

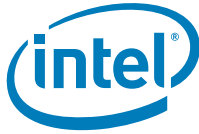


# Intel® Boot Agent Application Notes for BIOS Engineers

Intel® 10/100/1000 Mb/s and 10 GbE Ethernet Controllers

---

*June 2009*



INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. Intel products are not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications.

Intel may make changes to specifications and product descriptions at any time, without notice.

Intel Corporation may have patents or pending patent applications, trademarks, copyrights, or other intellectual property rights that relate to the presented subject matter. The furnishing of documents and other materials and information does not provide any license, express or implied, by estoppel or otherwise, to any such patents, trademarks, copyrights, or other intellectual property rights.

**IMPORTANT - PLEASE READ BEFORE INSTALLING OR USING INTEL® PRE-RELEASE PRODUCTS.**

Please review the terms at [http://www.intel.com/netcomms/prerelease\\_terms.htm](http://www.intel.com/netcomms/prerelease_terms.htm) carefully before using any Intel® pre-release product, including any evaluation, development or reference hardware and/or software product (collectively, "Pre-Release Product"). By using the Pre-Release Product, you indicate your acceptance of these terms, which constitute the agreement (the "Agreement") between you and Intel Corporation ("Intel"). In the event that you do not agree with any of these terms and conditions, do not use or install the Pre-Release Product and promptly return it unused to Intel.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

Intel processor numbers are not a measure of performance. Processor numbers differentiate features within each processor family, not across different processor families. See [http://www.intel.com/products/processor\\_number](http://www.intel.com/products/processor_number) for details.

The Ethernet Controllers described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Hyper-Threading Technology requires a computer system with an Intel® Pentium® 4 processor supporting HT Technology and a HT Technology enabled chipset, BIOS and operating system. Performance will vary depending on the specific hardware and software you use. See [http://www.intel.com/products/ht/Hyperthreading\\_more.htm](http://www.intel.com/products/ht/Hyperthreading_more.htm) for additional information.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an ordering number and are referenced in this document, or other Intel literature, may be obtained from:

Intel Corporation  
P.O. Box 5937  
Denver, CO 80217-9808

or call in North America 1-800-548-4725, Europe 44-0-1793-431-155, France 44-0-1793-421-777, Germany 44-0-1793-421-333, other Countries 708-296-9333.

Intel and Intel logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

Copyright © 2009, Intel Corporation. All Rights Reserved.



## Contents

---

<b>1.0</b>	<b>Introduction</b>	5
<b>2.0</b>	<b>Scope</b>	5
<b>3.0</b>	<b>Definitions</b>	5
<b>4.0</b>	<b>Boot Agent Image Options</b>	6
4.1	Split ROM vs. Monolithic Images	6
<b>5.0</b>	<b>PCI 3.0 Considerations</b>	7
5.1	Differences Between PCI 2.x and PCI 3.0 Implementations	7
5.1.1	PCI 2.x BIOS	7
5.1.2	PCI 3.0 BIOS	7
<b>6.0</b>	<b>EEPROM Considerations</b>	8
<b>7.0</b>	<b>Causes of Boot Agent Related EEPROM Corruption</b>	8
7.1	Block Allocation Error	8
7.2	Vendor Specific Component Capabilities (VSCC) Not Set Correctly	8

## Table

1. Boot Agent Error Messages	9
------------------------------	---



## Revision History

---

Date	Revision	Description
June 2009	1.2	Major update (all sections).
December 2008	1.1	<ul style="list-style-type: none"><li>• Added Boot Agent release versions.</li><li>• Added Boot Agent memory requirements.</li><li>• Added new sections.</li><li>• Updated Table 1.</li></ul>
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 it can 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) and 10 GbE controllers are 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

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

## 3.0 Definitions

- Initialization Code — This is a part of the option ROM that the BIOS calls during 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 — This is the runtime code that controls the execution of the boot agent, and includes the actual PXE protocol functions. This module is not hardware-specific and operates with any Universal Network Device Interface (UNDI) driver.
- UNDI Driver — This is UNDI and is just the driver for the Ethernet controller. There is an UNDI driver for each LAN family (10/100, GbE, PCI-X 10 GbE, and PCIe 10 GbE).
- Setup Code — This is 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.
- IBABuild.exe — A DOS\* tool supplied by Intel used to create Boot Agent images. The accompanying file IBABuild.txt contains complete usage details.

## 4.0 Boot Agent Image Options

When creating a boot image, programmers have several options to consider. First, there are different agents for 10/100 Mb/s, 10/100/1000 Mb/s, and 10 GbE devices. Please note that the gigabit version of the Boot Agent is always used in conjunction with the MAC embedded in the I/O Controller Hub 8 (ICH8), ICH9, and ICH10 components, even when the 82562V 10/100 Mb/s PHY is attached. Secondly, the boot agent can be created as a split-ROM image or a monolithic image. A monolithic image contains all the elements of the Boot Agent in a single binary. A split image has the UNDI in one binary and the Base Code in a separate binary.

### 4.1 Split ROM vs. Monolithic Images

The benefit of a split ROM comes from the way that option ROMs are initialized on Lan On Motherboard (LOM) systems. There are three pieces to a PXE option ROM image:

- Init/loader code
- UNDI driver
- PXE base code

For the typical monolithic Flash image, all three pieces are treated as a single option ROM and loaded into memory at the same time. If the BIOS supports POST Memory Management (PMM), the init code can copy the UNDI driver and PXE base code into temporary memory storage and shrink itself down to a smaller upper memory size after initialization (refer to the following table). This enables the BIOS to be able to handle more option ROM images in the available memory space.

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
<b>10/100 Fast Ethernet</b>			
Monolithic Image		45 KB	4 KB
Split Image	Init/UNDI	17 KB	4 KB
	Base Code	30 KB	3 KB
<b>1 GbE</b>			
Monolithic Image		56 KB	4 KB
Split Image	Init/UNDI	30 KB	4 KB
	Base Code	30 KB	3 KB
<b>10 GbE (PCIe)</b>			
Monolithic Image		55 KB	4 KB
Split Image	Init/UNDI	27 KB	4 KB
	Base Code	30 KB	3 KB



The monolithic PXE image is relatively large, so this means even if the BIOS supports PMM, it has to have that much free space just to begin option ROM initialization. By using a split ROM, the PXE base code is broken out into its own option ROM image. Now there are two option ROMs that need to be initialized. The Boot Agent split ROM consists of two modules, each of which is smaller than the monolithic image. The combined size of the two is greater, but the BIOS requires less free space to initialize each one. Each image also shrinks after initialization. The final result is that it takes up a little more space in upper memory at boot time but doesn't take as much during initialization.

The monolithic and split Init/UNDI images are PCI 3.0 compliant and can be loaded in conventional memory during initialization. The split base code image is still PCI 2.x compliant and must be loaded into the UMB.

## 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, then BX needs to be set to 0xC900 before calling the PXE initialization code.

The initialization/UNDI component is created with an integrated table of all supported PCI ID's, whether it is created separately in a split image, or as part of a monolithic image. A PCI 2.x BIOS does not use this table; it only associates with the PCI ID that was specified to IBABuild 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.

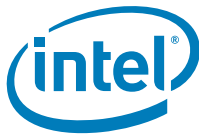
### 5.1 Differences Between PCI 2.x and PCI 3.0 Implementations

#### 5.1.1 PCI 2.x BIOS

- If using monolithic images, a complete image must be built for each individual controller. Also, each image contains one and only one UNDI.
- If using split images, it is possible to create two different UNDI images and only one base code; however, this requires a skilled and knowledgeable BIOS developer to control how the components are handled by the BIOS. If done this way, the two controllers can be of different families (such as 1 GbE and 10 GbE), but one UNDI for each controller is still required.

#### 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).
- If using split images, only one UNDI is needed, assuming the controllers are of the same family.
- It is possible to support controllers from two different families (such as 1 GbE and 10 GbE) by using two UNDI components, but only one base code image. Again, this is not recommended unless the BIOS developer is experienced in this type of configuration.



## 6.0 EEPROM Considerations

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

The Boot Agent writes 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.
- For ICH8/9/10 only, if the NVM checksum in word 0x3F is incorrect, and the NVM signature bits indicate that the checksum has not been previously calculated, the Boot Agent calculates the checksum, stores it, and then modifies the signature bits to indicate that the checksum has been set. If the checksum is invalid and the Boot Agent finds that the signature bits are set, then the Boot Agent does not correct the checksum and instead declares the checksum invalid and exits.

## 7.0 Causes of Boot Agent Related EEPROM Corruption

### 7.1 Block Allocation Error

When using a shared Flash, such as used with the ICH8, ICH9 or ICH10, the entire EEPROM image area, including the boot agent configuration words, must be duplicated to allow for both an active and an inactive copy of the EEPROM image. This enables the EEPROM image update code to erase and then program the inactive area, mark it active, and mark the previously active copy as inactive. If there are not enough blocks of Flash memory available to maintain both copies, when the boot agent or software tools write to the Flash, it might cause the EEPROM image to become corrupt and render the LAN device unusable. Note that each area must be at least 4 KB in size and at least as large as the Flash device sector erase size. For example, if the Flash has a 64 KB erase size, programmers need to allocate at least 128 KB since when the update process erases the inactive area. Note that this is the smallest size it can erase.

### 7.2 Vendor Specific Component Capabilities (VSCC) Not Set Correctly

The ICH8, ICH9, and ICH10 use a single Flash component that is shared between the BIOS and other firmware components, such as Alert Standards Format (ASF) or Intel® Active Management Technology (Intel® AMT). If the network MAC in these components is also used (in conjunction with the 82566 or 82562V PHY), the EEPROM image for the network interface is also stored in its own partition called the gigabit region. The Boot Agent accesses this region just as it would the dedicated EEPROM on a NIC.

To allow more flexibility in selecting Flash components, the ICH8 stores the codes needed to operate the shared flash component in a reserved area. These are referred to as the VSCC. The ICH9 and ICH10 add additional flexibility by the use of two of these areas, referred to as the Upper Vendor Specific Component Capabilities (UVSCC) and Lower Vendor Specific Component Capabilities (LVSCC).

The values entered in the VSCC must be correct for the Flash component type plus other configurable attributes (block sizes, etc.). These values are set by the BIOS and are not controlled by the Boot Agent, but must be used by the Boot Agent to read and write to the Flash. If corruption occurs to the NVM as a result of a Boot Agent write operation, it is possible that the VSCC values are not set correctly. Please refer to the *SPI Programming Guide* for the ICH8, ICH9 or ICH10 for more details.



**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.
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 or 82562 in combination with the ICH8 or ICH9, 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 ICH9 SPI Flash Programming Guide for more details.
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.



Message Text	Cause/Comments
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.
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. This should only happen on split ROM configurations.</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.</p>
PXE-EC9: PXENV + structure was not found in UNDI driver code segment.	The Boot Agent could not locate the needed PXENV+ structure. An incorrect Flash image is installed or the image has become corrupted. This should only happen on split ROM configurations.
PXE-M0F: Exiting Intel Boot Agent.	Ending execution of the ROM image.
This option has been locked and cannot be changed.	There was an attempt to change a configuration setting that has been locked by IBAUtil. 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.
PXE-M0E: Retrying network boot; press ESC to cancel.	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. The continuous retry feature is controlled by boot agent word 0x31 (0x34 for second port of a two port device) bit 11 in the NVM and is disabled by default.