

Internet Engineering Task Force (IETF)
Request for Comments: 5970
Category: Standards Track
ISSN: 2070-1721

T. Huth
J. Freimann
IBM Germany R&D GmbH
V. Zimmer
Intel
D. Thaler
Microsoft
September 2010

DHCPv6 Options for Network Boot

Abstract

The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) provides a framework for passing configuration information to nodes on a network. This document describes new options for DHCPv6 that SHOULD be used for booting a node from the network.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in [Section 2 of RFC 5741](#).

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <http://www.rfc-editor.org/info/rfc5970>.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Conventions	3
3. Options	3
3.1. Boot File Uniform Resource Locator (URL) Option	3
3.2. Boot File Parameters Option	4
3.3. Client System Architecture Type Option	5
3.4. Client Network Interface Identifier Option	6
4. Appearance of the Options	7
5. Download Protocol Considerations	7
6. IANA Considerations	7
7. Security Considerations	8
8. Acknowledgements	8
9. References	9
9.1. Normative References	9
9.2. Informative References	9

1. Introduction

This document describes DHCPv6 options that SHOULD be used to provide configuration information for a node that must be booted using the network rather than from local storage.

Network booting is used, for example, in some environments where administrators have to maintain a large number of nodes. By serving all boot and configuration files from a central server, the effort required to maintain these nodes is greatly reduced.

A typical boot file would be, for example, an operating system kernel or a boot-loader program. To be able to execute such a file, the firmware running on the client node must perform the following two steps (see Figure 1): First get all information that is required for downloading and executing the boot file. Second, download the boot file and execute it.

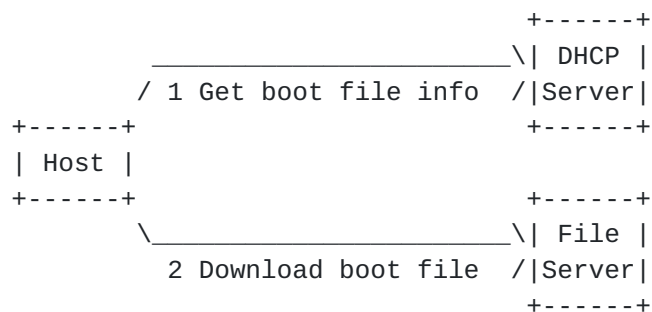


Figure 1: Network Boot Sequence

The information that is required for booting over the network MUST include at least the details about the server on which the boot files can be found, the protocol to be used for the download (for example, HTTP [RFC2616] or TFTP [RFC1350]), and the path and name of the boot file on the server. Additionally, the server and client MAY exchange information about the parameters that should be passed to the OS kernel or boot-loader program, respectively, or information about the supported boot environment.

DHCPv6 allows client nodes to ask a DHCPv6 server for configuration parameters. This document provides new options that a client can request from the DHCPv6 server to satisfy its requirements for booting. It also introduces a new IANA registry for processor architecture types that are used by the OPTION_CLIENT_ARCH_TYPE option (see [Section 3.3](#)).

2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#) [RFC2119].

Terminology specific to IPv6 and DHCPv6 are used in the same way as is defined in the "Terminology" sections of [RFC3315].

3. Options

Option formats comply with DHCPv6 options per [RFC3315] ([Section 6](#)). The boot-file-url option (see [Section 3.1](#)) is mandatory for booting, all other options are optional.

3.1. Boot File Uniform Resource Locator (URL) Option

The server sends this option to inform the client about a URL to a boot file.

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           OPT_BOOTFILE_URL           |           option-len       |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     |                               |
|           boot-file-url (variable length)           |             |
|                                     |                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```


Format description:

option-code OPT_BOOTFILE_URL (59).

option-len Length of the boot-file-url in octets.

boot-file-url This string is the URL for the boot file. It MUST comply with STD 66 [RFC3986]. The string is not NUL-terminated.

If the host in the URL is expressed using an IPv6 address rather than a domain name, the address in the URL then MUST be enclosed in "[" and "]" characters, conforming to [RFC3986]. Clients that have DNS implementations SHOULD support the use of domain names in the URL.

3.2. Boot File Parameters Option

This option is sent by the server to the client. It consists of multiple UTF-8 ([RFC3629]) strings. They are used to specify parameters for the boot file (similar to the command line arguments in most modern operating systems). For example, these parameters could be used to specify the root file system of the OS kernel, or the location from which a second-stage boot-loader program can download its configuration file.

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           OPT_BOOTFILE_PARAM           |           option-len       |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| param-len 1                             |                             |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
.                                         parameter 1                  .
.                                         (variable length)           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
.                                         .
.                                         <multiple Parameters>    .
.                                         .
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| param-len n                             |                             |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
.                                         parameter n                  .
.                                         (variable length)           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```


Format description:

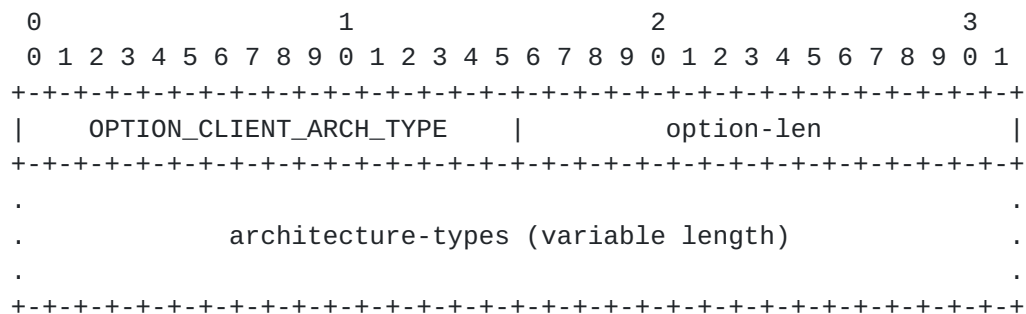
option-code	OPT_BOOTFILE_PARAM (60).
option-len	Length of the Boot File Parameters option in octets (not including the size of the option-code and option-len fields).
param-len 1...n	This is a 16-bit integer that specifies the length of the following parameter in octets (not including the parameter-length field).
parameter 1...n	These UTF-8 strings are parameters needed for booting, e.g., kernel parameters. The strings are not NUL-terminated.

When the boot firmware executes the boot file that has been specified in the OPT_BOOTFILE_URL option, it MUST pass these parameters, if present, in the order that they appear in the OPT_BOOTFILE_PARAM option.

3.3. Client System Architecture Type Option

This option provides parity with the Client System Architecture Type option defined for DHCPv4 in [Section 2.1 of \[RFC4578\]](#).

The format of the option is:



option-code	OPTION_CLIENT_ARCH_TYPE (61).
option-len	Length of the "architecture-types" field in octets. It MUST be an even number greater than zero. See Section 2.1 of [RFC4578] for details.
architecture-types	A list of one or more architecture types, as specified in Section 2.1 of [RFC4578] . Each architecture type identifier in this list is a 16-bit value that describes the pre-boot runtime

environment of the client machine. A list of valid values is maintained by the IANA (see [Section 6](#)).

The client MAY use this option to send a list of supported architecture types to the server, so the server can decide which boot file should be provided to the client. If a client supports more than one pre-boot environment (for example, both 32-bit and 64-bit executables), the most preferred architecture type MUST be listed as first item, followed by the others with descending priority.

If the client used this option in the request, the server SHOULD include this option to inform the client about the pre-boot environments that are supported by the boot file. The list MUST only contain architecture types that have initially been queried by the client. The items MUST also be listed in order of descending priority.

3.4. Client Network Interface Identifier Option

If the client supports the Universal Network Device Interface (UNDI) (see [\[PXE21\]](#) and [\[UEFI23\]](#)), it may send the Client Network Interface Identifier option to a DHCP server to provide information about its level of UNDI support.

This option provides parity with the Client Network Interface Identifier option defined for DHCPv4 in [Section 2.2 of \[RFC4578\]](#).

The format of the option is:

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|               OPTION_NII               |               option-len   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      Type      |      Major      |      Minor      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

option-code OPTION_NII (62).

option-len 3

Type As specified in [Section 2.2 of \[RFC4578\]](#).

Major As specified in [Section 2.2 of \[RFC4578\]](#).

Minor As specified in [Section 2.2 of \[RFC4578\]](#).

The list of valid Type, Major, and Minor values is maintained in the Unified Extensible Firmware Interface specification [[UEFI23](#)].

4. Appearance of the Options

These options MUST NOT appear in DHCPv6 messages other than the types Solicit, Advertise, Request, Renew, Rebind, Information-Request, and Reply.

The option-codes of these options MAY appear in the Option Request option in the DHCPv6 message types Solicit, Request, Renew, Rebind, Information-Request, and Reconfigure.

5. Download Protocol Considerations

The Boot File URL option does not place any constraints on the protocol used for downloading the boot file, other than that it MUST be possible to specify it in a URL. For the sake of administrative simplicity, we strongly recommend that, at a minimum, implementers of network boot loaders implement the well-known and established HyperText Transfer Protocol (HTTP) [[RFC2616](#)] for downloading. Please note that for IPv6, this supersedes [[RFC906](#)], which recommended using TFTP for downloading (see [[RFC3617](#)] for the 'tftp' URL definition).

When using the Internet Small Computer System Interface (iSCSI) for booting, the 'iscsi' URI is formed as defined in [[RFC4173](#)]. The functionality attributed in [RFC 4173](#) to a root path option is provided for IPv6 by the Boot File URL option instead.

6. IANA Considerations

The following options have been assigned by the IANA from the option number space defined in [Section 24](#) of the DHCPv6 RFC [[RFC3315](#)].

Option name	Value	Specified in
OPT_BOOTFILE_URL	59	Section 3.1
OPT_BOOTFILE_PARAM	60	Section 3.2
OPTION_CLIENT_ARCH_TYPE	61	Section 3.3
OPTION_NII	62	Section 3.4

This document also introduces a new IANA registry for processor architecture types. The name of this registry is "Processor Architecture Types". Registry entries consist of a 16-bit integer recorded in decimal format and a descriptive name. The initial values of this registry can be found in [[RFC4578](#)], [Section 2.1](#).

The assignment policy for values is through Expert Review (see [\[RFC5226\]](#)), and any requests for values must supply the descriptive name for the processor architecture type.

7. Security Considerations

In untrusted networks, a rogue DHCPv6 server could send the new DHCPv6 options described in this document. The booting clients could then be provided with a wrong URL so that either the boot fails or, even worse, the client boots the wrong operating system that has been provided by a malicious file server. To prevent this kind of attack, clients SHOULD use authentication of DHCPv6 messages (see [Section 21 in \[RFC3315\]](#)).

Note also that DHCPv6 messages are sent unencrypted by default. So the boot file URL options are sent unencrypted over the network, too. This can become a security risk since the URLs can contain sensitive information like user names and passwords (for example, a URL like "ftp://username:password@servername/path/file"). At the current point in time, there is no possibility to send encrypted DHCPv6 messages, so it is strongly RECOMMENDED not to use sensitive information in the URLs in untrusted networks (using passwords in URLs is deprecated anyway, according to [\[RFC3986\]](#)).

Even if the DHCPv6 transaction is secured, this does not protect against attacks on the boot file download channel. Consequently, we recommend that either (a) implementers use protocols like HTTPS [\[RFC2818\]](#) or Transport Layer Security (TLS) within HTTP [\[RFC2817\]](#) to prevent spoofing or (b) the boot-loader software implement a mechanism for signing boot images and a configurable signing key. The latter is done so that if a malicious image is provided, it can be detected and rejected.

8. Acknowledgements

The authors would like to thank Ruth Li, Dong Wei, Kathryn Hampton, Phil Dorah, Richard Chan, and Fiona Jensen for discussions that led to this document.

The authors would also like to thank Ketan P. Pancholi, Alfred Hoenes, Gabriel Montenegro, and Ted Lemon for corrections and suggestions.

9. References

9.1. Normative References

- [PXE21] Johnston, M., "Preboot Execution Environment (PXE) Specification", September 1999, <<http://www.pix.net/software/pxeboot/archive/pxespec.pdf>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", [RFC 3315](#), July 2003.
- [RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, [RFC 3629](#), November 2003.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, [RFC 3986](#), January 2005.
- [RFC4173] Sarkar, P., Missimer, D., and C. Sapuntzakis, "Bootstrapping Clients using the Internet Small Computer System Interface (iSCSI) Protocol", [RFC 4173](#), September 2005.
- [RFC4578] Johnston, M. and S. Venaas, "Dynamic Host Configuration Protocol (DHCP) Options for the Intel Preboot eXecution Environment (PXE)", [RFC 4578](#), November 2006.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", [BCP 26](#), [RFC 5226](#), May 2008.
- [UEFI23] UEFI Forum, "Unified Extensible Firmware Interface Specification, Version 2.3", May 2009, <<http://www.uefi.org/>>.

9.2. Informative References

- [RFC906] Finlayson, R., "Bootstrap Loading using TFTP", [RFC 906](#), June 1984.
- [RFC1350] Sollins, K., "The TFTP Protocol (Revision 2)", STD 33, [RFC 1350](#), July 1992.

- [RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", [RFC 2616](#), June 1999.
- [RFC2817] Khare, R. and S. Lawrence, "Upgrading to TLS Within HTTP/1.1", [RFC 2817](#), May 2000.
- [RFC2818] Rescorla, E., "HTTP Over TLS", [RFC 2818](#), May 2000.
- [RFC3617] Lear, E., "Uniform Resource Identifier (URI) Scheme and Applicability Statement for the Trivial File Transfer Protocol (TFTP)", [RFC 3617](#), October 2003.

Authors' Addresses

Thomas H. Huth
IBM Germany Research & Development GmbH
Schoenaicher Strasse 220
Boeblingen 71032
Germany

Phone: +49-7031-16-2183
EMail: thuth@de.ibm.com

Jens T. Freimann
IBM Germany Research & Development GmbH
Schoenaicher Strasse 220
Boeblingen 71032
Germany

Phone: +49-7031-16-1122
EMail: jfrei@de.ibm.com

Vincent Zimmer
Intel
2800 Center Drive
DuPont WA 98327
USA

Phone: +1 253 371 5667
EMail: vincent.zimmer@intel.com

Dave Thaler
Microsoft
One Microsoft Way
Redmond WA 98052
USA

Phone: +1 425 703-8835
EMail: dthaler@microsoft.com

