Argon Managed PC Boot Agent (MBA)

TECHNICAL REFERENCE: Integrating Argon MBA into a PC BIOS

February 9, 2000

The Argon Managed PC Boot Agent (MBA) is a package of multi-protocol preboot firmware and software tools that enables the network administrator to remotely administer software, operating systems, and applications over the network, eliminating the need to visit the client's desktop in person. MBA supports multiple boot protocols and network environments such as traditional TCP/IP, NetWare, and RPL; also includes support for all of today's most used protocols including DHCP, BOOTP, RPL, NCP/IPX (802.2, 802.3, Ethernet II), and the Wired for Management (WfM) 2.0 specification Preboot Execution Environment (PXE).

In addition to being integrated on a NIC, MBA client firmware can also be integrated into a PC's BIOS. BIOS integration is required for LAN On Motherboard (LOM) implementations but also can be done when a discrete NIC will be installed to reduce the total NIC cost by eliminating the need for another boot ROM part on the NIC.

This document will explain how MBA can be integrated into a PC BIOS and assumes a good knowledge of BIOS engineering.

Contents

1. MBA Architecture

MBA is comprised of two WfM-compliant components—the MBA base code and the Universal Network Device Interface (UNDI).

The MBA base code handles all the core functionality of MBA such as protocol handling, network boot failure handling, etc. and is independent of the type of NIC being used. The base code also includes the PCI Loader code, which is responsible for initializing MBA, preparing UMB and saving initialization parameters for later use.

The UNDI handles the interface with the specific NIC hardware. It provides NIC initializations, low-level transmits and receive functionality, etc. The MBA base code uses the UNDI to interface with the NIC. Other base code modules could also use the UNDI to establish low-

level transmit and receive functionality with the specific NIC.

It is the UNDI that is the boot (or IPL) device. When executed, the UNDI can scan UMB for PXE-compliant base code (such as the MBA base code) or it may use its own base code.

2. General Integration Notes

Argon Technology will provide the necessary binary file(s) to the BIOS Engineers. A special build of MBA firmware is required for BIOS integration and the BIOS Engineer must indicate that a BIOS integration is intended. The difference between a build intended for BIOS integration vs. a build intended for a boot ROM on a NIC lies in the PCI Loader code. The PCI Loader for a BIOS-integration version of MBA must keep the whole MBA code in UMB if Post Memory Manager (PMM – see related section) is not supported in the BIOS.

When integrating MBA into the BIOS, BIOS Engineers typically compress the MBA firmware so it will require less BIOS ROM space. The compression mechanism is irrelevant to MBA and the mechanism usually used to include PCI option ROMs into the BIOS can be used. During autoscan time the BIOS must decompress the MBA firmware and load it into UMB space.

3. Monolithic vs. Split

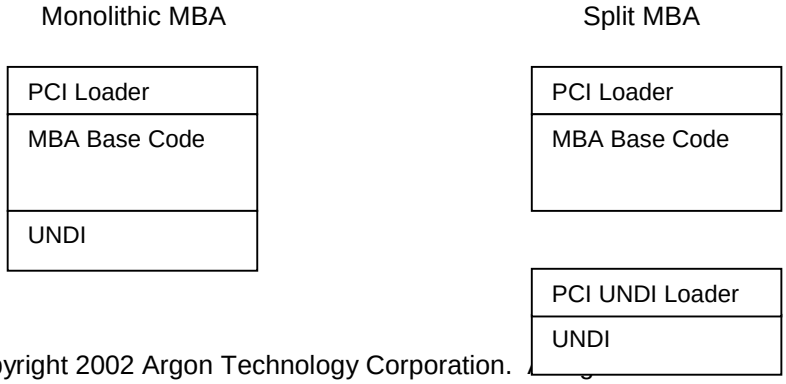
The concept of Monolithic and Split Option ROMs was introduced with the PXE 2.0 specification. For Monolithic ROMs, the option ROM header encapsulates all MBA components as a single entity.

In a Split ROM configuration, the UNDI is separated from the MBA base code resulting in two separate entities. This configuration has a few benefits. First, the UNDI can be used independently by other system components to provide low-level communication functionality via the NIC. For example, the BIOS itself may use the UNDI to transmit alerts to a management console if errors occur during the PC's boot process.

Another benefit to the Split ROM configuration is that of code re-use. If more than one NIC can be installed into a PC, using a monolithic ROM for each NIC means the base code is redundantly replicated in the system. Also, these base codes can be from different vendors and at various version levels resulting in performance and/or behavioral differences depending on which NIC is used. However, with a Split ROM implementation, the common base code can exist once in the BIOS and the NIC-specific UNDIs can bind with it and use it when they are executed. This eliminates code redundancy and provides a consistent base code for each NIC's UNDI.

MBA firmware is available as both Monolithic and Split ROMs, either of which can be integrated into the PC BIOS.

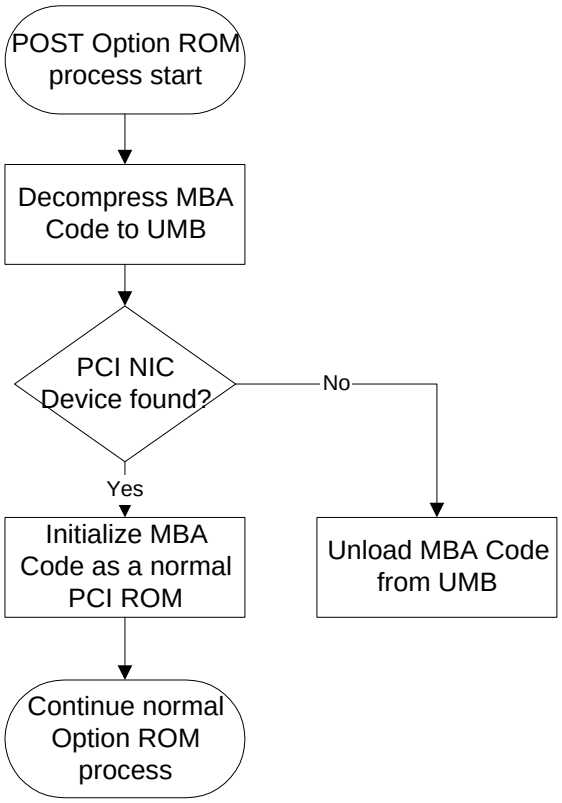
Refer to the PXE Specification for more details regarding Monolithic vs. Split ROM configurations.



4. Integrating Monolithic MBA

In this case of a Monolithic MBA, there is only one binary that needs to be integrated into the BIOS. This binary consists of all the MBA components, i.e. Base Code and UNDI. The Base Code is not visible by other UNDIs that may be in the PC (e.g. on a NIC) so that it is impossible for another UNDI to find the Base Code and bind to it at boot time.

The integration of a Monolithic MBA is straightforward. The binary is a standard PCI option ROM binary. The PCI Vendor ID and PCI Device ID is located in the PCI header structure and should be used by the BIOS to match MBA with its supported NIC. If the supported NIC does not exist in the system, the MBA binary should not be initialized. The following flowchart illustrates this procedure.



5. Integrating Split MBA

In the case of Split MBA, there are multiple binaries that need to be integrated into the BIOS. One binary is the generic MBA Base Code. When loading the MBA Base Code, the typical matching of PCI Vendor ID and Device ID to an installed NIC is not required. However, it is still a PCI option ROM and should be initialized as such.

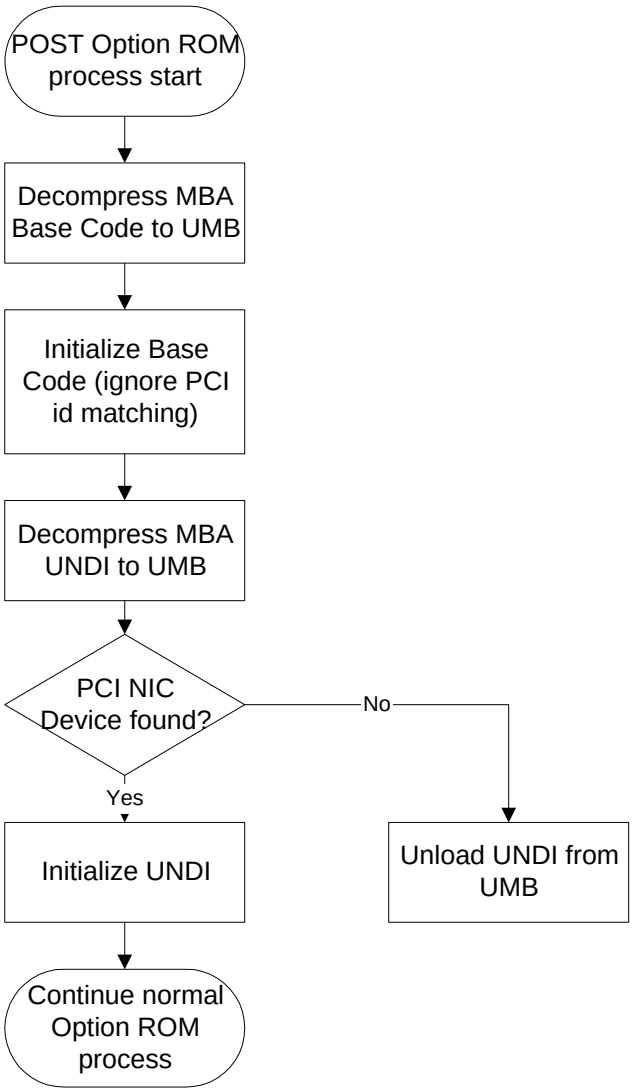
The other binaries are the UNDIs for each NIC to be supported by this BIOS integration, i.e. if only one NIC is to be supported then only the UNDI binary for that NIC should be included; and, if

multiple NICs are to be supported then the UNDI binaries for each NIC should be included. Since the UNDI binaries are specific to a particular NIC, the standard PCI Vendor ID and Device ID matching procedure must be followed to determine whether it should be initialized.

The UNDIs are the actual boot (or IPL) devices. For example, if the BIOS is BBS compatible (see related topic on BBS), the UNDI will appear as the bootable device in the boot order menu. When an UNDI is executed, it will scan UMB space looking for a PXE-compatible Base Code, which the MBA Base Code is. If found, the UNDI will bind and use this Base Code.

UNDIs themselves may not have to be integrated into the BIOS. Consider the scenario where the MBA Base Code is integrated into the BIOS along with the UNDI for NIC X and say another NIC, NIC Y, is installed in the PC. If NIC Y contains an UNDI in its boot ROM and that UNDI is executed as the boot device, it will find and bind with the Base Code from the BIOS.

The following flowchart illustrates how to integrate Split MBA into the BIOS.



The loading order of the MBA Base Code and UNDIs does not matter. It is not required that the Base Code be loaded and initialized before any UNDIs. As a matter of fact, the process shown in the flowchart above can be altered such that the UNDIs are processed first and the Base Code is only loaded if an UNDI was loaded. This prevents unnecessarily loading the Base Code if there is no PCI Vendor ID and device ID match between any UNDI and installed NICs.

6. BBS vs. Non-BBS

The BIOS Boot Specification (BBS) describes a methodology by which the BIOS will identify all IPL (Initial Program Load) boot devices in a system, allow the user to prioritize them in a desired boot order, and then sequentially attempt to boot from each device when the PC is powered on. BBS defines a boot scheme that is generic and flexible enough to allow booting from virtually any existing IPL device and for the definition of future IPL devices as well.

MBA complies with the latest BBS version 1.01 and the MBA UNDI is an IPL device. When initialized on a PC that has a BBS compatible BIOS, MBA will detect that BBS is supported and will not hook any boot vectors (INT 18H or INT 19H). When MBA completes its initialization and returns control to BIOS, the BBS BIOS stores MBA's Bootstrap Entry Vector (BEV) in its boot order menu. This allows users to specify the position of MBA in the boot order with respect to the other boot devices in the menu. If the PC's BIOS is not BBS-compliant, MBA always attempts to control the boot process by hooking both boot vectors (INT 18H and INT 19H).

By default, MBA attempts to detect whether BBS is supported by the BIOS and performs the actions described about. However, these MBA actions are configurable and can be forced by a user. For example, the user may force MBA to always hook the boot vectors regardless of whether BBS is supported or not.

7. Post Memory Manager

When integrating MBA into a BIOS it is strongly recommended that the BIOS support Post Memory Manager (PMM). When PMM is available, both Monolithic MBA and Split MBA can reduce their UMB footprint to 2KB. However, if PMM is not available, the footprint will be large. The actual amount varies between MBA implementations for various NICs however, as an example, without PMM Monolithic MBA may consume approximately 57KB of UMB and a Split MBA Base Code will consume approximately 44KB of UMB plus the UNDI's amount of approximately 16KB.

8. Storing MBA Configuration Data

MBA has many options that can be configured by the user such as the boot protocol, network boot failure behavior, etc. MBA requires 32-bits of non-volatile storage space to save these configuration settings. Typically MBA stores this configuration data in EEPROM located on the NIC. Therefore, if a LOM implementation keeps the NIC's EEPROM intact, the MBA that is integrated into the BIOS can continue storing the data as normal.

Since EEPROM implementations and storage locations for MBA data vary among various NICs, the MBA code that reads and writes the configuration data is NIC-specific. Most of the MBA options pertain to functionality available in the MBA Base Code. Therefore, the MBA Base Code must contain this NIC-specific code to access the configuration data from EEPROM. When Argon Technology provides a MBA Base Code binary for the BIOS integration, it will include the NIC-specific configuration code for the particular NIC that is intended to be supported by the PC. If the intended NIC is used, the MBA code will be able to access the configuration data in EEPROM and a fully configurable MBA will be available. However, if another NIC is installed into the PC,

the MBA code will not be able to access the configuration data and MBA will not be configurable. In this case MBA will default to standard PXE-only functionality with no configurable options.

Another option is to provide 32-bits of CMOS space for MBA to store its configuration data. This has two main advantages. First, since the data does not reside on a NIC and instead is located in a generic system location, the MBA code will be able to access the data regardless of which NIC is installed into the PC. Therefore, a fully configurable MBA will be available regardless of which NIC is used. Second, typically the user interface to configure the MBA options is displayed and handled by the MBA Base Code. This can still be the case even when the data resides in CMOS; however, another option is now available in that the user interface to configure MBA options can be integrated with the BIOS Setup user interface. Therefore, the user can configure MBA options at the same time that you configure BIOS options with one consistent user interface instead of two different ones. To leverage the benefits of using CMOS to store MBA configuration data, the BIOS engineers must specify to Argon Technology which CMOS locations to be used and any access procedures. Argon Technology will then provide MBA binaries that are customized for this application.

9. Related Information

The following publications and sources of information may be useful to you:

Preboot Execution Environment (PXE), Version 2.1, Intel Corporation, 1999,
http://developer.intel.com/ial/wfm/wfmspecs.htm

Post Memory Manager, Version 1.01, Phoenix Technologies Ltd., Intel Corporation, 1997,
http://www.phoenix.com/products/specs.html

BIOS Boot Specification Version 1.01, Compaq Computer Corporation, Phoenix Technologies Ltd., Intel Corporation, 1996,
http://www.phoenix.com/products/specs.html

Argon Technology Corporation
7895 Tranmere Drive, Suite 201A
Mississauga, Ontario
L5S 1V9 Canada

www.ArgonTechnology.com