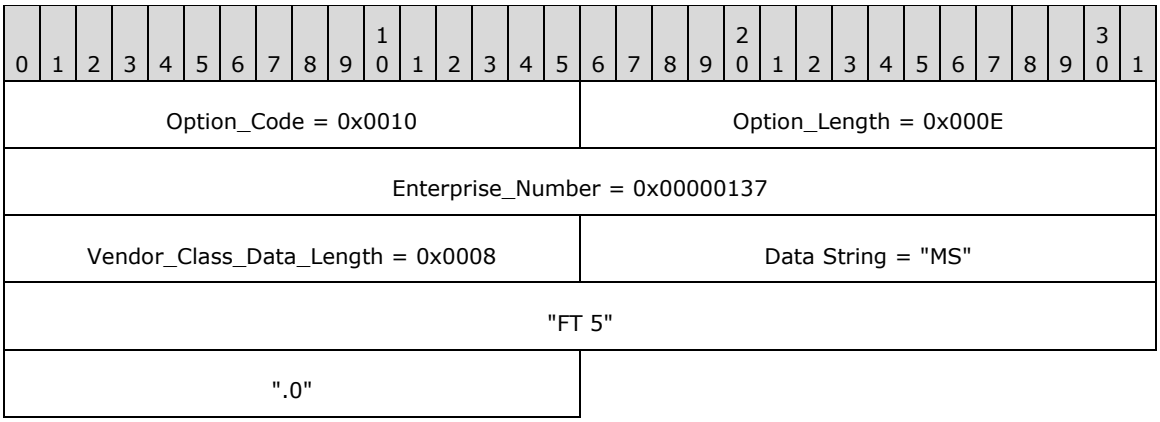
5. The DHCPv4 client will receive this Vendor-specific Option from the DHCPv4 server and use the appropriate router-metric value (in this example, 10) as specified by the DHCPv4 server on that network interface.

If an administrator wants to send Vendor-specific Information to clients through DHCPv6 on the local network, this can be done based on the vendor class identifier.

Copyright © 2015 Microsoft Corporation

1. The administrator configures the DHCPv6 server to send the desired information to clients if the vendor-class identifier received from the client is "MSFT 5.0" as described previously.

2. The DHCPv6 client joins the local network and sends a DHCPv6 **Solicit** message that includes a Vendor Class Option (section 2.2.5). For instance, the DHCPv6 client sends the vendor-class data as "MSFT 5.0".



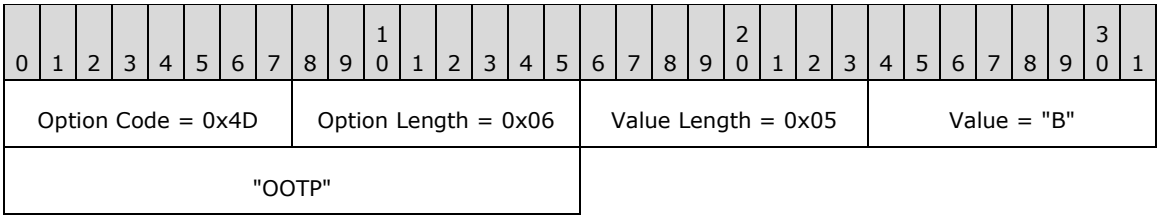
3. The DHCPv6 server ignores the Vendor Class Option and responds with a DHCPv6 **Advertise** message. The DHCPv6 client accepts the offer by sending a DHCPv6 **Request** message, again including a Vendor Class Option as before.

4. The DHCPv6 server interprets the vendor-class identifier sent by the DHCPv6 client in the DHCPv6 **Request** message and sends the appropriate standard options to the DHCPv6 client in the DHCPv6 **Reply** message. Depending on the DHCPv6 server configuration, the option values selected by the server for inclusion in the **Reply** message may be based on the Vendor Class Option value sent by the client in the **Request** message.

5. The DHCPv6 client receives and applies the option information sent by the DHCPv6 server.

In another example, an administrator wants to send specific information to DHCPv4 clients on the local network when the administrator sends the DHCPv4 message as a BOOTP message.

1. In this case, the administrator can configure the DHCPv4 server to look for the User-Class option containing a User-Class subpacket with the value "BOOTP" (as described in section 2.2.6) in the DHCPv4 message sent by the DHCPv4 client. If the message contains this user-class subpacket, the DHCPv4 server is configured to respond with the desired information that the administrator wants to send to the DHCPv4 client.



2. DHCPv4 clients that send a DHCPv4 message as a BOOTP message (see [RFC1534] section 2) will include a User Class Option in the message containing the User Class subpacket with the value "BOOTP" as previously shown. Thus, if the DHCPv4 server is configured as explained previously, the DHCPv4 client will receive the desired information in the response from the server.

As an example of the use of the Microsoft Classless Static Route Option, see the examples on pages 4 and 5 of [RFC3442], with the only difference being that the code used for this option is 249 instead of option code 121, used in [RFC3442].

Copyright © 2015 Microsoft Corporation

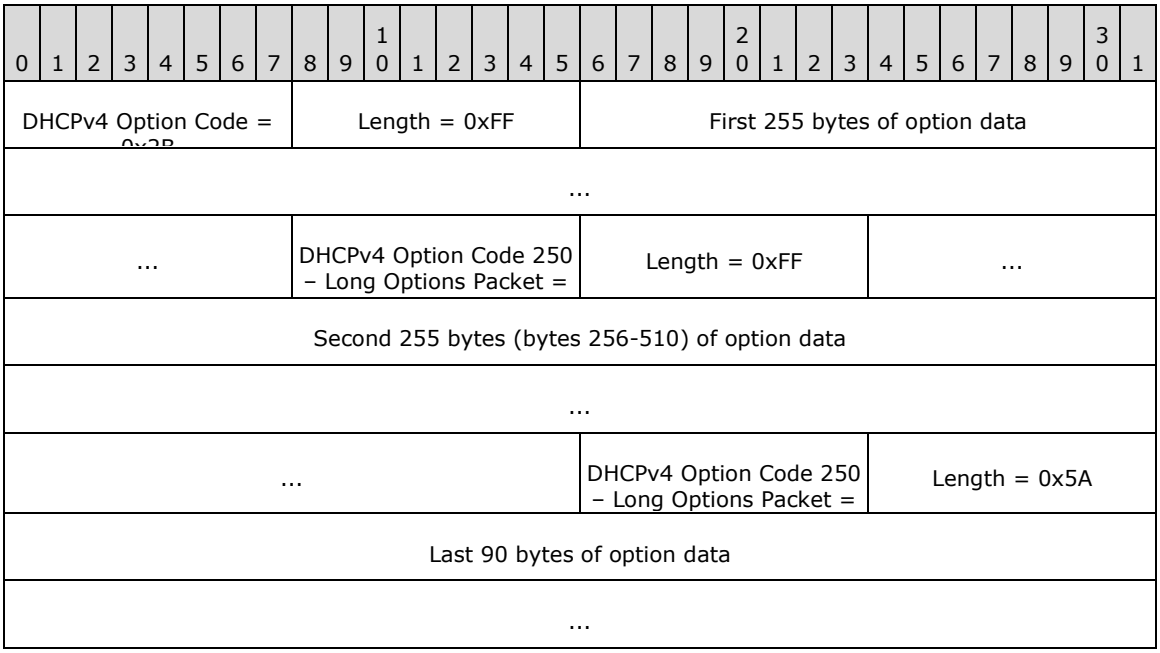
In another example, say that an administrator wants to send Vendor-specific Information through DHCPv4 to a DHCPv4 client on the local network as DHCPv4 Vendor-Specific Information Option 43 (0x2B). However, this information, when encapsulated in Option 43 as per [RFC2132], is 600 bytes, exceeding the 255-byte limit of a DHCPv4 option length.

1. The administrator configures the DHCPv4 server to send Vendor-specific Options to the client.

2. The DHCPv4 client joins the network and sends a **DHCPDISCOVER** message that includes a Vendor Class Identifier Option with, for example, the value "MSFT 5.0".

3. The DHCPv4 server ignores the Vendor Class Identifier Option and responds with a **DHCPOFFER** message. It does not include any option defined in this specification in the **DHCPOFFER** message. The DHCPv4 client accepts the offer by sending a **DHCPREQUEST** message again, including the Vendor Class Identifier Option as before.

4. The DHCPv4 server recognizes the value in the Vendor Class Identifier Option in the DHCPv4 message from the client and sends a **DHCPACK** message that includes the Vendor-Specific Information option with the desired value as configured by the administrator, while formatting it as described in section 2.2.9, by sending Option 43 (0x2B) of size 255 bytes, followed by Option 250 with the next 255 bytes, and then again Option 250 with the remaining 90 bytes.



5. DHCPv4 clients on the local network that initiate a DHCPv4 transaction with the preceding server will thus receive the configured Vendor-specific Information that exceeds 255 bytes. Similarly, standard option values that exceed 255 bytes can also be sent to clients by formatting the options as described in section 2.2.9.

The following example demonstrates the use of User Class option when a user wants to see all User Classes configured on DHCPv4 server by an administrator.

1. The administrator configures a User Class on the DHCPv4 server with name as "test", description as "desc" and binarydata as "123".

2. When a DHCPv4 client gets connected in this network, the client gets the IP and other configuration information from the DHCPv4 server.

Copyright © 2015 Microsoft Corporation

3. The user instructs the DHCPv4 client in an implementation-specific manner to retrieve all User Classes on the DHCPv4 server. The DHCPv4 client sends a **DHCPINFORM** packet containing the Option 55 to theDHCPv4 server requesting Option 77, shown as follows.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Option Code = 55								Option Length = 1 byte								Option Requested = 77															

4. In the reply to the **DHCPINFORM** packet, the server sends **DHCPACK** with the Option 77 record in it. The format of the option record is as follows.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Option Code = 77								Option Length = 30 bytes								User Class Binary Data Length = 3															
'1'								'2'								'3'								'\0' (Padding)					
User Class Name Length = 10 (0x0A)																'\0'								'T'					
'\0'								'E'								'\0'								'S'					
'\0'								'T'								'\0'								'\0'					
User Class Description Length = 10 (0x0A)																'\0'								'D'					
'\0'								'E'								'\0'								'S'					
'\0'								'C'								'\0'								'\0'					

Copyright © 2015 Microsoft Corporation

5 Security

5.1 Security Considerations for Implementers

All of the security considerations that are applicable to DHCPv4 (as described in [RFC2131] section 7) and DHCPv6 (as described in [RFC3315] section 23) apply to the implementation of this specification.

5.2 Index of Security Parameters

None.

Copyright © 2015 Microsoft Corporation

6 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs.

Note: Some of the information in this section is subject to change because it applies to a preliminary product version, and thus may differ from the final version of the software when released. All behavior notes that pertain to the preliminary product version contain specific references to it as an aid to the reader.

- Microsoft Windows 98 operating system
- Windows 2000 operating system
- Windows Millennium Edition operating system
- Windows XP operating system
- Windows Server 2003 operating system
- Windows Vista operating system
- Windows Server 2008 operating system
- Windows 7 operating system
- Windows Server 2008 R2 operating system
- Windows 8 operating system
- Windows Server 2012 operating system
- Windows 8.1 operating system
- Windows Server 2012 R2 operating system
- Windows 10 operating system
- Windows Server 2016 Technical Preview operating system

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

<1> Section 2.2.1: In all versions of Windows, except Windows 8, Windows 8.1, Windows 10, Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016 Technical Preview, DHCP clients send the host name encoded using the **original equipment manufacturer (OEM) code page** that was installed as the current system code page at system boot time in Option 12. In all versions of Windows, except Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016 Technical Preview, DHCP servers treat the host name as being encoded using the **OEM code page** that was installed on the DHCP server as the current system code page at system boot time.

In Windows 8, Windows 8.1, Windows 10, Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016 Technical Preview, the DHCP client sends the host name option in the OEM code page in the **DHCPDISCOVER** message (see section 3.1.4.1). In the **DHCPREQUEST** message (see

Copyright © 2015 Microsoft Corporation

section 3.1.4.1), if the FQDN option is sent in **canonical IDNA** (see section 2.2.7), then the host name option is sent in **IDNA** in the same message. If the FQDN option is not sent in canonical IDNA encoding or if the FQDN option is not sent in the **DHCPREQUEST**, then the host name is sent in the OEM code page in the same message.

<2> Section 2.2.2.1: Windows DHCPv4 servers send this option in the **DHCPOFFER** and **DHCPACK** messages, if configured to do so by the administrator.

<3> Section 2.2.2.1: Windows 98 and Windows Millennium Edition DHCPv4 clients do not support this option.

<4> Section 2.2.2.2: Windows 98 and Windows Millennium Edition DHCPv4 clients do not support this option.

<5> Section 2.2.2.3: Windows 98 and Windows Millennium Edition DHCPv4 clients do not support this option.

<6> Section 2.2.2.3: By default (if not overridden by this option), the TCP/IP stack instead computes the route metric based on link speed as follows for Windows 98, Windows Millennium Edition, Windows 2000, Windows XP, and Windows XP operating system Service Pack 1 (SP1) clients.

Link speed	Metric
Greater than 200 Mbps	0x0000000A (10)
Greater than 20 Mbps, and less than or equal to 200 Mbps	0x00000014 (20)
Greater than 4 Mbps, and less than or equal to 20 Mbps	0x0000001E (30)
Greater than 500 Kbps, and less than or equal to 4 Mbps	0x00000028 (40)
Less than or equal to 500 Kbps	0x00000032 (50)

<7> Section 2.2.4: Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows 10, Windows Server 2012 R2, and Windows Server 2016 Technical Preview DHCPv6 clients request the User Classes defined on the Windows DHCPv6 server whenever a user tries to set the User Class for the DHCPv6 client by executing "Ipconfig /setclassid6" or whenever a user tries to see the User Classes defined on the DHCPv6 server by executing "Ipconfig /showclassid6". DHCPv6 clients on Windows Vista operating system with Service Pack 1 (SP1) and Windows Server 2008 do not support the User Class Option.

<8> Section 2.2.4: Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016 Technical Preview send one or more User Class Options depending on whether one or more User Classes are configured on the DHCP server. DHCPv6 server on Windows Server 2008 does not support User Class Option.

<9> Section 2.2.5: DHCPv6 Client support is implemented in Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows 10, and Windows Server 2012 R2, and Windows Server 2016 Technical Preview. DHCPv6 server support is available in Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016 Technical Preview.

<10> Section 2.2.6.1: By default, Windows DHCPv4 clients do not send the User Class Option in the DHCPv4 messages. Users can configure any data string value to be sent as the user class value by the DHCPv4 client to the server.

<11> Section 2.2.6.2: DHCPv4 clients request the User Classes defined on the Windows DHCP server whenever a user tries to set the User Class for the DHCPv4 client by executing "Ipconfig /setclassid"

Copyright © 2015 Microsoft Corporation

or whenever a user tries to see the User Classes defined on the DHCPv4 server by executing "Ipconfig /showclassid".

<12> Section 2.2.6.2: Windows Server operating system sends on or more User Class Options depending on whether one or more User Classes are configured on the DHCPv4 server.

<13> Section 2.2.7: All Windows DHCPv4 clients send FQDNs in Option 81 with the E bit set to 0 by default but can be configured to send with the E bit set to 1.

<14> Section 2.2.7: When sending with the E bit set to 0, all Windows DHCPv4 clients encode the host name using the OEM code page that was installed as the current system code page at system boot time. In all versions of Windows, except Windows 8, Windows 8.1, Windows 10, Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016 Technical Preview, when sending with the E bit set to 1, the host name is encoded using **UTF-8** encoding.

In Windows 8, Windows 8.1, Windows 10, Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016 Technical Preview, the registry value "DhcpUseE1" of type REG_DWORD under the registry key HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\<GUID of the interface> is read. If "DhcpUseE1" has a nonzero value, the host name is encoded in canonical IDNA and is sent with the E bit set to 1. If "DhcpUseE1" has a zero value or if the registry value does not exist, then the registry value "DhcpUseUTF8" of type REG_DWORD under the same registry key is read. If "DhcpUseUTF8" is nonzero, then the host name is encoded using UTF-8 encoding and is sent with the E bit set to 1. If the registry value has a zero value or if it doesn't exist, then the host name is sent with the E bit set to 0 using the OEM code page.

<15> Section 2.2.7: Windows DHCPv4 servers treat FQDNs with the E bit set to 0 in Option 81 as being encoded using the OEM code page that was installed on the DHCPv4 server as the current system code page at system boot time. In all versions of Windows Server except Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016 Technical Preview, DHCPv4 servers treat FQDNs with the E bit set to 1 as being encoded using UTF-8 encoding. In Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016 Technical Preview, DHCPv4 servers check FQDNs with the E bit set to 1 to determine if the FQDN is encoded in canonical IDNA. If the FQDN is not encoded in that manner, the FQDN is treated as being encoded by using UTF-8 encoding.

<16> Section 2.2.8: Windows XP and Windows Server 2003 DHCPv4 clients and servers use Option Code 249 for requesting and sending Classless Static Routes (CSRs) instead of Option Code 121, as specified in [RFC3442]. These clients and servers ignore Option 121 if included in a DHCPv4 message.

Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows 10, Windows Server 2012 R2, and Windows Server 2016 Technical Preview DHCPv4 clients use both Option 121 and Option 249.

<17> Section 2.2.11: In all versions of Windows Server except Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016 Technical Preview, the DHCP server sends the domain name encoded by using the original equipment manufacturer (OEM) code page that was installed as the current system **code page** at system boot time. In all versions of Windows except Windows 8, Windows 8.1, Windows 10, Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016 Technical Preview, the DHCP client treats the domain name as though it were encoded using the OEM code page that was installed on the DHCP client as the current system code page at system boot time.

In Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016 Technical Preview, the DHCP server sends the DHCPv4 Option Code 15 in the IDNA encoding in the **DHCPOFFER** message (see 3.1.5.1). If the client FQDN option (see 2.2.7) was sent in the canonical IDNA encoding by the DHCP client in the **DHCPREQUEST** message (see 3.1.4.1), then DHCPv4 Option Code 15 is sent in the IDNA encoding in the **DHCPACK** message (see 3.1.5.2). If the client FQDN option was not sent in the canonical IDNA encoding by the client in the **DHCPREQUEST**, then the domain name is sent in the OEM code page in the **DHCPACK**.

Copyright © 2015 Microsoft Corporation

<18> Section 3.1.4.1: All Windows DHCPv4 clients include Vendor Class Identifier Option (Option 60) in **DHCPDISCOVER**, **DHCPREQUEST**, and **DHCPINFORM** messages.

<19> Section 3.1.4.1: By default, Windows DHCPv4 clients do not send the User Class Option in the DHCPv4 messages. Users can configure any data string value to be sent as the user class value by the DHCPv4 client to the server.

Windows DHCPv4 clients using BOOTP to boot from the network send the Default BOOTP class (as defined in section 2.2.6) as their user class.

<20> Section 3.1.4.1: DHCPv4 clients request the User Classes defined on the Windows DHCPv4 server whenever a user tries to set the User Class for the DHCPv4 client by executing "Ipconfig /setclassid" or whenever a user tries to see the User Classes defined on the DHCPv4 server by executing "Ipconfig /showclassid".

<21> Section 3.1.4.2: All Windows DHCPv6 clients include a Vendor Class Option (Option 16) in DHCPv6 **Solicit**, **Request**, and Information-Request messages.

<22> Section 3.1.4.2: Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows 10, Windows Server 2012 R2, and Windows Server 2016 Technical Preview DHCPv6 clients request the User Classes defined on the Windows DHCPv6 server whenever a user tries to set the User Class for the DHCPv6 client by executing "Ipconfig /setclassid6" or whenever a user tries to see the User Classes defined on the DHCPv6 server by executing "Ipconfig /showclassid6". DHCPv6 clients on Windows Vista and Windows Server 2008 do not support the User Class Option.

<23> Section 3.1.5.1: Windows DHCPv4 clients parse each of the options in the **OFFER** received and silently discard the message if any of the options do not conform to the syntax. If the DHCPv4 client has requested a particular IP Address, it chooses the **OFFER** which has value of allocated IP address same as the requested IP address.

<24> Section 3.1.5.1: All Windows DHCPv4 clients include a Vendor Class Identifier Option (Option 60) in **DHCPDISCOVER**, **DHCPREQUEST**, and **DHCPINFORM** messages.

<25> Section 3.1.5.1: By default, Windows DHCPv4 clients do not send the User Class Option in the DHCPv4 messages. Users can configure any data string value to be sent as the user class value by the DHCPv4 client to the server.

Windows DHCPv4 clients using BOOTP to boot from the network send the Default BOOTP class (as defined in section 2.2.6) as their user class.

<26> Section 3.1.5.1: Windows XP and Windows Server 2003 DHCPv4 clients request only option code 249 in the Parameter Request List. Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows 10, Windows Server 2012 R2, and Windows Server 2016 Technical Preview DHCPv4 clients request both option code 121 and option code 249 in the Parameter Request List.

<27> Section 3.1.5.3: Windows DHCPv6 clients parse each of the options in the ADVERTISE received and silently discard the message if any of the options do not conform to the syntax.

<28> Section 3.2.5.1: Administrative controls are implemented only by Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016 Technical Preview DHCPv4 servers.

<29> Section 3.2.5.1: Windows DHCPv4 servers includes all the standard and requested options in the reply.

<30> Section 3.2.5.2: Administrative controls are implemented only by Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016 Technical Preview DHCPv4 servers.

Copyright © 2015 Microsoft Corporation

<31> Section 3.2.5.2: Windows DHCPv4 servers interpret all unrecognized User classes (including cases where the client sends a User Class Option of length zero or where the client does not send the User Class Option) to be the Default User class.

<32> Section 3.2.5.2: Windows 2000 and Windows Server 2003 DHCPv4 servers send the Classless Static Route information to clients in Option 249, even if the client requests both option code 121 and option code 249. Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016 Technical Preview send the Classless Static Route information to clients in Option 121 if the client requests both option code 121 and option code 249 in the Parameter Request List.

<33> Section 3.2.5.3: Windows DHCP servers ignore the Vendor Class Option.

<34> Section 3.2.5.4: Administrative controls are implemented by Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016 Technical Preview.

<35> Section 3.2.5.4: Windows Server sends one or more User Class Options depending on whether one or more User Classes are configured on the DHCPv4 server.

<36> Section 3.2.5.4: For Windows 2000 Server operating system, Windows Server 2003, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016 Technical Preview, a DHCPv4 server can be administratively authorized using the mechanism specified in Appendix B.

<37> Section 3.2.5.4: DHCPv4 servers on Windows 2000 Server, Windows Server 2003, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016 Technical Preview do not reply if no subnets are configured on the DHCPv4 server.

<38> Section 3.2.5.4: DHCPv4 servers on Windows 2000 Server, Windows Server 2003, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016 Technical Preview do not reply if no subnets are configured on the DHCPv4 server.

<39> Section 3.2.5.5: Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016 Technical Preview DHCPv6 servers send one or more User Class Options depending on whether one or more User Classes are configured on the DHCPv6 server. DHCPv6 server on Windows Server 2008 does not support User Class Option.

<40> Section 3.2.5.5: Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016 Technical Preview DHCPv6 servers send one or more User Class Options depending on whether one or more User Classes are configured on the DHCPv6 server. DHCPv6 server on Windows Server 2008 does not support User Class Option.

<41> Section 3.2.5.5: A DHCP server can be administratively authorized using the mechanism specified in Appendix B for Windows 2000 Server, Windows Server 2003, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016 Technical Preview.

<42> Section 3.2.5.5: Windows 2000 Server, Windows Server 2003, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016 Technical Preview DHCP servers reply even if no subnets are configured on the DHCPv6 server.

<43> Section 3.2.5.5: Windows 2000 Server, Windows Server 2003, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016 Technical Preview DHCP servers reply even if no subnets are configured on the DHCPv6 server.

<44> Section 3.3: DHCP server on Windows 2000 Server, Windows Server 2003, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016 Technical Preview implement Rogue Detection.

Copyright © 2015 Microsoft Corporation

<45> Section 3.3: An authorization check is performed after every one hour by default by **DHCP** server on Windows 2000 Server, Windows Server 2003, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016 Technical Preview. The time interval after which the authorization check should be done is configurable and the minimum time interval is 5 minutes.

<46> Section 3.3: The time interval after which the authorization check is done does not vary based on the authorization state of a DHCP server on Windows 2000 Server, Windows Server 2003, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016 Technical Preview.

<47> Section 3.3.5.1: The maximum number of retries for sending the **DHCPINFORM** message in the Rogue Detection mechanism is 4 for DHCP servers on Windows 2000 Server, Windows Server 2003, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016 Technical Preview.

<48> Section 3.3.5.1: In Rogue Detection, after all retry attempts to send the **DHCPINFORM** message are exhausted, DHCP servers on Windows 2000 Server and Windows Server 2003 consider themselves authorized. DHCP servers on Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016 Technical Preview continue validation using the DHCPv6 **Information-Request** message.

<49> Section 3.3.5.2: The maximum number of retries for sending DHCPv6 **Information-Request** messages in Rogue Detection is 4 for Windows 2000 Server, Windows Server 2003, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016 Technical Preview DHCP servers.

<50> Section 3.3.6: The maximum number of retries is 4 for DHCP servers on Windows 2000 Server, Windows Server 2003, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016 Technical Preview.

<51> Section 3.3.6: Implementations on Windows 2000 Server, Windows Server 2003, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016 Technical Preview wait 2 seconds after sending a **DHCPINFORM** message to receive **DHCPACK** messages when validating DHCP server authorization using Rogue Detection.

<52> Section 3.3.6: In Rogue Detection, after all retry attempts to send **DHCPINFORM** messages are exhausted, DHCP servers on Windows 2000 Server and Windows Server 2003 consider themselves authorized. DHCP servers on Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016 Technical Preview continue validation using DHCPv6 **Information-Request** messages.

<53> Section 3.3.6: The maximum number of retries for sending DHCPv6 **Information-Request** messages in Rogue Detection is 4 for Windows 2000 Server, Windows Server 2003, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016 Technical Preview DHCP servers.

<54> Section 3.3.6: Implementations on Windows 2000 Server, Windows Server 2003, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016 Technical Preview wait 2 seconds after sending an **Information-Request** message to receive **Reply** messages when validating DHCP server authorization using Rogue Detection.

Copyright © 2015 Microsoft Corporation

7 Appendix B: Administrative Authorization of Windows DHCP server

The information in this section is applicable to the following Microsoft products:

- Windows 2000 Server operating system
- Windows Server 2003 operating system
- Windows Server 2008 operating system
- Windows Server 2008 R2 operating system
- Windows Server 2012 operating system
- Windows Server 2012 R2 operating system
- Windows Server 2016 Technical Preview operating system

7.1 Windows DHCP Server Authorization in Domain Joined Scenario

A domain joined Windows server with DHCP server deployed can validate itself. Authorization mechanism of a DHCP server in a domain joined scenario is as follows:

- A domain joined DHCP server is authorized by a domain administrator in **Active Directory Domain Services (AD DS)**. Any DHCP server which is domain joined and is required to service DHCP clients must have an **Active Directory** object in the Active Directory.
- The DHCP server validates its authorization in AD DS every hour. It uses **LDAP** protocol [MS-ADTS] for the purpose of communicating with the Active Directory and validating whether it is authorized to serve IP addresses.
- When installed in a multiple forest environment, DHCP servers seek authorization from within. Once authorized, DHCP servers in a multiple forest environment lease IP addresses to all reachable clients.

7.2 DHCP Server AD DS Path and Objects

A domain joined DHCP server is authorized in the Active Directory Domain Services (AD DS). The "DhcpRoot" object and <DHCP server name> objects, which are of type "dhcpClass" [MS-ADSC] are added in the AD DS. The attribute "dhcpServers" and other mandatory attributes of the class "dhcpClass" are also updated in the AD DS. The section below describes the "dHCPClass" objects, their attribute values in different conditions, and the containers in AD DS.

7.3 Active Directory Path for dhcpClass Objects

The **ADsPath** where the dHCPClass [MS-ADSC] objects are stored is:

"LDAP://<domain name>/CN=NetServices, CN=Services, CN=Configuration [,DC=<domain component1> [,DC=<domain component2>] …]"

Format of the "dhcpServers" attribute of "dHCPClass"

The "dhcpServers" [MS-ADA1] attribute of the "dHCPClass" object MUST be updated with the value defined below.

"i<server ip address>$rcn=<relative ADsPath Name>$f<flags>$s<server name>$"

The following table provides the specifics of the string.

Copyright © 2015 Microsoft Corporation

Field	Description	Examples
server ip address	IPv4 address of the DHCP server which is being authorized	"57.60.41.211"
relative ADsPath Name	Relative LDAP Path name of the object in which the attribute "dhcpServer" is being updated.	"dhcpserver.contoso.com"
Flags	Unused field. This is set to 0.	0x00000000
server name	Server name of the DHCP server which is being authorized.	"dhcpserver.contoso.com"

The server ip address field takes an IPv4 address only. In an IPv6 scenario where the IPv4 is uninstalled or disabled, the DHCP server adds itself to the AD DS with the server name of the DHCPv6 server and a fixed IP address of 255.0.0.1.

The following table specifies the characters in the "dhcpServers" attribute value string.

Value	Meaning
L'$'	A field separator.
L'i'	Precedes an IP address.
L'r'	Precedes a relative Active Directory Path.
L'f'	Precedes a flag entry
L's'	Precedes a server entry

7.4 Mandatory Attribute Values for the DHCPRoot Object

The mandatory attributes of the "dHCPClass" class SHOULD be updated with values mentioned below when creating a "DhcpRoot" object.

OBJECT ATTRIBUTES	Value
dhcpUniqueKey	0
dhcpType	0
dhcpIdentification	L"This is a server"
dhcpFlags	0
instanceType	0x04

7.5 Mandatory Attribute Values for the <DHCP server> Object

The mandatory attributes of the "dHCPClass" class SHOULD be updated with values mentioned below when creating a <Dhcp server> object.

OBJECT ATTRIBUTES	Value
dhcpUniqueKey	0

Copyright © 2015 Microsoft Corporation

OBJECT ATTRIBUTES	Value
dhcpType	1
dhcpIdentification	L"DHCP Server Object"
dhcpFlags	0
instanceType	0x04

7.6 Unauthorization Filter

To unauthorize a DHCP server, the server object added in AD DS MUST be removed. The server object to be deleted is identified by the "dhcpServers" attribute value. The filter [MS-ADTS] required to identify the server object corresponding to the specific DHCP server is described below.

"(&(objectCategory=dHCPClass)(&(dhcpServers=i<server ip address>$*)(dhcpServers=*s<server name>$*)))"

7.7 Validation Filter

The filter [MS-ADTS] required to validate DHCP server authorization in AD DS is described below.

"(&(objectCategory=dHCPClass)(|(dhcpServers=i<server ip address>$*)(dhcpServers=*s<server name>$*)))"

7.8 Authorizing a DHCP Server in Active Directory Domain Services

A DHCP server that is domain joined is authorized by a domain administrator in the AD DS.

The authorization MUST first check to see if a "CN=DhcpRoot" object is present in the AD DS in the ADsPath.

If it is not found it MUST be created in the AD DS using the following:

- Object Relative Distinguished Name: CN= "DhcpRoot"
- Object Class: "dHCPClass" (defined in the AD schema [MS-ADSC])

When creating "DhcpRoot" object, the "dHCPClass" attributes SHOULD be updated.

Once the object "DhcpRoot" exists, a new object by the name of the DHCP server authorizing itself in AD DS MUST be created.

The LDAP ADsPath of the new object MUST be specified using the following:

- Object Distinguished Name = <server name>
- Object Class = "dHCPClass"

When creating DHCP server object to authorize in AD DS, the "dHCPClass" attributes SHOULD be updated.

The new server object attribute "dhcpServers" MUST be updated.

Copyright © 2015 Microsoft Corporation

7.9 Unauthorizing a DHCP Server from Active Directory Domain Services

A DHCP server is unauthorized from AD DS when the "dHCPClass" object corresponding to the server is deleted from the AD DS.

The filter required to unauthorize a DHCP server MUST be based on its IP address and server name as specified above under section Unauthorization Filter.

This filter matches a "dHCPClass" object with the "dhcpServers" attribute matching "i<server ip address>$" and "s<server name>$". Only one such object matches the filter. Delete that object from the AD DS.

7.10 Validating DHCP Server Authorization in Active Directory Domain Services

A domain joined DHCP server verifies if it is authorized to service DHCP clients. It validates itself in AD DS.

A DHCP server MUST verify if the "DhcpRoot" object exists in the AD DS. If the object is not present, the server MUST work as a non-authorized DHCP server. A DHCP server MUST make a filter query to verify if it is authorized in the AD DS. If the query succeeds, the DHCP server is authorized, otherwise it is not.

The filter required to validate if a DHCP server is authorized MUST be based on a server IP address or server name.

This filter MUST match any "dHCPClass" objects with "dhcpServers" attribute matching "i<server IP address>$" or "s<server name>$".

Copyright © 2015 Microsoft Corporation

8 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

Copyright © 2015 Microsoft Corporation

9 Index

Copyright © 2015 Microsoft Corporation

Copyright © 2015 Microsoft Corporation

T

U

V

W

Copyright © 2015 Microsoft Corporation