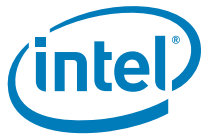


Intel® Boot Agent Application Notes for BIOS Engineers

Intel® 10/100/1000 Mb/s, 10 GbE &
40 GbE Ethernet Controllers for Servers

March 2015



No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

The products and services described may contain defects or errors which may cause deviations from published specifications.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting www.intel.com/design/literature.htm.

Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries.

* Other names and brands may be claimed as the property of others.

© 2015 Intel Corporation.



Contents

1.0	Introduction	5
2.0	Scope	5
3.0	Definitions	5
4.0	Boot Agent Image Options	6
4.1	Memory Requirements	6
4.2	BootUtil	6
5.0	PCI 3.0 Considerations	8
5.1	Differences Between PCI 2.x and PCI 3.0 Implementations	8
5.1.1	PCI 2.x BIOS	8
5.1.2	PCI 3.0 BIOS	8
6.0	EEPROM Considerations	8
7.0	EEPROM Contents	9
7.1	Compatibility — Word 0x03 (1 GbE Controllers); Word 0x10 (10 GbE Controllers)	9
7.2	Main Setup Options (Word 0x30)	10
7.3	Configuration Customization Options (Word 0x31)	11
7.4	Boot Agent Version (Word 0x32)	13
7.5	IBA Capabilities (Word 0x33)	13
7.6	IBA Secondary Port Configuration (Words 0x34 and 0x35)	14
7.7	iSCSI Option ROM Version (Word 0x36)	14
7.8	IBA Port Three Configuration (Words 0x38 and 0x39)	14
7.9	IBA Port Four Configuration (Words 0x3A and 0x3B)	14
7.10	VLAN Configuration Block Pointer (Word 0x3C)	14
7.11	iSCSI Boot Block Pointer (Word 0x3D)	14
8.0	VLAN / 802.1p Support	14
9.0	Troubleshooting Tips	16

Table

1. Boot Agent Error Messages	16
------------------------------	----



Revision History

Date	Revision	Description
March 2015	1.8	<ul style="list-style-type: none">Major update (all sections).
October 2014	1.7	<ul style="list-style-type: none">Updated Compliance to include 40 GbE PCI Express* (PCIe*) 1.0.00.Added note referencing the X710, XL710 controllers' expansion ROM residency in Flash storage requirement precluding integration into the BIOS and referral to datasheet for pre-boot function solution/details.Added note referencing PXE boot in UEFI environments.
November 2013	1.6	Updated sections 6.0, 8.0 and 9.0.
August 2012	1.5	Added VLAN Block Examples table to section 9.0.
July 2012	1.4	Revised section 3.0 (Definitions; BootUtil). Revised section 8.1 (Main Setup Options (Word 0x30; added bit 5 description). Revised section 8.4 (IBA Capabilities (Word 0x33; bit 6 and 5 description). Added new section 8.6 (iSCSI Option ROM Version; Word 0x36). Added new sections 8.9 (VLAN Configuration Block Pointer; Word 0x3C), 8.10 (iSCSI Boot Block Pointer; Word 0x3D), and 9.0 (VLAN/802.1p Support). Added VLAN support details. Added VLAN Configuration Block Pointer (Word 0x3C) details. Added iSCSI Boot Block Pointer (Word 0x3D) details.
October 2011	1.3	Major update (all sections).
June 2009	1.2	Major update (all sections).
December 2008	1.1	<ul style="list-style-type: none">Added Boot Agent release versions.Added Boot Agent memory requirements.Added new sections.Updated Table 1.
September 2007	1.0	Initial public release.



1.0 Introduction

The Intel® Boot Agent (Boot Agent) is a personal computer option ROM designed to supply industry standard PXE compliant pre-boot services for x86 architecture computers. The agent can be stored on a Flash component on an add-in Network Interface Card (NIC) or be integrated into the BIOS. Configuration data for the Boot Agent is stored in words 0x30-0x3E of a network device's EEPROM/NVM

The Intel Boot Agent for 10/100 Fast Ethernet, 1 Gigabit Ethernet (GbE), 10 GbE and 40 GbE controllers is compliant to the PCI Firmware Specification R3.0 according to the following release versions

- 10/100 - 4.2.00
- 10/100/1000 - 1.3.00
- 10 GbE PCI Express* (PCIe*) 2.1.00
- 40 GbE PCI Express* (PCIe*) 1.0.00

2.0 Scope

The purpose of this document is to assist BIOS developers in the process of integrating the Boot Agent into the BIOS and troubleshooting common problems (refer to [Table 1](#)).

Note: X710 and XL710 10/40 GbE controllers require that all expansion ROMs reside in the attached Flash storage. Consequently, integration of the expansion ROMs into the BIOS for those devices is not supported. This document addresses those devices in a very limited fashion. For details on pre-boot functions for such devices, please reference the *Intel® Ethernet Controller XL710 Datasheet*.

This document should be used in conjunction with a related EEPROM/NVM Programming Guide for your networking device.

This document does not address Pre-Boot Execution Environment (PXE) boot in Unified Extensible Firmware Interface (UEFI) environments. Intel provides UEFI drivers that are used in conjunction with other UEFI software components to provide network capabilities in UEFI environments. Refer to www.uefi.org for information on UEFI.

3.0 Definitions

- **Initialization Code** — The part of the option ROM that the BIOS calls during the Power On Self Test (POST). It verifies the LAN On Motherboard (LOM) or NIC configuration and signals the BIOS that the NIC is a bootable device. It is a part of all Boot Agent option ROM images.
- **Base Code** — The run-time code that controls the execution of the boot agent and includes the actual PXE protocol functions. This module is not hardware-specific and works with any Universal Network Device Interface (UNDI) driver.



- **UNDI Driver** — Used solely for the Ethernet controller. There is a UNDI driver for each LAN family: 10/100 GbE, 10 GbE and 40 GbE.
- **Setup Code** — An optional part of the Initialization Code that brings up the Ctrl-S setup screen allowing pre-boot configuration of the boot agent. Note that setup code is not typically used in LOM designs.
- **Monolithic Image** — a single image containing all the components of the Boot Agent (initialization code, UNDI driver, base code, etc.).
- **Split Image** — The base code and UNDI are created as separate images and are called separately by the BIOS. Each image has its own initialization code to provide the BIOS interface. Split images are no longer supported.
- **BootUtil** — The tool supplied by Intel used to create boot agent images and to perform field upgrades of option ROMs on Intel plug-in NICs. This tool is available for DOS*, Extensible Firmware Interface (EFI), Linux*, and Windows*. Bootutil can only create monolithic PXE images.
- **IBABuild.exe** — A DOS tool formerly supplied by Intel used to create Boot Agent images. IBABuild has been discontinued.

4.0 Boot Agent Image Options

When creating a boot image, be aware that there are different agents for 10/100 Mb/s, 10/100/1000 Mb/s, 10 GbE devices and 40 GbE devices. Split images are no longer supported. Only monolithic images can be created using BootUtil. In addition, split ROM images are no longer supported.

4.1 Memory Requirements

The Boot Agent modules have the following memory space requirements. Note that these values are approximate and might increase with future releases of the Boot Agent.

Memory Requirements	Initialization	Runtime
10/100 Fast Ethernet		
Monolithic Image	45 KB	4 KB
1 GbE		
Monolithic Image	56 KB	4 KB
10 GbE (PCIe)		
Monolithic Image	55 KB	4 KB
40 GbE (PCIe)		
Monolithic Image	62 KB	4 KB

4.2 BootUtil

BootUtil is provided for DOS, EFI (IA64 and x64 versions), Windows (IA32 and x64 versions) and Linux (IA32 and x64 versions). For Windows and Linux, BootUtil must first be installed before using. See the installation instructions provided with BootUtil for details.

Images are created with Bootutil by using the following command line parameters:

- **devid=0xyyyy** – This specifies the PCI device ID of the target device, where yyyy represents the four hexadecimal digit value of the ID.



- di=(desired contents of the image) – This specifies the components that can be included in the resulting image. Separate each entry with a plus (+) sign and no spaces. Possible values are;
 - PXE - for legacy PXE)
 - iSCSI - for legacy Internet Small Computer System Interface (iSCSI) boot
 - FCoE - for legacy Fibre Channel over Ethernet (FCoE) boot
 - EFI - for x64 version of the UEFI driver
 - CLP - for the SMASH/CLP agent, which is needed in cases where multiple legacy expansion ROMs co-exist in a single combination ROM image.
- LOM - This specifies that the resulting output file should be formatted for inclusion in a BIOS for a LOM implementation and created with a file name extension of .LOM. If this is not specified on the command line, the resulting image is formatted for storage on a Flash chip locally attached to the controller and created with a file name extension of .NIC. LOM images normally contain only one component and only PXE, iSCSI boot or FCoE boot.

Example 1, using Bootutil to create a PXE LOM image for device ID 0x1533:

```
Bootutil -devid=0x1533 -di=pxe -lom
```

Example 2, using Bootutil to create a combo image for a NIC with a device ID of 0x1533 containing PXE, iSCSI boot and EFI:

```
Bootutil -devid=0x1533 -di=pxe+iscsi+clp+efi
```



5.0 PCI 3.0 Considerations

If the BIOS reports itself as supporting PCI firmware version 3.0 or higher, it must also set the BX CPU register to the final memory address of the option ROM memory area before calling the initialization code. This, and the required input parameters, are detailed in the PCI Firmware 3.0 (BIOS) specification. The PXE software has no method to validate this address and, therefore, uses whatever value the BIOS passes in. For example, if the option ROM is loaded at 0xC900:0, BX then needs to be set to 0xC900 before calling the PXE initialization code.

The size of the destination block in the Upper Memory Block (UMB) must be at least the run-time size as specified in the PCI Data Structure (PCIR) header of the option ROM. If not, the system might not be able to boot.

The initialization/UNDI component is created with an integrated table of all supported PCI ID's. A PCI 2.x BIOS does not use this table; it only associates with the PCI ID that was specified to BootUtil on the command line when it was created. A PCI 3.0 BIOS supports all ID's listed in the table in the initialization/UNDI component.

A PCI 3.0 BIOS should restrict embedded PXE for LOMs only. It must not allow it to load for compatible plug-in NICs, as they should carry their own version if applicable.

5.1 Differences Between PCI 2.x and PCI 3.0 Implementations

5.1.1 PCI 2.x BIOS

A complete image must be built for each individual controller. Also, each image contains one and only one UNDI.

5.1.2 PCI 3.0 BIOS

- If using monolithic images, only one image is needed for multiple controllers, assuming they are of the same family (such as all GbE controllers).

6.0 EEPROM Considerations

The adapter EEPROM image holds the Boot Agent configuration data in words 0x30 through 0x3E for 1 GbE components, and words 0x30 through 0x36 for 10 GbE components, with the checksum stored in word 0x3F. Configuration data can be stored for up to four interfaces. Refer to the documentation for your specific Ethernet component for details.

The Boot Agent writes up to four interfaces to the EEPROM when it initializes under some circumstances.

- If the Boot Agent version number stored in word 0x32 does not match the actual Boot Agent version, it corrects the version number.
- If configuration changes are made via the setup utility that is built into the Boot Agent and invoked by pressing Ctrl-S when the appropriate prompt appears after POST, any changes are written to the EEPROM.



7.0 EEPROM Contents

The EEPROM configuration data in this section is included for reference purposes only. The EEPROM images supplied for Intel Ethernet controllers are properly configured by default. Editing the EEPROM image should only be done after verifying the changes with your Intel technical representative.

7.1 Compatibility — Word 0x03 (1 GbE Controllers); Word 0x10 (10 GbE Controllers)

Bit(s)	Name	Function
15:12	N/A	Component specific. Refer to the appropriate datasheet for your Ethernet controller.
11	LOM	<p>This bit indicates whether the NVM attached to the Ethernet controller contains a dedicated area for storing option ROM firmware. This bit is used by option ROM software update utilities to decide whether or not to enable updating the option ROM firmware as well as whether or not to allow enabling/disabling the option ROM by means of the PCI Expansion ROM Base Address Register (BAR).</p> <p>0b = NIC (Attached Flash has an area for storing option ROM firmware).</p> <p>1b = LOM (Attached Flash has no area for storing option ROM firmware). This bit must be set for a LOM. If not, the utility software might enable the expansion ROM BAR, which can cause failures in designs lacking the corresponding dedicated Flash.</p> <p>Note: This bit is not applicable to the 40 GbE capable controllers (X710/XL710). All option ROMs are stored locally on these devices.</p>
10:0	N/A	Component specific. Refer to the appropriate datasheet for your Ethernet controller.



7.2 Main Setup Options (Word 0x30)

Configuring software is controlled by the EEPROM on a NIC. The Main Setup options are stored in word 0x30. These options are those that can be changed by the user via the Control-S setup menu or by using the BootUtil.exe utility. On the X710 and XL710 controllers, this field is a pointer to the start of the PXE setup options block. See the *Intel® Ethernet Controller XL710 Datasheet* for those devices for details. Word 0x30 has the following format.

Bit(s)	Name	Function
15	RFU	Reserved. Must be 0b.
14	RFU	Reserved. Must be 0b.
13	RFU	Reserved. Must be 0b.
12:10	FSD	Bits 12:10 control forcing speed and duplex during driver operation. Valid values are: 000b = Auto-negotiate. 001b = 10 Mb/s Half duplex. 010b = 100 Mb/s Half duplex. 011b = Not valid (treated as 000b). 100b = Not valid (treated as 000b). 101b = 10 Mb/s Full duplex. 110b = 100 Mb/s Full duplex. 111b = 1000 Mb/s Full duplex. Only applicable for twisted-pair copper adapters. Default value is 0x000. This setting is not documented in the datasheet, as it can have negative consequences if used inappropriately.
9	LWS	Legacy OS Wakeup Support. (For 82559-based adapters only) If set to 1b, the agent enables PME in the NIC's PCI configuration space during initialization. This enables remote wake up under legacy operating systems that don't normally support it. Note that enabling this makes the NIC technically non-compliant with the ACPI specification, which is why the default is disabled. 0b = Disabled (default value). 1b = Enabled.
8	DSM	Display Setup Message. If the bit is set to 1b, the "Press Control-S" message is displayed after the title message. Default value is 1b.
7:6	PT	Prompt Time. These bits control how long the "Press Control-S" setup prompt message is displayed, if enabled by DIM. 00b = 2 seconds (default) 01b = 3 seconds 10bb = 5 seconds 11 = 0 seconds Note: The Ctrl-S message is not displayed if the zero seconds prompt time is selected.



Bit(s)	Name	Function
5	iBD	iSCSI Boot Disabled. When the iSCSI Boot Disabled bit is set and the adapter port is neither iSCSI primary nor secondary, the setup code must not be loaded; Otherwise, the iSCSI banner and Setup menu should be accessible. 0b = iSCSI enabled. 1b = iSCSI disabled.
4:3	DBS	Default Boot Selection. These bits select which device is the default boot device. These bits are only used if the agent detects that the BIOS does not support boot order selection or if the MODE field of word 0x31 is set to MODE_LEGACY. 00b = Network boot, then local boot. 01b = Local boot, then network boot. 10b = Network boot only. 11b = Local boot only.
2:0	PS	Protocol Select. These bits select the boot protocol. 000b = PXE enabled on this port. 001b = This port disabled for PXE, iSCSI boot and FCoE. 010b = This port is the iSCSI boot primary port. 011b = This port is the iSCSI boot secondary port. 100b = This port enabled for FCoE. 101b = Invalid. 110b = Invalid. 111b = Invalid.

7.3 Configuration Customization Options (Word 0x31)

Word 0x31 of the EEPROM contains settings that can be programmed by an OEM or network administrator to customize the operation of the software. These settings cannot be changed from within the Control-S setup menu. The lower byte contains settings typically configured by a network administrator using the BootUtil.exe utility since these settings generally control which setup menu options are changeable. The upper byte is generally settings for use by OEMs to control the operation of the agent in a LOM environment, although there is nothing in the agent to prevent their use on a NIC implementation. On the X710 and XL710 controllers, this field is a pointer to the start of the PXE Configuration Customization options block. See the datasheet for those devices for details. Word 0x31 has the following format.



Bit(s)	Name	Function
15:14	SIG	Signature. Must be set to 01b to indicate that the agent or other configuration software has programmed this word.
13	RFU	Reserved. Must be 0b.
12	RFU	Reserved. Must be 0b.
11	RETRY	If set to 1b, enables continuous retry. If enabled, this feature modifies the boot agent behavior to continually attempt to connect to the server until the server becomes available. The default is 0b. This feature is not documented in the datasheet, since it can have negative consequences if used inappropriately.
10:8	MODE	Selects the agent's boot order setup mode. This field changes the agent's default behavior in order to make it compatible with systems that do not completely support the BBS and PnP Expansion ROM standards. Valid values and their meanings are:
		000b Normal behavior. The agent attempts to detect BBS and PnP expansion ROM support as it normally does.
		001b Force Legacy Mode. The agent does not attempt to detect BBS or PnP Expansion ROM support in the BIOS and assumes the BIOS is not compliant. User's can change the BIOS boot order in the Setup Menu.
		010b Force BBS Mode. The agent assumes the BIOS is BBS-compliant, even though it might not be detected as such by the agent's detection code. User's CANNOT change the BIOS boot order in the Setup Menu.
		011b Force PnP Int18 Mode. The agent assumes the BIOS enables boot order setup for PnP Expansion ROMs and hooks interrupt 0x18 (to inform the BIOS that the agent is a bootable device) in addition to registering as a BBS IPL device. User's CANNOT change the BIOS boot order in the Setup Menu.
		100b Force PnP Int19 Mode. The agent assumes the BIOS enables boot order setup for PnP Expansion ROMs and hooks interrupt 0x19 (to inform the BIOS that the agent is a bootable device) in addition to registering as a BBS IPL device. User's CANNOT change the BIOS boot order in the Setup Menu.
		101b Reserved for future use. If specified, is treated as a value of 000b.
		110b Reserved for future use. If specified, is treated as a value of 000b.
		111b Reserved for future use. If specified, is treated as a value of 000b.
7	RFU	Reserved. Must be 0b.
6	RFU	Reserved. Must be 0b.
5	DFU	Disable Flash Update. If this bit is set to 1b, the user is not allowed to update the Flash image using Intel PROSet for Windows* Device Manager. Default value is 0b.
4	DLWS	Disable Legacy Wakeup Support. If this bit is set to 1b, the user is not allowed to change the Legacy OS Wakeup Support menu option. Default value is 0b.
3	DBS	Disable Boot Selection. If this bit is set to 1b, the user is not allowed to change the boot order menu option. Default value is 0b.



Bit(s)	Name	Function
2	DPS	Disable Protocol Select. If set to 1b, the user is not allowed to change the boot protocol. Default value is 0b.
1	DTM	Disable Title Message. If this bit is set to 1b, the title message displaying the version of the boot agent is suppressed; the Control-S message is also suppressed. This is for OEMs who do not want the boot agent to display any messages at system boot. Default value is 0b.
0	DSM	Disable Setup Menu. If this bit is set to 1b, the user is not allowed to invoke the setup menu by pressing Control-S. In this case, the EEPROM can only be changed via an external program. Default value is 0b.

7.4 Boot Agent Version (Word 0x32)

Word 0x32 of the EEPROM is used to store the version of the boot agent that is stored in the Flash image. When the boot agent loads, it can check this value to determine if any first-time configuration needs to be performed. The agent then updates this word with its version. Some diagnostic tools that report the version of the boot agent in the Flash also read this word. The format of this word is:

Bit(s)	Name	Function
15:12	MAJ	PXE Boot Agent Major Version. Default value is 0b.
11:8	MIN	PXE Boot Agent Minor Version. Default value is 0b.
7:0	BLD	PXE Boot Agent Build Number. Default value is 0b.

7.5 IBA Capabilities (Word 0x33)

Word 0x33 of the EEPROM is used to enumerate the boot technologies programmed into the Flash. This is updated by IBA configuration tools and is not updated or read by IBA.

Bit(s)	Name	Function
15:14	SIG	Signature. Must be set to 01b to indicate that this word has been programmed by the agent or other configuration software.
13	PXEC	Enable PXE execution control. If this bit is set to 1b, PXE reads the setup options word associated to the port and only executes if PXE is enabled (bits 2:0 = 000b). If this bit is set to 0b, PXE executes when called regardless of the individual port setting. The default value is 0b.
12:6	RFU	Reserved. Must be 0b.
5	FCoE	FCoE boot capability is present in Flash if this bit is set to 1b (applicable to PCIe 10 GbE connections).
4	iSCSI	iSCSI boot capability is present in Flash if set to a 1b (applicable to PCIe GbE and 10 GbE connections).
3	EFI	EFI UNDI capability is present in Flash if set to a 1b.
2	RPL	No longer used.
1	UNDI	PXE/UNDI capability is present in Flash if set to a 1b.
0	BC	PXE base code is present in Flash if set to a 1b.



7.6 IBA Secondary Port Configuration (Words 0x34 and 0x35)

Words 0x34 and 0x35 are used to provide a unique configuration for the second port of multiple MAC devices such as the 82546. The format is the same as that used in words 0x30 and 0x31 for the first port. For X710/XL710, see the datasheet for details.

7.7 iSCSI Option ROM Version (Word 0x36)

NVM Word 0x36 is used to store the version of the iSCSI Option ROM. It is stored in the same format as the PXE Version at Word 0x32. For the X710 or XL710, see the datasheet for details.

7.8 IBA Port Three Configuration (Words 0x38 and 0x39)

Words 0x38 and 0x39 are used to provide a unique configuration for the third port of quad MAC devices such as the 82580. These words are not used by 10 GbE components. The format is the same as that used in words 0x30 and 0x31 for the first port.

7.9 IBA Port Four Configuration (Words 0x3A and 0x3B)

Words 0x3A and 0x3B are used to provide a unique configuration for the fourth port of quad port MAC devices such as the 82580. These words are not used by 10 GbE components. The format is the same as that used in words 0x30 and 0x31 for the first port. For X710/XL710. See the datasheet for details.

7.10 VLAN Configuration Block Pointer (Word 0x3C)

Word 0x3C contains a pointer to the starting address of the VLAN configuration block for GbE PCIe products (address 0x20 is used for 10 GbE PCIe devices). A value of 0xFFFF or 0x0000 in this location means that the block is not present, thus PXE VLAN tagging is not possible for a given adapter/NIC. This feature is only available on PCIe devices (see [Section 8.0](#)). For X710/XL710, see the datasheet for details.

7.11 iSCSI Boot Block Pointer (Word 0x3D)

Word 0x3D contains a pointer to the starting address of the iSCSI boot configuration block for GbE PCIe products (address 0x17 is used for 10 GbE PCIe devices). This feature is only available on PCIe devices. For X710/XL710, see the datasheet for details.

8.0 VLAN / 802.1p Support

Starting with GbE PXE version 1.3.95, 10 GbE PXE version 2.2.05 and 40 GbE PXE version 1.0.00, VLAN tagging and priority tagging is supported on PCIe devices only. This feature is not disclosed in end user documentation. The EEPROM image must be configured for VLAN support before software utilities enable the setting of VLAN or priority.

To specify a priority value, the command line parameter `-PXEVLANPRIORITY = <PRIORITYVALUE>` must be used in conjunction with the `-PXEVLAN` parameter. If the `-PXEVLANPRIORITY` parameter is used without the `-PXEVLAN` parameter, it is ignored. This feature is not documented for end users but is available on supported devices.



Values passed to the -PXEVLAN option must be a 12-bit number, in a range from 0 to 4095 (FFF in hexadecimal), specified in decimal or hexadecimal form. A value of 0 disables VLANs and a value from 1 to 4095 enables VLANs and set the stated VLAN number. Any value exceeding this range is rejected by BootUtil.

A hexadecimal form must be preceded by a 0x prefix (such as -PXEVLAN=0x24). Also, a decimal form must not be preceded by any prefix (such as -PXEVLAN=36).

When -PXEVLANPRIORITY is entered, the option value is used as a VLAN priority value, in a range of 0 to 7, specified in decimal or hexadecimal form.

A hexadecimal form must be preceded by the 0x prefix (such as -PXEVLANPRIORITY=0x5).

- A decimal form must not be preceded by any prefix (such as -PXEVLANPRIORITY=5).
- In all cases, the Bootutil command must also specify the NIC number. For example, BootUtil -NIC=5 -PXEVLAN=0x23 -PXEVLANPRIORITY=0x2.

The EEPROM stores the VLAN configuration in a block. The starting address of the block is defined in word 0x3C for GbE PCIe devices and word 0x20 for 10 GbE PCIe devices. The total size of the block varies depending on how many ports the device supports. One, two or four ports can be supported.

VLAN configuration data is stored in the PXE VLAN configuration block as listed in the following table:

Configuration Item	Size in Bytes	Comments
VLAN Block Signature	2	V and L.
Structure Version	1	Version of this structure. Should be set to 1b.
Block Size	1	Total byte size of the configuration block.
Port 0 VLAN Tag	2	VLAN tag and priority values for the first port of the Ethernet controller (see bit definition in the table that follows for details). A value of zero means no VLAN is configured for the port.
Port 1 VLAN Tag	2	VLAN tag and priority values for the second port of the Ethernet controller (see bit definitions in the table that follows for details). A value of zero means no VLAN is configured for the port.
Port 2 VLAN Tag	2	VLAN tag and priority values for the fourth port of the Ethernet controller (see bit definitions in the table that follows for details). A value of zero means no VLAN is configured for the port.
Port 3 VLAN Tag	2	VLAN tag and priority values for the third port of the Ethernet controller (see bit definitions in the table that follows for details). A value of zero means no VLAN is configured for the port.



The VLAN tag value and priority tag value is stored separately for every port, as listed in the previous table named Port x VLAN tag, where x is a port number in the range of 0 to 3), one NVM word per port (16 bits), and in following form:

Bits	15:13	12	11:0
Field Name	Priority (0: 7)	Reserved, always 0x0	VLAN ID (1 to 4095)

VLAN Block Examples	
4C56 0601 E01A	1 port device, set to priority 7 and VLAN 0x1A (26d)
4C56 0801 0000 C001	2 port device, port 1 set to VLAN disabled, port 2 set to priority 6 and VLAN 0x1
4C56 0C01 0000 0000 0000 0000	4 port device, VLAN disabled on all ports

9.0 Troubleshooting Tips

- The Intel boot agent is a BEV device, and does not operate correctly if configured as a BCV device.
- Unpredictable results can be due to extraneous copies of the option ROM in the BIOS image. As BIOS are developed and maintained, sometimes whole copies of old versions of the boot agent, or individual components, become orphaned in the BIOS image. Search out and eliminate any extra copies of the option ROM.
- When using split images, all components should be updated, not just the UNDI. Updates and fixes are frequently made to other components, and they all should be updated to the same release.
- Legacy expansion ROMs, such as the Intel Boot Agent, are limited by the legacy BIOS architecture to a maximum of 32-bit memory access. If the BIOS assigns a memory BAR to the controller that is above 32-bit, the Boot Agent is unable to access the device. In this case, there are 4 options to consider:
 - a. Configure the BIOS to only assign 32-bit memory BAR addresses.
 - b. Modify the BIOS to selectively assign 32-bit memory BAR addresses to devices that support PXE.
 - c. Configure the Ethernet controller to only request 32-bit memory BAR addresses. All Intel PCIe Ethernet controllers have EEPROM settings that limit the device to advertising only 32-bit memory BAR support.
 - d. If the BIOS is UEFI capable, implement a UEFI-based solution instead of legacy PXE. Memory addressing is not a limitation of the UEFI environment.

Table 1. Boot Agent Error Messages

Message Text	Cause/Comments
Invalid PMM function number.	PMM is not installed or is not working correctly.
PMM allocation error.	PMM could not or did not allocate the requested amount of memory for driver usage.
PXE-E01: PCI Vendor and Device IDs do not match!	The device ID of the interface is not on the Boot Agent list of supported ID's. Validate that the device ID is correct. Programmers might need a newer version of the Boot Agent.



Message Text	Cause/Comments
PXE-E04: Error reading PCI configuration space. The Intel Boot Agent cannot continue.	PCI configuration space could not be read. The Boot Agent was unable to read one or more of the NIC's PCI configuration registers. The Boot Agent returns control to the BIOS and does not attempt to remote boot. Try using a newer version of the Boot Agent.
PXE-E05: The LAN adapter's configuration is corrupted or has not been initialized. The Intel Boot Agent cannot continue.	The Boot Agent determined that the adapter EEPROM checksum is incorrect. The agent returns control to the BIOS and does not attempt to remote boot. On products that use a shared Flash, such as the 82566, 82567, 82577, 82578, or 82579 in combination with the ICH or PCH, insure that the BIOS has set the VSCC values correctly, and that an adequate number of blocks of Flash have been allocated to store two complete versions of the boot agent. Refer to the SPI Flash Programming Guide for the chipset you are using.
PXE-E06: Option ROM requires DDIM support.	The system BIOS does not support DDIM. The BIOS does not support the mapping of the PCI expansion ROMs into upper memory as required by the PCI specification. The Intel Boot Agent cannot function in this system. The Intel Boot Agent returns control to the BIOS and does not attempt to remote boot.
PXE-E07: PCI BIOS calls not supported.	BIOS-level PCI services not available. System is probably not PCI compliant.
PXE-E09: Unexpected UNDI loader error. Status == xx	The UNDI loader returned an unknown error status. xx is the status returned.
PXE-E20: BIOS extended memory copy error.	BIOS could not move the image into extended memory.
PXE-E20: BIOS extended memory copy error. AH == xx	Error occurred while trying to copy the image into extended memory. xx is the BIOS failure code.
PXE-E51: No DHCP or BOOTP offers received.	The Boot Agent did not receive any DHCP or BOOTP responses to its initial request. Please make sure that the DHCP server (and/or proxy DHCP server, if one is in use) is properly configured and has sufficient IP addresses available for lease. If you are using BOOTP instead, make sure that the BOOTP service is running and is properly configured. This is most likely a network configuration problem.
PXE-E53: No boot filename received.	The Boot Agent received a DHCP or BOOTP offer, but has not received a valid file name to download. If using PXE, check the PXE and BINL configuration. If using BOOTP, be sure that the TFTP service is running and that the specific path and file name are correct. This is most likely a network configuration problem.
PXE-E61: Media test failure.	The NIC does not detect link. Make sure that the cable is good and is attached to a working hub or switch. The link light visible from the back of the NIC should be lit.
PXE-EC1: Base-code ROM ID structure was not found.	No base code could be located. An incorrect Flash image is installed or the image has become corrupted. This should only happen on split ROM configurations.
PXE-EC3: BC ROM ID structure is invalid.	Base code could not be installed. An incorrect Flash image is installed or the image has become corrupted. This should only happen on split ROM configurations.
PXE-EC4: UNDI ROM ID structure was not found.	UNDI ROM ID structure signature is incorrect. An incorrect Flash image is installed or the image has become corrupted. This should only happen on split ROM configurations.
PXE-EC5: UNDI ROM ID structure is invalid.	The structure length is incorrect. An incorrect Flash image is installed or the image has become corrupted. This should only happen on split ROM configurations.
PXE-EC6: UNDI driver image is invalid.	The UNDI driver image signature was invalid. An incorrect Flash image is installed or the image has become corrupted. This should only happen on split ROM configurations.



Message Text	Cause/Comments
PXE-EC8: !PXE structure was not found in UNDI driver code segment.	<p>The Boot Agent could not locate the needed !PXE structure resource. An incorrect Flash image is installed or the image has become corrupted.</p> <p>When PXE tries to boot the system, it copies the data from PMM memory to conventional memory (below 640 KB). Then it tries to find the !PXE data structure that is embedded in the UNDI driver code segment. This error implies that the BIOS is either corrupting the PMM memory area after PXE saves itself in conventional memory, or the memory allocated by PMM is not writeable and therefore the PXE image doesn't get saved correctly. This can also be caused by the memory BAR being set above the 4G boundary.</p>
PXE-EC9: PXENV + structure was not found in UNDI driver code segment.	<p>The Boot Agent could not locate the needed PXENV+ structure. An incorrect Flash image is installed or the image has become corrupted. This can be caused by the memory BAR being set above the 4G boundary.</p>
PXE-M0F: Exiting Intel Boot Agent.	Ending execution of the ROM image.
This option has been locked and cannot be changed.	<p>There was an attempt to change a configuration setting that has been locked by BootUtil. This message appears either from within Intel® PROSet's Boot Options tab when operating under Windows* or from the Configuration Setup Menu when operating in a stand-alone environment.</p>
PXE-M0E: Retrying network boot; press ESC to cancel.	<p>The Boot Agent did not successfully complete a network boot due to a network error (such as not receiving a DHCP offer). The Boot Agent continues to attempt to boot from the network until successful or until canceled by the user. This only happens if the continuous retry feature is enabled in the Custom Configuration word in the EEPROM for this port.</p>