DRAFT FOR REAFFIRMATION REVIEW

E1.37-2 – 2015 (R2021)

Entertainment Technology—Additional Message Sets for ANSI E1.20 (RDM) – Part 2, IPv4 & DNS Configuration Messages

Document number CP/2012-1001r8

Approved by ANSI's Board of Standards Review on _____

NOTICE and DISCLAIMER

ESTA does not approve, inspect, or certify any installations, procedures, equipment or materials for compliance with codes, recommended practices or standards. Compliance with a ESTA standard or an American National Standard developed by ESTA is the sole and exclusive responsibility of the manufacturer or provider and is entirely within their control and discretion. Any markings, identification or other claims of compliance do not constitute certification or approval of any type or nature whatsoever by ESTA.

ESTA neither guarantees nor warrants the accuracy or completeness of any information published herein and disclaims liability for any personal injury, property or other damage or injury of any nature whatsoever, whether special, indirect, consequential or compensatory, directly or indirectly resulting from the publication, use of, or reliance on this document. In issuing and distributing this document.

In issuing this document, ESTA does not either (a) undertake to render professional or other services for or on behalf of any person or entity, or (b) undertake any duty to any person or entity with respect to this document or its contents. Anyone using this document should rely on his or her own independent judgment or, as appropriate, seek the advice of a competent professional in determining the exercise of reasonable care in any given circumstance.

Published by:
Entertainment Services and Technology Association
P.O. Box 23200,
Brooklyn, NY 11202-3200
USA
Phone: 1-212-244-1505
Fax: 1-212-244-1502
standards@esta.org

ESTA's Technical Standards Program

ESTA's Technical Standards Program was created to serve the ESTA membership and the entertainment industry in technical standards-related matters. The goal of the program is to take a leading role regarding technology and safety within the entertainment industry by creating recommended practices and standards, monitoring standards issues around the world on benefit of our members, and improving communications and safety within the industry. In its technical standards development activities, ESTA works closely with other industry organizations, including ESA, CITT, USITT and VPLT, in addition to representing the interests of ESTA members to ANSI, UL, ASCE, ICC, and NFPA. ESTA is an ANSI Accredited Standards Developer.

The Technical Standards Council (TSC) established by ESTA's Board of Directors to oversee and coordinate the Technical Standards Program. Made up of individuals experienced in standards development work from throughout our industry, the Committee approves all projects undertaken and assigns them to the appropriate working group. The Technical Standards Council employs a Technical Standards Manager to coordinate the work of the Committee and its working groups as well as maintaining a "Standards Watch" on behalf of members. Working groups include: Control Protocols, Electrical Power, Event Safety, Floors, Fog and Smoke, Followspot Positions, Mental Health & Well-being Management, Photometrics, Rigging, Stage Machinery, and Prop Weapons Safety.

ESTA encourages active participation in the Technical Standards Program. There are several ways to become involved. The easiest way to actively participate is to respond to any of the public reviews advertised on ESTA's public review web page. The next level of participation requires completion of an application to become a working group member; applications are available from the TSP's procedural documents web page. Application as an Observer with non-voting status affords access to updates on standards development documents. Application as a voting participant affords full access to a consensus voice that helps shape the industry. Voting status carries responsibilities of responding to letter ballots and attending meetings, but membership in ESTA or any other organization is not a requirement for participation in the TSP. One can also become involved by requesting that the TSC develop a standard or a recommended practice in an area of concern to them.

The Control Protocols Working Group, which authored this standard, consists of a cross section of entertainment industry professionals representing a diversity of interests. ESTA is committed to developing consensus-based standards and recommended practices in an open setting.

IInvestors in Innovation

The Technical Standard Program (TSP) is financially supported by ESTA and by companies and individuals who make undirected donations to the TSP. Contributing companies and individuals who have helped fund the TSP are recognized as “Investors in Innovation.” The Investors in Innovation when this standard was approved by ANSI’s Board of Standards Review, are as follows:

[This is a placeholder for the table]

Memorial donor: The Estate of Ken Vannice

Contact Information

Technical Standards Manager

Richard J. Nix
ESTA
271 Cadman Plaza PO Box 23200
New York, NY 11202-3200
USA
+1-212-244-1505
richard.nix@esta.org

Technical Standards Council Co-chairpersons

Alan Rowe
I.A.T.S.E Local 728
+1-310-702-2909
amrowe@iatse728.org

Dan Culhane
Wenger Corp
+1-612-868-4769
culhane.dan@gmail.com

Control Protocol Working Group Co-chairpersons

Javid Butler
Goddard Design LLC
+1-702-759-2427
javid@goddard.design

Maya Nigrosh
mnigrosh@alumni.cmu.edu

Acknowledgments

The Control Protocols Working Group members when the motion was made in July 2020 to reaffirm the 2015 edition of this document are shown below.

Voting members:

Observer members:

Interest category codes:
CP = Custom-market Producer
DE = Designer
DR = Dealer or Rental company
G = General interest
MP = Mass-market Producer
U = User

Table of Contents

Index of Tables

1 Introduction

1.1 E1.20 Basic Features

The ANSI E1.20 Remote Device Management Protocol (RDM) [RDM] permits intelligent bi-directional communication between devices from multiple manufacturers using a modified DMX512 data link. RDM is an EF 1.0 implementation of ANSI E1.11 (DMX512-A).

RDM permits a console or other controlling device to discover and then configure, monitor, and manage intermediate and end-devices connected through a DMX512 network. RDM provides intelligent control of devices on a DMX512 network.

RDM commands may also be sent over transports other than DMX512-A., for example, IPv4. The specification for sending RDM commands over non-DMX512-A transports is outside the scope of this standard.

1.2 Overview & Scope

This document provides additional Get/Set parameter messages (PIDs) for use with the ANSI E1.20 Remote Device Management protocol. Messages in this document are intended for configuring network interfaces, routing information and Domain Name System [DNS] settings on devices with IPv4 [IPv4] addresses.

The messages in this standard enable a controller to retrieve a list of network interfaces on a device, and for each interface, do the following:

- If appropriate, determine the EUI-48 hardware address [EUI]. Such addresses are commonly referred to as MAC addresses. Hardware addresses in a format other than EUI-48 are not covered by this standard.
- Retrieve and configure the IPv4 address & subnet mask. Addressing schemes other than IPv4 are outside the scope of this standard.
- Enable / disable DHCP [DHCPv4].
- Renew and release the DHCP lease.
- Enable and disable Zeroconf [IPv4LL].

Commands are also provided to:

- Set the hostname of the device.
- Set the DNS domain of the device.
- Set the DNS name servers of the device.
- Set the default IPv4 route of the device.

2 Normative References

[ARP-PARAMS] - Address Resolution Protocol (ARP) Parameters
[http://www.iana.org/assignments/arp-parameters/arp-parameters.xhtml]

This registry is maintained by:

Internet Engineering Task Force (IETF) Secretariat
c/o Association Management Solutions, LLC (AMS)
48377 Fremont Blvd., Suite 117
Fremont, California 94538
USA
+1-510-492-4080
http://www.ietf.org

[DHCPv4] RFC 2131 - Dynamic Host Configuration Protocol. 1997. [http://tools.ietf.org/html/rfc2131]

This standard is maintained by the IETF.

[DNS] RFC 1035 - Domain Names – Implementation and Specification. 1987.
[http://tools.ietf.org/html/rfc1035].

This standard is maintained by the IETF. This standard has been updated, please see the link for the relevant updates.

[DNS-CLARIFICATIONS] RFC 2181 - Clarifications to the DNS Specification. 1997.
[http://tools.ietf.org/html/rfc2181]

This standard is maintained by the IETF.

[EUI] Guidelines for use of a 48-bit Extended Unique Identifier (EUI-48™) [http://standards.ieee.org/develop/regauth/tut/eui48.pdf]

This document is maintained by:

IEEE Operations Center
445 Hoes Lane
Piscataway, NJ 08854-4141
USA
+1-732-981-0060
http://standards.ieee.org/

[HOST—NAME] RFC 1123 - Requirements for Internet Hosts -Application and Support. 1989. [http://tools.ietf.org/html/rfc1123]

This standard is maintained by the IETF.

[IDNA] RFC 3490 - Internationalizing Domain Names in Applications. 2003. [http://tools.ietf.org/html/rfc3490]

This standard is maintained by the IETF.

[IPv4] RFC 791– Internet Protocol. 1999. [http://tools.ietf.org/html/rfc791]

This standard is maintained by the IETF. This standard is commonly referred to as Internet Protocol Version 4.

[IPv4LL] RFC 3927– Dynamic Configuration of IPv4 Link-Local Addresses. 2005. [http://tools.ietf.org/html/rfc3927]

This standard is maintained by the IETF.

[RDM] ANSI E1.20 Entertainment Technology – Remote Device Management over DMX512 networks.

This standard is maintained by ESTA.

3 General

3.1 General

These parameter messages are intended for configuring IPv4 network interfaces, the default IPv4 route, the Domain Name System [DNS] servers, the device's hostname, and the device's domain name.

They are for general purpose use across any type of device and not limited to any specific class of products.

3.2 Sub-Device Handling

Refer to ANSI E1.20 Section 9 for information on Sub-Device usage. This document does not change or modify the requirements stated in ANSI E1.20. Requirements stated in this document are in addition to the stated ANSI E1.20 requirements.

3.3 Text Field Handling

Text field encoding shall conform to Section 10.1 in [RDM]. Where the parameter data contains both a text field and one or more fixed length fields, the parameter data length shall be the sum of the size of the fixed length fields as well as the actual size of the text string being sent.

3.4 Byte Ordering

All multi-byte data shall be transmitted as specified in Section 6.1 of [RDM]. This includes IPv4 Addresses.

3.5 IP Addressing Modes

Three IPv4 address configuration modes are provided: static configuration, DHCP [DHCPv4] and Zeroconf [IPv4LL]. The IP Addresses used for an interface shall be determined by:

- If a static address is configured, it shall be used.
- If no static address is configured, and DHCP mode is enabled, the device shall attempt to acquire an address for the interface using DHCP.
- If no static address is configured, and DHCP failed to provide an address (or was disabled / unsupported) a link local address shall be used if Zeroconf mode is enabled.
- If none of the above provided an address, the interface shall remain unconfigured.

If DHCP mode is enabled and no address was obtained using DHCP, the device shall retry the DHCP process again at a later time. Devices should retry no more than once every 5 minutes.

Per Section 4.4.5 of [DHCPv4], if a DHCP lease expires before the device receives a positive acknowledgment of renewal (DHCPACK) it must stop using the previous address. If Zeroconf mode is enabled the device shall switch to a link local address, and retry DHCP requests as described above.

4 Parameter Messages

4.1 Get Interface List (LIST_INTERFACES)

This parameter returns a packed list of network interface descriptors, representing the IPv4 network interfaces on the device.

Controller: (GET)

(Port Id) 0x01 - 0xFF	(Message Count) 0x00	(Sub-Device) 0x0000 (Root) or 0x0001-0x0200
(CC) GET_COMMAND	(PID) LIST_INTERFACES	(PDL) 0x00
(PD) Not Present		

Response:

(Response Type) ACK	(Message Count) 0x00 - 0xFF	(Sub-Device) Copy of Controller SD
(CC) GET_COMMAND_RESPONSE	(PID) LIST_INTERFACES	Variable (0x00 – 0xE6)
(PD) Packed group of repeated interface descriptors (48 bits per interface) Interface Identifier (32-bit) Interface Hardware Type (16-bit)		

Data Description:

The data returned is a packed list of network interface descriptors. Each interface descriptor contains two fields, the *Interface Identifier* and the *Interface Hardware Type*.

The Interface Identifier is a 32-bit value which the controller will use to refer to the interface in other commands. The interface identifiers returned shall range from 1 to 0xFFFFFF00. Interface identifiers are not required to be contiguous.

The Interface Hardware Type field identifies the underlying hardware of the interface. The values are defined in [ARP-PARAMETERS] under the 'Hardware Types' section.

Devices may queue a GET LIST_INTERFACES response if the set of interfaces changes.

4.2 Get Interface Name (INTERFACE_LABEL)

This parameter is used to retrieve the label for a network interface.

Responders shall return NR_DATA_OUT_OF_RANGE if the interface identifier refers to an interface that would not be listed in the LIST_INTERFACES response.

Controller: (GET)

(Port Id) 0x01 - 0xFF	(Message Count) 0x00	(Sub-Device) 0x0000 (Root) or 0x0001-0x0200
(CC) GET_COMMAND	(PID) INTERFACE_LABEL	(PDL) 0x04
(PD) Interface Identifier (32-bit)		

Response:

(Response Type) ACK	(Message Count) 0x00 - 0xFF	(Sub-Device) Copy of Controller SD
(CC) GET_COMMAND_RESPONSE	(PID) INTERFACE_LABEL	Variable (0x04 – 0x24)
(PD) Interface Identifier (32-bit) ASCII text label. Up to 32 characters.		

Data Description:

The data returned is an ASCII string of up to 32 characters, conforming to Section 3.3.

4.3 Get Hardware Address (INTERFACE_HARDWARE_ADDRESS_TYPE1)

This parameter is used to fetch the EUI-48 [EUI] hardware address of an interface.

Responders shall return NR_DATA_OUT_OF_RANGE if the interface identifier refers to an interface that would not be listed in the LIST_INTERFACES response.

Requests for the hardware address of an interface that does not have an EUI-48 hardware address shall return NR_DATA_OUT_OF_RANGE.

Controller: (GET)

(Port Id) 0x01 - 0xFF	(Message Count) 0x00	(Sub-Device) 0x0000 (Root) or 0x0001-0x0200
(CC) GET_COMMAND	(PID) INTERFACE_HARDWARE_ADDRESS_TYPE1	(PDL) 0x04
(PD) Interface Identifier (32-bit)		

Response:

(Response Type) ACK	(Message Count) 0x00 - 0xFF	(Sub-Device) Copy of Controller SD
(CC) GET_COMMAND_RESPONSE	(PID) INTERFACE_HARDWARE_ADDRESS_TYPE1	(PDL) 0x0a
(PD) Interface Identifier (32-bit) Hardware Address (48-bit)		

Data Description:

The data returned is the EUI-48 hardware address, commonly known as the MAC address, in big endian format.

4.4 Get/Set DHCP (IPV4_DHCP_MODE)

This parameter is used to retrieve or change the Dynamic Host Configuration Protocol [DHCPv4] mode for an interface. Changes to the DHCP mode shall not take effect until INTERFACE_APPLY_CONFIGURATION SET request is received.

The GET command shall return the configured DHCP mode for the interface, which may not reflect the actual state. To retrieve the actual state controllers can use IPV4_CURRENT_ADDRESS.

Responders shall return NR_DATA_OUT_OF_RANGE if the interface identifier refers to an interface that would not be listed in the LIST_INTERFACES response.

If the interface does not support DHCP then GET requests shall always return a value of 0 (disabled) and SET requests shall return NR_ACTION_NOT_SUPPORTED.

As described in Section 3.5, DHCP mode shall only take effect when the configured IPv4 address (see Section 4.6) is set to IPV4_UNCONFIGURED and the DHCP mode option is enabled.

Controller: (GET)

(Port Id) 0x01 - 0xFF	(Message Count) 0x00	(Sub-Device) 0x0000 (Root) or 0x0001-0x0200
(CC) GET_COMMAND	(PID) IPV4_DHCP_MODE	(PDL) 0x04
(PD) Interface Identifier (32-bit)		

Response:

(Response Type) ACK	(Message Count) 0x00 - 0xFF	(Sub-Device) Copy of Controller SD
(CC) GET_COMMAND_RESPONSE	(PID) IPV4_DHCP_MODE	(PDL) 0x05
(PD) Interface Identifier (32-bit) Disabled/Enabled (0/1)		

Controller: (SET)

(Port Id) 0x01 - 0xFF	(Message Count) 0x00	(Sub-Device) 0x0000 (Root) or 0x0001-0x0200 or 0xFFFF
(CC) SET_COMMAND	(PID) IPV4_DHCP_MODE	(PDL) 0x05
(PD) Interface Identifier (32-bit) Disabled/Enabled (0/1)		

Response:

(Response Type) ACK	(Message Count) 0x00 - 0xFF	(Sub-Device) Copy of Controller SD
(CC) SET_COMMAND_RESPONSE	(PID) IPV4_DHCP_MODE	(PDL) 0x00
(PD) Not Present		

4.5 Get/Set Zeroconf Mode (IPV4_ZEROCONF_MODE)

This parameter is used to retrieve or change the IPv4 Link Local Addressing (Zeroconf) [IPv4LL] mode on an interface. Changes to the Zeroconf mode shall not take effect until an INTERFACE_APPLY_CONFIGURATION SET request is received.

Zeroconf allows IPv4 addresses to be assigned to an interface, without configuration servers (such as DHCP servers) or operator intervention.

When Zeroconf is enabled, the algorithm described in [IPv4LL] shall be used to assign an IPv4 address to an interface when all of the following conditions are true:

- No static address is configured.
- DHCP mode is disabled or no DHCP lease has been received.

Responders shall return NR_DATA_OUT_OF_RANGE if the interface identifier refers to an interface that would not be listed in the LIST_INTERFACES response.

If the interface does not support Zeroconf then GET requests shall always return a value of 0 (disabled) and SET requests shall return NR_ACTION_NOT_SUPPORTED.

Controller: (GET)

(Port Id) 0x01 - 0xFF	(Message Count) 0x00	(Sub-Device) 0x0000 (Root) or 0x0001-0x0200
(CC) GET_COMMAND	(PID) IPV4_ZEROCONF_MODE	(PDL) 0x04
(PD) Interface Identifier (32-bit)		

Response:

(Response Type) ACK	(Message Count) 0x00 - 0xFF	(Sub-Device) Copy of Controller SD
(CC) GET_COMMAND_RESPONSE	(PID) IPV4_ZEROCONF_MODE	(PDL) 0x05
(PD) Interface Identifier (32-bit) Disabled/Enabled (0/1)		

Controller: (SET)

(Port Id) 0x01 - 0xFF	(Message Count) 0x00	(Sub-Device) 0x0000 (Root) or 0x0001-0x0200 or 0xFFFF
(CC) SET_COMMAND	(PID) IPV4_ZEROCONF_MODE	(PDL) 0x05
(PD) Interface Identifier (32-bit) Disabled/Enabled (0/1)		

Response:

(Response Type) ACK	(Message Count) 0x00 - 0xFF	(Sub-Device) Copy of Controller SD
(CC) SET_COMMAND_RESPONSE	(PID) IPV4_ZEROCONF_MODE	(PDL) 0x00
(PD) Not Present		

4.6 Get IPv4 Address / Netmask (IPV4_CURRENT_ADDRESS)

This parameter is used to retrieve the current IPv4 Address and Netmask information for an interface.

Responders shall return NR_DATA_OUT_OF_RANGE if the interface identifier refers to an interface that would not be listed in the LIST_INTERFACES response.

Controller: (GET)

(Port Id) 0x01 - 0xFF	(Message Count) 0x00	(Sub-Device) 0x0000 (Root) or 0x0001-0x0200
(CC) GET_COMMAND	(PID) IPV4_CURRENT_ADDRESS	(PDL) 0x04
(PD) Interface Identifier (32-bit)		

Response:

(Response Type) ACK	(Message Count) 0x00 - 0xFF	(Sub-Device) Copy of Controller SD
(CC) GET_COMMAND_RESPONSE	(PID) IPV4_CURRENT_ADDRESS	(PDL) 0x0a
(PD) Interface Identifier (32-bit)		
IPv4 Address (32-bit)		
Netmask	DHCP Status (0/1/2)	

Data Description:

The data returned is the current IPv4 address (as determined in Section 3.5) and current network mask. Interfaces without a configured IPv4 Address shall return IPV4_UNCONFIGURED.

The netmask is the number of bits in the network portion of the address. The valid range is from 0-32.

The DHCP Status field values are enumerated in Table A-3. Some devices may not be able to determine if the current address was obtained via DHCP. If the device does not support this feature, the value of the DHCP Status field shall be DHCP_STATUS_UNKNOWN.

If the feature is supported:

- If the current address was obtained via DHCP the value shall be set to DHCP_STATUS_ACTIVE.
- If the current address was not obtained via DHCP (i.e. either statically assigned or a zeroconf address) the value shall be set to DHCP_STATUS_INACTIVE.

Interaction with DHCP and Zeroconf Mode

Devices under DHCP control shall respond to a GET IPV4_CURRENT_ADDRESS with the current IPv4 address assigned by the DHCP server or IPV4_UNCONFIGURED if no address has been assigned and Zeroconf mode is disabled.

Devices may queue a GET IPV4_CURRENT_ADDRESS response when the address changes either via DHCP or any other mechanism.

4.7 Get/Set IPv4 Static Address (IPV4_STATIC_ADDRESS)

This parameter is used to statically configure the IPv4 address and network mask on an interface. Changes made to an interface's configuration shall not take effect until a SET INTERFACE_APPLY_CONFIGURATION request is received.

Responders shall return NR_DATA_OUT_OF_RANGE if the interface identifier refers to an interface that would not be listed in the LIST_INTERFACES response.

Controller: (GET)

(Port Id) 0x01 - 0xFF	(Message Count) 0x00	(Sub-Device) 0x0000 (Root) or 0x0001-0x0200
(CC) GET_COMMAND	(PID) IPV4_STATIC_ADDRESS	(PDL) 0x04
(PD) Interface Identifier (32-bit)		

Response:

(Response Type) ACK	(Message Count) 0x00 - 0xFF	(Sub-Device) Copy of Controller SD
(CC) GET_COMMAND_RESPONSE	(PID) IPV4_STATIC_ADDRESS	(PDL) 0x09
(PD) Interface Identifier (32-bit) IPv4 Address (32-bit) Netmask		

Controller: (SET)

(Port Id) 0x01 - 0xFF	(Message Count) 0x00	(Sub-Device) 0x0000 (Root) or 0x0001-0x0200 or 0xFFFF
(CC) SET_COMMAND	(PID) IPV4_STATIC_ADDRESS	(PDL) 0x09
(PD) Interface Identifier (32-bit) IPv4 Address (32-bit) Netmask		

Response:

(Response Type) ACK	(Message Count) 0x00 - 0xFF	(Sub-Device) Copy of Controller SD
(CC) SET_COMMAND_RESPONSE	(PID) IPV4_STATIC_ADDRESS	(PDL) 0x00
(PD) Not Present		

Data Description:

The netmask is the number of bits in the network portion of the address. The valid range is from 0-32.

4.8 Apply Interface Configuration (INTERFACE_APPLY_CONFIGURATION)

This parameter applies the stored configuration to an interface. The configuration of an interface shall include, but is not limited to, the settings in IPV4_STATIC_ADDRESS, IPV4_DHCP_MODE (if supported) and IPV4_ZEROCONF_MODE (if supported).

Responders shall return NR_DATA_OUT_OF_RANGE if the interface identifier refers to an interface that would not be listed in the LIST_INTERFACES response.

Controllers should be aware that some devices may reboot after applying the interface configuration.

Controller: (SET)

(Port Id) 0x01 - 0xFF	(Message Count) 0x00	(Sub-Device) 0x0000 (Root) or 0x0001-0x0200 or 0xFFFF
(CC) SET_COMMAND	(PID) INTERFACE_APPLY_CONFIGURATION	(PDL) 0x04
(PD) Interface Identifier (32-bit)		

Response:

(Response Type) ACK	(Message Count) 0x00 - 0xFF	(Sub-Device) Copy of Controller SD
(CC) SET_COMMAND_RESPONSE	(PID) INTERFACE_APPLY_CONFIGURATION	(PDL) 0x00
(PD) Not Present		

Response Behavior

Controllers are advised that if the underlying transport relies on IPv4 connectivity using the interface specified in the request, no response may be received.

Following a SET INTERFACE_APPLY_CONFIGURATION request, controllers may send a GET IPV4_CURRENT_ADDRESS to read the new address.

4.9 Renew DHCP Lease (INTERFACE_RENEW_DHCP)

This parameter causes the device to attempt to renew its DHCP address.

Responders shall return NR_DATA_OUT_OF_RANGE if the interface identifier refers to an interface that would not be listed in the LIST_INTERFACES response.

Upon receiving a SET INTERFACE_RENEW_DHCP, the device shall do one of the following:

- If the specified interface has a static address configured, NR_ACTION_NOT_SUPPORTED shall be returned.
- If no static address is configured, DHCP mode is enabled, and the interface is currently configured with an IPv4 address, the interface shall enter the RENEWING state as described in Section 4.4.5 of [DHCPv4].
- If no static address is configured, DHCP mode is enabled, and the interface does not have a DHCP-assigned IPv4 address, the interface shall enter the INIT state as described in Section 4.4.1 of [DHCPv4].
- If no static address is configured and DHCP mode was not enabled, NR_ACTION_NOT_SUPPORTED shall be returned.

Controller: (SET)

(Port Id) 0x01 - 0xFF	(Message Count) 0x00	(Sub-Device) 0x0000 (Root) or 0x0001-0x0200 or 0xFFFF
(CC) SET_COMMAND	(PID) INTERFACE_RENEW_DHCP	(PDL) 0x04
(PD) Interface Identifier (32-bit)		

Response:

(Response Type) ACK	(Message Count) 0x00 - 0xFF	(Sub-Device) Copy of Controller SD
(CC) SET_COMMAND_RESPONSE	(PID) INTERFACE_RENEW_DHCP	(PDL) 0x00
(PD) Not Present		

Response Behavior

Controllers are advised that if the underlying transport relies on IPv4 connectivity using the interface specified in the request, no response may be sent.

Following a SET INTERFACE_RENEW_DHCP request, controllers may send a GET IPV4_CURRENT_ADDRESS to read the new address.

4.10 Release DHCP Lease (INTERFACE_RELEASE_DHCP)

This parameter causes the device to release its DHCP address.

Responders shall return NR_DATA_OUT_OF_RANGE if the interface identifier refers to an interface that would not be listed in the LIST_INTERFACES response.

Upon receiving a SET INTERFACE_RELEASE_DHCP, the device shall do one of the following:

- If no static address is configured, DHCP mode is enabled, and the interface is currently configured with an IPv4 address, the device shall send a DHCPRELEASE message in accordance with item 6 of section 3.1 in [DHCPv4] and remove the address from the interface. If Zeroconf mode is enabled the device shall generate a link local address.
- Otherwise NR_ACTION_NOT_SUPPORTED shall be returned.

Controller: (SET)

(Port Id) 0x01 - 0xFF	(Message Count) 0x00	(Sub-Device) 0x0000 (Root) or 0x0001-0x0200 or 0xFFFF

(CC) SET_COMMAND	(PID) INTERFACE_RELEASE_DHCP	(PDL) 0x04
(PD) Interface Identifier (32-bit)		

Response:

(Response Type) ACK	(Message Count) 0x00 - 0xFF	(Sub-Device) Copy of Controller SD
(CC) SET_COMMAND_RESPONSE	(PID) INTERFACE_RELEASE_DHCP	(PDL) 0x00
(PD) Not Present		

Response Behavior

Controllers are advised that if the underlying transport relies on IPv4 connectivity using the interface specified in the request, no response may be sent.

4.11 Get/Set Default Route (IPV4_DEFAULT_ROUTE)

This parameter is used to get and set the default IPv4 route for a device. The default route is a global property of a device.

If a default route exists, it may point to a gateway or, in the case of a point-to-point link, to an interface without a gateway. If a gateway is used, the device may be able to provide the interface the gateway is reachable on. This is extra information which may facilitate debugging, but devices are not required to provide it.

Controllers should be aware that some devices may reboot after changing the default route.

Controller: (GET)

(Port Id) 0x01 - 0xFF	(Message Count) 0x00	(Sub-Device) 0x0000 (Root) or 0x0001-0x0200
(CC) GET_COMMAND	(PID) IPV4_DEFAULT_ROUTE	(PDL) 0x00
(PD) Not Present		

Response:

(Response Type) ACK	(Message Count) 0x00 - 0xFF	(Sub-Device) Copy of Controller SD
(CC) GET_COMMAND_RESPONSE	(PID) IPV4_DEFAULT_ROUTE	(PDL) 0x08
(PD) Interface Identifier (32-bit) IPv4 Default Route (32-bit)		

Controller: (SET)

(Port Id) 0x01 - 0xFF	(Message Count) 0x00	(Sub-Device) 0x0000 (Root) or 0x0001-0x0200 or 0xFFFF
(CC) SET_COMMAND	(PID) IPV4_DEFAULT_ROUTE	(PDL) 0x08
(PD) Interface Identifier (32-bit) IPv4 Default Route (32-bit)		

Response:

(Response Type) ACK	(Message Count) 0x00 - 0xFF	(Sub-Device) Copy of Controller SD
(CC) SET_COMMAND_RESPONSE	(PID) IPV4_DEFAULT_ROUTE	(PDL) 0x00

(PD) Not Present

Get Behavior:

If the device is not configured with a default route, both the IPv4 Default Route and Interface Identifier fields shall be NO_DEFAULT_ROUTE.

If a point-to-point link is used, the Interface Identifier field represents the interface used for the default route. The IPv4 Default Route field shall be set to NO_DEFAULT_ROUTE.

If a gateway is used, the IPv4 Default Route field shall contain the IPv4 address of the gateway. The Interface Identifier field may contain the interface used to reach the gateway if the device can provide this information. If the device cannot provide this information the Interface Identifier field shall be set to NO_DEFAULT_ROUTE.

Set Behavior:

To remove the default route, the IPv4 Default Route and Interface Identifier fields shall be set to NO_DEFAULT_ROUTE.

To use a point-to-point link as the default route, the IPv4 Default Route field shall be set to NO_DEFAULT_ROUTE. The Interface Identifier field shall contain the interface to use as the default route.

If a gateway is to be used as the default route, the IPv4 Default Route field shall contain the IPv4 address of the gateway to use. The Interface Identifier field shall be set to NO_DEFAULT_ROUTE.

4.12 Get/Set Name Servers (DNS_IPV4_NAME_SERVER)

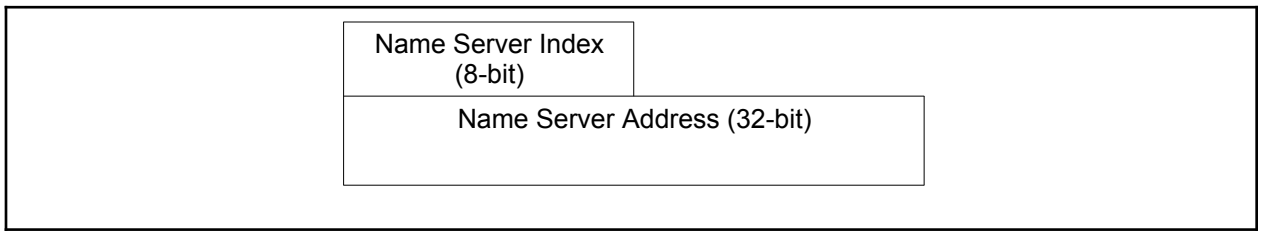
This parameter is used to set the IPv4 DNS name servers for a device. Up to three IPv4 name servers may be configured.

Controller: (GET)

(Port Id) 0x01 - 0xFF	(Message Count) 0x00	(Sub-Device) 0x0000 (Root) or 0x0001-0x0200
(CC) GET_COMMAND	(PID) DNS_IPV4_NAME_SERVER	(PDL) 0x01
(PD) Name Server Index (8-bit)		

Response:

(Response Type) ACK	(Message Count) 0x00 - 0xFF	(Sub-Device) Copy of Controller SD
(CC) GET_COMMAND_RESPONSE	(PID) DNS_IPV4_NAME_SERVER	(PDL) 0x05
(PD)		


Name Server Index
(8-bit)
Name Server Address (32-bit)

Controller: (SET)

(Port Id) 0x01 - 0xFF	(Message Count) 0x00	(Sub-Device) 0x0000 (Root) or 0x0001-0x0200 or 0xFFFF
(CC) SET_COMMAND	(PID) DNS_IPV4_NAME_SERVER	(PDL) 0x05
Name Server Index (8-bit) Name Server Address (32-bit)		

Response:

(Response Type) ACK	(Message Count) 0x00 - 0xFF	(Sub-Device) Copy of Controller SD
(CC) SET_COMMAND_RESPONSE	(PID) DNS_IPV4_NAME_SERVER	(PDL) 0x00
(PD) Not Present		

Data Description:

The valid IPv4 Name Server Index values are 0 to 2 inclusive. Requests outside this range shall be NACKed with NR_DATA_OUT_OF_RANGE.

If a device supports fewer than 3 IPv4 name servers, it shall use the name server index values starting from 0 and incrementing by one for each name server. Requests for all other values shall return NR_DATA_OUT_OF_RANGE.

The Name Server value is an IPv4 Address. It shall be set to IPV4_UNCONFIGURED if no name server is configured.

4.13 Get/Set Host Name (DNS_HOSTNAME)

This parameter is used to get and set the unqualified host name for a device. The hostname shall conform to the requirements in Section 2.1 of [HOST-NAME].

Devices shall not truncate the hostname. Upon receiving a SET, if the device has insufficient storage capacity to store the entire hostname, a NR_BUFFER_FULL shall be returned.

If the system hostname does not fit within 63 bytes a NR_HARDWARE_FAULT shall be returned.

Controller: (GET)

(Port Id) 0x01 - 0xFF	(Message Count) 0x00	(Sub-Device) 0x0000 (Root) or 0x0001-0x0200

(CC) GET_COMMAND	(PID) DNS_HOSTNAME	(PDL) 0x00
(PD) Not Present		

Response:

(Response Type) ACK	(Message Count) 0x00 - 0xFF	(Sub-Device) Copy of Controller SD
(CC) GET_COMMAND_RESPONSE	(PID) DNS_HOSTNAME	(PDL) (0x01 - 0x3F)
(PD) ASCII text. 1 to 63 characters.		

Controller: (SET)

(Port Id) 0x01 - 0xFF	(Message Count) 0x00	(Sub-Device) 0x0000 (Root) or 0x0001-0x0200 or 0xFFFF
(CC) SET_COMMAND	(PID) DNS_HOSTNAME	(PDL) (0x01 - 0x3F)
(PD) ASCII text. 1 to 63 characters.		

Response:

(Response Type) ACK	(Message Count) 0x00 - 0xFF	(Sub-Device) Copy of Controller SD
(CC) SET_COMMAND_RESPONSE	(PID) DNS_HOSTNAME	(PDL) 0x00
(PD) Not Present		

4.14 Get/Set Domain Name (DNS_DOMAIN_NAME)

This parameter is used to get and set the DNS domain name for a device. Domain names shall conform to Section 11 of [DNS-CLARIFICATIONS].

Furthermore, domain names for RDM devices shall be a maximum of 231 characters. This ensures the domain name fits in a single SET command.

Internationallized domain names may be supported by encoding with Punycode [IDNA].

An empty domain name implies there is no domain name configured. Setting the domain name to an empty string shall remove any existing domain name.

Devices shall not truncate the domain name. If the device has insufficient storage capacity to store the entire domain name, a NR_BUFFER_FULL shall be returned.

If the system domain name does not fit within 231 bytes a NR_HARDWARE_FAULT shall be returned.

Controller: (GET)

(Port Id) 0x01 - 0xFF	(Message Count) 0x00	(Sub-Device) 0x0000 (Root) or 0x0001-0x0200
(CC) GET_COMMAND	(PID) DNS_DOMAIN_NAME	(PDL) 0x00
(PD) Not Present		

Response:

(Response Type) ACK	(Message Count) 0x00 - 0xFF	(Sub-Device) Copy of Controller SD
(CC) GET_COMMAND_RESPONSE	(PID) DNS_DOMAIN_NAME	(PDL) (0x00 - 0xE7)
(PD) ASCII text. Up to 231 characters.		

Controller: (SET)

(Port Id) 0x01 - 0xFF	(Message Count) 0x00	(Sub-Device) 0x0000 (Root) or 0x0001-0x0200 or 0xFFFF
(CC) SET_COMMAND	(PID) DNS_DOMAIN_NAME	(PDL) (0x00 - 0xE7)
(PD) ASCII text. Up to 231 characters.		

Response:

(Response Type) ACK	(Message Count) 0x00 - 0xFF	(Sub-Device) Copy of Controller SD
(CC) SET_COMMAND_RESPONSE	(PID) DNS_DOMAIN_NAME	(PDL) 0x00
(PD) Not Present		

Appendix A: Defined Parameters (Normative)

IPV4_UNCONFIGURED 0x00000000

NO_DEFAULT_ROUTE 0x00000000

Table A-1: RDM Parameter ID Defines

GET Allowed	SET Allowed	RDM Parameter ID's (Slot 21-22)	Value	Comment	Required
		Category – IP & DNS Configuration			
✓		LIST_INTERFACES	0x0700	*Support required only if any other PID in Table A-1 is supported	✓✉
✓		INTERFACE_LABEL	0x0701		
✓		INTERFACE_HARDWARE_ADDRESS_TYPE1	0x0702		
✓	✓	IPV4_DHCP_MODE	0x0703		
✓	✓	IPV4_ZEROCONF_MODE	0x0704		
✓		IPV4_CURRENT_ADDRESS	0x0705		
✓	✓	IPV4_STATIC_ADDRESS	0x0706		
	✓	INTERFACE_RENEW_DHCP	0x0707		
	✓	INTERFACE_RELEASE_DHCP	0x0708		
	✓	INTERFACE_APPLY_CONFIGURATION	0x0709	*Support required only if the SET command for any of IPV4_DHCP_MODE, IPV4_ZEROCONF_MODE or IPV4_STATIC_ADDRESS are supported	✓*
✓	✓	IPV4_DEFAULT_ROUTE	0x070A		
✓	✓	DNS_IPV4_NAME_SERVER	0x070B		
✓	✓	DNS_HOSTNAME	0x070C		
✓	✓	DNS_DOMAIN_NAME	0x070D		

Table A-2: Additional NACK Reason Codes*

Response NACK Reason Codes	Value	Description
NR_ACTION_NOT_SUPPORTED	0x000B	The parameter data is valid but the SET operation cannot be performed with the current configuration.

*These are in addition to NACK Reason Codes defined in Table A-17 of E1.20

Table A-3: DHCP Mode Defines

DHCP Mode Defines	Value	Description
DHCP_STATUS_INACTIVE	0x00	The IP address was not obtained via DHCP.

DHCP_STATUS_ACTIVE	0x01	The IP address was obtained via DHCP.
DHCP_STATUS_UNKNOWN	0x02	The system cannot determine if the address was obtained via DHCP.

Appendix B: Example Message Exchange

The following examples are for a device which has two interfaces:

- eth0, the first interface, with a hardware type of 0x0001 (Ethernet) and an address of 192.168.0.1
- eth1, the second interface, also with a hardware type of 0x0001.

Reading the Current State

Controller sends GET: LIST_INTERFACES:

(Port Id) 0x01	(Message Count) 0x00	(Sub-Device) 0x0000	
(CC) GET_COMMAND	(PID) LIST_INTERFACES		(PDL) 0x00
(PD) Not Present			

Responder returns the packed list of interfaces:

(Response Type) ACK	(Message Count) 0x00	(Sub-Device) 0x0000	
(CC) GET_COMMAND_RESPONSE	(PID) LIST_INTERFACES		(PDL) 0x0c
(PD) 0x00000001 0x0001 0x00000002 0x0001			

Controller sends GET: INTERFACE_LABEL for the first interface:

(Port Id) 0x01	(Message Count) 0x00	(Sub-Device) 0x0000	
(CC) GET_COMMAND	(PID) INTERFACE_LABEL		(PDL) 0x04
(PD) 0x00000001			

Responder returns the interface name:

(Response Type) ACK	(Message Count) 0x00	(Sub-Device) 0x0000	
(CC)	(PID)		Variable

GET_COMMAND_RESPONSE	INTERFACE_LABEL	0x08
(PD) 0x00000001 “eth0”		

Controller sends GET: IPV4_CURRENT_ADDRESS for the first interface:

(Port Id) 0x01	(Message Count) 0x00	(Sub-Device) 0x0000
(CC) GET_COMMAND	(PID) IPV4_CURRENT_ADDRESS	(PDL) 0x04
(PD) 0x00000001		

Responder returns the interface's IPv4 address, network mask and DHCP mode. This response indicates the responder has a statically configured address of 192.168.0.1/24.

(Response Type) ACK	(Message Count) 0x00	(Sub-Device) 0x0000
(CC) GET_COMMAND_RESPONSE	(PID) IPV4_CURRENT_ADDRESS	(PDL) 0x0c
(PD) 0x00000001 0xc0a80001 24 | 0		

Enabling DHCP

Controller sends SET: IPV4_DCHP_MODE to enable DHCP for the first interface:

(Port Id) 0x01	(Message Count) 0x00	(Sub-Device) 0x0000
(CC) SET_COMMAND	(PID) IPV4_DHCP_MODE	(PDL) 0x05
(PD) 0x00000001 1		

Responder acknowledges:

(Response Type) ACK	(Message Count) 0x00	(Sub-Device) 0x0000
(CC) GET_COMMAND_RESPONSE	(PID) IPV4_DHCP_MODE	(PDL) 0x00
(PD) Not Present		

At this point the changes have yet to be applied.

Controller sends SET: INTERFACE_APPLY_CONFIGURATION for the first interface:

(Port Id) 0x01	(Message Count) 0x00	(Sub-Device) 0x0000
(CC) SET_COMMAND	(PID) INTERFACE_APPLY_CONFIGURATION	(PDL) 0x04
(PD) 0x00000001		

Responder acknowledges:

(Response Type) ACK	(Message Count) 0x00	(Sub-Device) 0x0000
(CC) GET_COMMAND_RESPONSE	(PID) INTERFACE_APPLY_CONFIGURATION	(PDL) 0x00
(PD) Not Present		

Static Configuration

Following on from the previous example, the Controller sends SET: IPV4_DCHP_MODE to disable DHCP for the first interface:

(Port Id) 0x01	(Message Count) 0x00	(Sub-Device) 0x0000
(CC) SET_COMMAND	(PID) IPV4_DHCP_MODE	(PDL) 0x05
(PD) 0x00000001 0		

Responder acknowledges:

(Response Type) ACK	(Message Count) 0x00	(Sub-Device) 0x0000
(CC) GET_COMMAND_RESPONSE	(PID) IPV4_DHCP_MODE	(PDL) 0x00
(PD) Not Present		

At this point the changes have yet to be applied.

Controller sends SET: IPV4_STATIC_ADDRESS for the first interface to set the address to 10.0.0.32/8

(Port Id) 0x01	(Message Count) 0x00	(Sub-Device) 0x0000
(CC) SET_COMMAND	(PID) IPV4_STATIC_ADDRESS	(PDL) 0x09
(PD) 0x00000001 0x0a000020 8		

Response:

(Response Type) ACK	(Message Count) 0x00	(Sub-Device) 0x0000
(CC) SET_COMMAND_RESPONSE	(PID) IPV4_STATIC_ADDRESS	(PDL) 0x00
(PD) Not Present		

At this point the changes have still yet to be applied.

Controller sends SET: INTERFACE_APPLY_CONFIGURATION for the first interface:

(Port Id) 0x01	(Message Count) 0x00	(Sub-Device) 0x0000
(CC) SET_COMMAND	(PID) INTERFACE_APPLY_CONFIGURATION	(PDL) 0x04
(PD) 0x00000001		

Responder acknowledges:

(Response Type) ACK	(Message Count) 0x00	(Sub-Device) 0x0000
(CC) GET_COMMAND_RESPONSE	(PID) INTERFACE_APPLY_CONFIGURATION	(PDL) 0x00
(PD) Not Present		