

Introduction to Aptio 4.x

Moving beyond BIOS with solutions
for UEFI & the Framework

Revision 1.30 - May 1, 2009



Copyright © 2009 American Megatrends, Inc.

All Rights Reserved.

American Megatrends, Inc.

5555 Oakbrook Parkway

Suite 200

Norcross, GA 30093 (USA)

This publication contains proprietary information which is protected by copyright. No part of this publication may be reproduced, transcribed, stored in a retrieval system, translated into any language or computer language, or transmitted in any form whatsoever without the prior written consent of the publisher, American Megatrends, Inc.

American Megatrends, Inc. retains the right to update, change, and modify this publication at any time, without notice.

For Additional Information

Call American Megatrends BIOS Sales Department at 1-800-828-9264 for additional information.

Limitations of Liability

In no event shall American Megatrends be held liable for any loss, expenses, or damages of any kind whatsoever, whether direct, indirect, incidental, or consequential, arising from the design or use of this product or the support materials provided with the product.

Limited Warranty

No warranties are made, either expressed or implied, with regard to the contents of this work, its merchantability, or fitness for a particular use. American Megatrends assumes no responsibility for errors and omissions or for the uses made of the material contained herein or reader decisions based on such use.

Trademark and Copyright Acknowledgments

All product names used in this publication are for identification purposes only and are trademarks of their respective Companies. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Intel is a trademark of Intel Corporation in the U.S. and other countries.

Revision Information

Date	Rev	Description of Changes
2007-11-29	0.50	First release for internal review, based on Aptio 4.6.x core
2007-12-05	0.60	Revisions based on marketing feedback. Revised text in Section 1.1 (corrected) & 2.8 (condensed). Changed graphics on cover page.
2007-12-10	0.70	Revisions based on marketing feedback to Sections 2.3, 2.4, 2.8, 3.2 & 3.7.
2007-12-11	0.80	Removed “Legacy CMOS for BIOS setup option storage” from Section 3.8.
2007-12-20	0.90	Changed references for “AMI Debug to EFI” to “AMI Debug for UEFI”. Updated screenshots on pages 8, 11 & 12.
2007-12-26	1.00	Published version
2008-01-24	1.10	Added links to training information at ami.com. Updated copyright information for 2008. Updated MMTool information. Revised use of trademarks for Microsoft & Intel references. Add UEFI logo & link to Section 2.1. Reformatted Section 2.2.
2008-06-10	1.11	Updated corporate address.
2009-04-02	1.20	Updated references to the Framework (change EFI to UEFI). Add AMI Debug Rx information. Add links to fast boot IDF 2008 presentation. Add VIA to supported silicon list.
2009-05-01	1.30	Updated AMI Debug Rx information.

Table of Contents

1	INTRODUCTION	5
1.1	Purpose of this document.....	5
1.2	Aptio 4.x Key Features	5
2	UEFI BACKGROUND.....	6
2.1	What is UEFI?.....	6
2.2	How Does UEFI Relate To BIOS.....	6
2.3	A Brief History of BIOS.....	6
2.4	From BIOS to UEFI.....	6
2.5	The Intel Platform Innovation Framework for UEFI	7
2.6	Platform Initialization (PI).....	7
2.7	Aptio 4.x Support for UEFI	7
2.8	AMI's Work with UEFI	7
3	MAJOR FEATURES OF APTIO 4.X	8
3.1	Standards Based Development	8
3.2	Single UEFI Core for Any Application	8
3.3	Supports Multiple Silicon Vendors.....	8
3.4	Modular Core Design.....	8
3.5	Complete UEFI Solution	8
3.6	Superior Development Tools	9
3.7	Template Based Porting	9
3.8	Compatibility Support Module (CSM).....	9
3.9	Transitional Support for BIOS Customers	10
3.10	Scalable Solution Using eModules and UEFI Drivers	10

4	APTIO 4.X DEVELOPMENT ENVIRONMENT	11
4.1	Visual eBIOS Development Environment (VeB)	11
4.2	AMI Debug for UEFI	13
4.3	AMI Debug Rx	13
5	APTIO 4.X DEPLOYMENT UTILITIES	15
5.1	AMI BIOS Configuration Program (AMIBCP)	15
5.2	Module Management Tool (MMTool)	15
5.3	DMIEdit	15
5.4	ChangeLogo.....	16
5.5	AMI Firmware Update (AFU).....	16

1 Introduction

1.1 Purpose of this document

AMI presents Aptio as the next-generation solution for BIOS. Aptio incorporates over 20 years of experience delivering AMIBIOS solutions while moving beyond legacy BIOS limitations. Aptio is a highly modular solution, portable across a variety of platforms. The Aptio driver model, based on Unified EFI (UEFI) & the Intel® Platform Innovation Framework for UEFI, delivers higher flexibility than legacy BIOS and provides new opportunities for applications in the pre-boot environment.

1.2 Aptio 4.x Key Features

- ❖ Standards Based: UEFI & the Intel Platform Innovation Framework for UEFI
 - UEFI IA32 & x64 bindings
 - Integration with Intel EFI Development Kit (EDK) & “TianoCore” drivers
- ❖ A Single UEFI Core for Any Application
 - Desktop, Mobile, Server, Embedded & Ultra Mobile PC (UMPC)
 - Validated solutions in multiple PC market segments
- ❖ Supports Multiple Silicon Vendors
 - Broad CPU & chipset support
 - Including Intel, AMD, NVIDIA & SiS platforms
 - Available on Intel MPG, ECPD & UMG reference platforms
- ❖ A Complete UEFI Solution
 - Aptio 4.x Core
 - Codebase optimized for size & boot time
 - Template-based platform for a consistent development experience across different silicon platforms
 - Development & Debug Tools:
 - Visual eBIOS (VeB)
 - AMI Debug for UEFI
 - Deployment Utilities:
 - Flash utilities for multiple operating systems
 - AMIBCP, MMTool & Change Logo for ROM image maintenance
 - DMIEDIT for SMIBIOS data management in manufacturing
- ❖ Smooth Migration from BIOS to UEFI with Aptio
 - Compatibility Support Module (CSM), based on AMIBIOS8, adds legacy BIOS interfaces to Aptio for existing OS applications
 - VeB, a common porting environment for AMIBIOS8 & Aptio
 - AMI worldwide engineering support, cross-trained for AMIBIOS8 & Aptio solutions

2 UEFI Background

2.1 What is UEFI?

The Unified Extensible Firmware Interface (UEFI) specification defines a modern model for the interface between operating systems and platform firmware. The interface consists of data tables that contain platform-related information, plus boot and runtime service calls that are available to the operating system and its loader. Together, these provide a standard, modern environment for booting an operating system and running pre-boot applications.

For more information on UEFI, visit www.uefi.org.



2.2 How Does UEFI Relate To BIOS

UEFI works to standardize two primary functions of the PC Basic Input/Output System (BIOS): firmware-to-OS interface and platform initialization. The UEFI Specification Working Group (USWG) creates the UEFI specification, describing a firmware-to-OS interface analogous to BIOS software interrupts and the BIOS data area (BDA). The Platform Initialization Working Group (PIWG) specifications are intended to promote interoperability between firmware components provided by different entities, such as silicon vendors and firmware vendors.

2.3 A Brief History of BIOS

The Basic Input-Output System (BIOS) of the original IBM PC/XT & PC/AT, while being a very small part of the entire system package, was key to the success of the PC architecture. The clean definition of the PC/AT BIOS gave companies the ability to create “clone” systems which quickly dominated the market. The availability of MS-DOS outside of IBM allowed applications to run equally well across different brands of beige-box “PC clones”. More companies entered the market, prices went down, and basic economics took over. The firmware interface IBM designed for 250,000 IBM PC/XT systems helped spark an industry.

While BIOS is an OS-independent firmware model, it is not a platform independent model. At the heart of the BIOS is the Intel 8086 software interrupt model, which as the name implies, relies heavily on compatibility the Intel 8086. Today’s x86 processors still boot and run in 8086 “real mode” instead of their x64 extension mode ... PC BIOS interfaces do not support booting in 32-bit or 64-bit mode.

2.4 From BIOS to UEFI

The need for systems to be compatible with the BIOS interface currently dictates how companies like Intel design their processors & chipsets. This problem became apparent to Intel when developing the 64-bit Itanium architecture. EFI was created as a way to separate the processor architecture from the OS-to-firmware interface. After its use on Intel Itanium products, EFI was migrated into the IA32 architecture.

UEFI is the successor to the Intel Extensible Firmware Interface (EFI). The UEFI Forum, a collaborative, non-profit corporation, was formed to promote and manage the UEFI specifications. AMI is a UEFI Promoter and on the board of directors.

2.5 The Intel Platform Innovation Framework for UEFI

The Intel Platform Innovation Framework for UEFI (also known as "the Framework") is a set of specifications developed by Intel to describe a product-strength firmware implementation based on EFI. While UEFI specifies the OS-to-firmware interface, the Framework specifies the structure used to build the firmware beneath the OS-to-firmware interface.

Aptio 4.x is based on the Framework, allowing firmware components to leverage a documented set of architectural protocols. This structure allows Aptio to incorporate code from Intel's "Tiano" implementation of the Framework, and also leverage the open source EFI Development Kit (EDK) distributed at tianocore.org.

2.6 Platform Initialization (PI)

While the UEFI Specification defines the interface between the operating system and option ROM, the Platform Initialization (PI) specifications describe how to construct the "initialization environment" ... which is a major role performed by today's BIOS.

PI takes the concepts first defined by Intel in the Framework and extends them into an industry supported specification. PI is one possible method for components to initialize the system and publish UEFI boot interfaces, and allows 3rd party drivers to easily extend UEFI platforms.

2.7 Aptio 4.x Support for UEFI

AMI adopted EFI and the Intel Platform Innovation Framework for UEFI as the basis for the first Aptio product. The Aptio 4.x core architecture adds support for the latest UEFI Form specifications.

2.8 AMI's Work with UEFI

As an early adopter of EFI and a founding member of the UEFI Forum, AMI sees UEFI as an evolution of BIOS. AMIBIOS8 introduced modern development concepts such as graphical development tools & modular components ... but the BIOS interface is restricted by 16-bit x86 architecture. AMI leverages the UEFI specifications to create Aptio, which introduces new opportunities for firmware developers.

The UEFI model allows AMI engineers to work more on feature support and less on compatibility issues. One example is booting from USB. An OS-loader with no native support for USB devices relies on the INT 13h BIOS interface, emulating devices as floppy disks or hard disks. For BIOS to support the myriad of USB storage devices, AMI engineers must focus on legacy emulation instead of developing new features.

With UEFI, product features can be ported cross-platform using a new architecture independent model. Moving away from the 16-bit BIOS model of the 1980's allows UEFI firmware products like Aptio to move the market forward.

3 Major Features of Aptio 4.x

3.1 Standards Based Development

By supporting UEFI & the Intel Platform Innovation Framework for UEFI, Aptio offers developers a clean next-generation implementation based on the latest firmware standards. Aptio supports UEFI IA32 & x64 bindings, and offers integration with Intel EFI Development Kit (EDK) & “TianoCore” drivers.

3.2 Single UEFI Core for Any Application

Aptio is developed with the same philosophy that AMIBIOS8 successful: “one core for any application.” The modular Aptio architecture allows the same stable core to be deployed for Desktop, Mobile, Server, Embedded & Ultra Mobile PC (UMPC) projects. Aptio offers validated UEFI solutions in multiple PC market segments.

The “one core” approach is why Aptio is available on a variety of Intel Customer Reference Board (CRB) platforms from the Mobile Platform Group (MPG), Embedded & Communications Processor Division (ECPD) & Ultra Mobile Group (UMG).

3.3 Supports Multiple Silicon Vendors

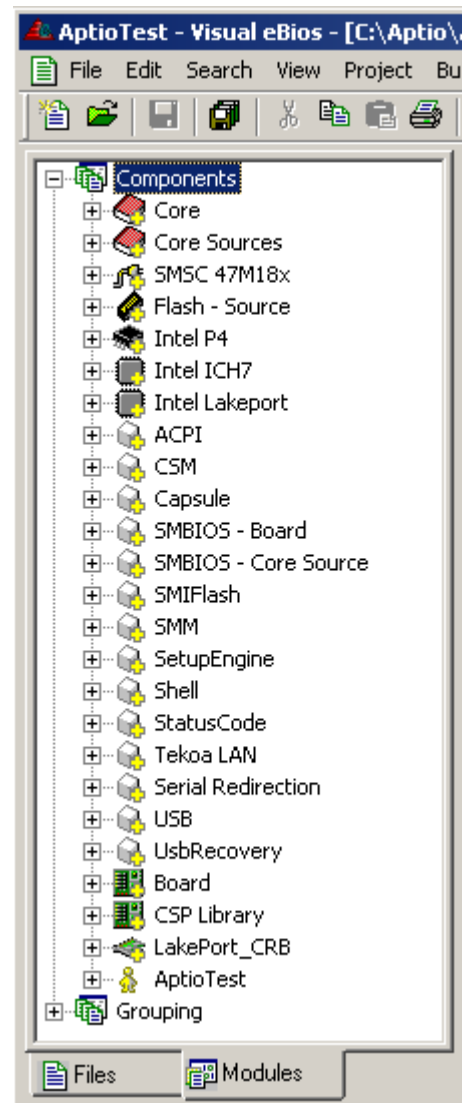
Aptio is based on AMI’s implementation of UEFI & the Framework, designed for easy porting to any vendor’s CPU or chipset design. Aptio offers broad CPU & chipset support including Intel, AMD, NVIDIA, SiS and VIA platforms.

3.4 Modular Core Design

By leveraging AMI’s Component Information Format (CIF) and System Description Language (SDL) technologies, Aptio developers combine UEFI concepts with AMI’s eModule design. Board level changes are separated from silicon & technology enabling modules. Customers can leverage eModules to quickly port OEM changes between platforms.

3.5 Complete UEFI Solution

Aptio is offered as a complete UEFI solution ... including an optimized codebase, development tools, deployment utilities, training programs and worldwide engineering support. The Aptio solution provides all of the pieces customers need to migrate platforms from legacy BIOS to UEFI.



3.6 Superior Development Tools

As with AMIBIOS8, Aptio uses the Visual eBIOS (VeB) development environment. VeB is built specifically for BIOS & firmware porting. Section 4 of this document describes the many advantages VeB offers to Aptio developers.

3.7 Template Based Porting

AMI has used “template-based porting” with AMIBIOS8 for a consistent development experience across different silicon platforms. Years of experience have brought this same template-based model to Aptio. This allows customers to easily isolate board-level changes from chipset & core code.

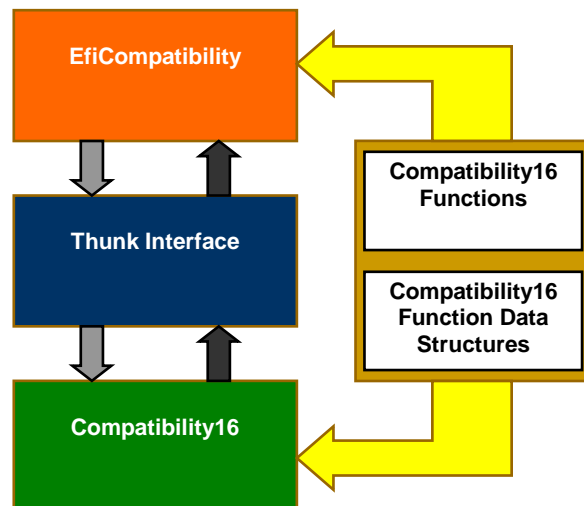
3.8 Compatibility Support Module (CSM)

UEFI removes legacy BIOS interfaces used by today’s hardware & operating systems:

- ❖ Legacy Interrupt Interface (INT XXh)
- ❖ BIOS Data Area (40:0) & Extended BIOS Data Area (EBDA)
- ❖ BIOS Boot Specification (BBS) initialization
- ❖ Pointers to firmware data used by “legacy BIOS” (SMBIOS, ACPI, ...)
- ❖ Execute Option ROM for on-board & add-on devices
- ❖ Support Master Boot Record (MBR) OS loaders

While upcoming operating systems & utilities add support for UEFI, these legacy BIOS interfaces are required to run today’s software.

The Compatibility Support Module (CSM) adds support for legacy BIOS interfaces along side UEFI 2.x interfaces provided by Aptio. This gives customers the flexibility of running newer UEFI-compliant software along side MS-DOS, Microsoft® Windows®, Linux and other existing operating systems.



Like other Aptio components, the CSM is self-contained and easily removed. Projects that have no legacy requirements do not need to add the CSM. Projects using CSM support both UEFI & legacy OS while migrating away from legacy BIOS design.

The Aptio CSM is based on AMIBIOS8 for maximum compatibility with existing software. The standalone CSM16 (16-bit binary) module requires no porting, designed to interface with any silicon combination. OEM changes are ported as a standalone binary, implementing customer features without changing the CSM16 base code.

3.9 Transitional Support for BIOS Customers

Customers moving their platforms from legacy BIOS to Aptio will find a number of product features designed specifically for a smooth transition.

- ❖ Visual eBIOS (VeB): Used by AMIBIOS8 and Aptio
 - Existing AMIBIOS8 customers have no need to learn a new porting tool
 - New AMI customers start using UEFI with a purpose-built development environment
- ❖ Compatibility Support Module (CSM)
 - Allows UEFI firmware to boot legacy OS
 - Based on AMIBIOS8 for maximum compatibility
 - Easily removed if not required
- ❖ AMI Worldwide Engineering Support
 - Engineers transfer years of experience in BIOS porting to UEFI designs
- ❖ Support for a wide variety of silicon
- ❖ Training is available now

For more information on AMI training programs for Aptio, visit www.ami.com/aptio

3.10 Scalable Solution Using eModules and UEFI Drivers

Aptio uses eModules to create solutions that redefine market segments for AMI customers. The eModule solution doesn't lock customers into the traditional desktop, mobile or server models ... customers build the BIOS they need for the application.

A prime example of Aptio's flexibility is fast boot solutions. A demonstration of a sub-second boot on an Intel Atom processor platform was shown by AMI at [Intel Developer Forum Taiwan 2008](#) ("[Optimizing the Framework for Embedded Applications](#)"). The presentation uses nine simple BIOS optimizations to quickly take the Intel Atom N270 platform boot time from six seconds to under one second using Aptio 4.x ... creating a fast boot embedded solution using low-cost netbook silicon.

4 Aptio 4.x Development Environment

4.1 Visual eBIOS Development Environment (VeB)

Aptio 4.x projects use the same Visual eBIOS (VeB) environment utilized by AMIBIOS8™ since its release in 2001. VeB is a comprehensive tool that is used during all aspects of platform development.

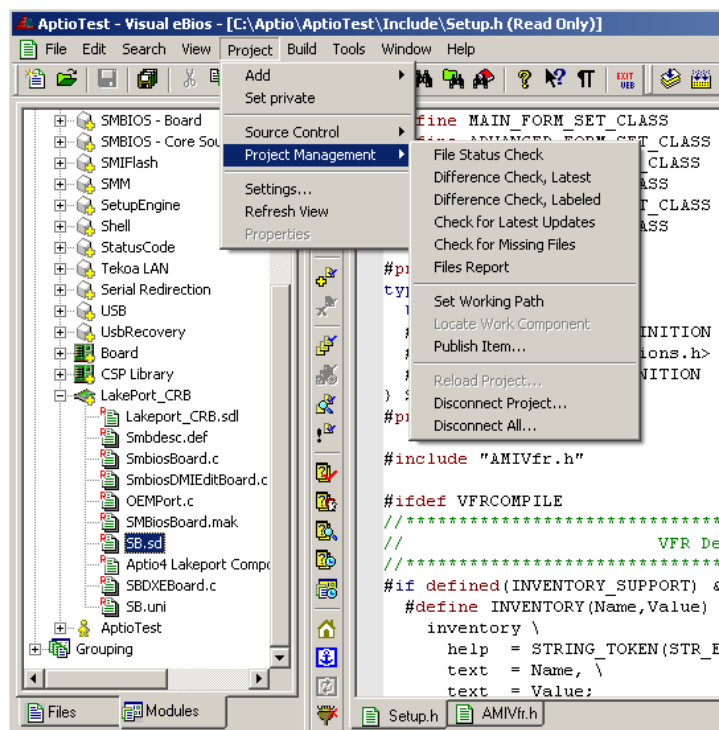
4.1.1 Project Management

VeB includes tools that allow an engineer to quickly define a new project based on a global component library. VeB provides a new project wizard that assists the engineer in pulling together components needed to form the project ... either based on reference projects or built from scratch for custom designs. Once a project has been created, additional components may be added or removed at any time.

4.1.2 Source Control

VeB contains integrated source control, with support for multiple formats:

- ❖ Visual Source Safe
- ❖ ClearCase
- ❖ PVCS
- ❖ Dimensions
- ❖ Subversion
- ❖ AMI Remote Source Control (RSC), providing customer access to AMI source databases 24/7/365



All standard source control operations can be done directly from VeB including check out, check in, get latest, add, remove, history, and difference. VeB provides a more convenient way to manage source.

4.1.3 Source File Views

VeB allows the engineer to view a project two different ways:

1. A tree of files organized exactly as they are in the project's directory tree.
2. A tree of files organized by BIOS component.

This very powerful feature allows the engineer to quickly navigate through the source and find the code easily. The component view also gives a better understanding to what is happening when BIOS components are added to or removed from a project.

4.1.4 Built in Editor

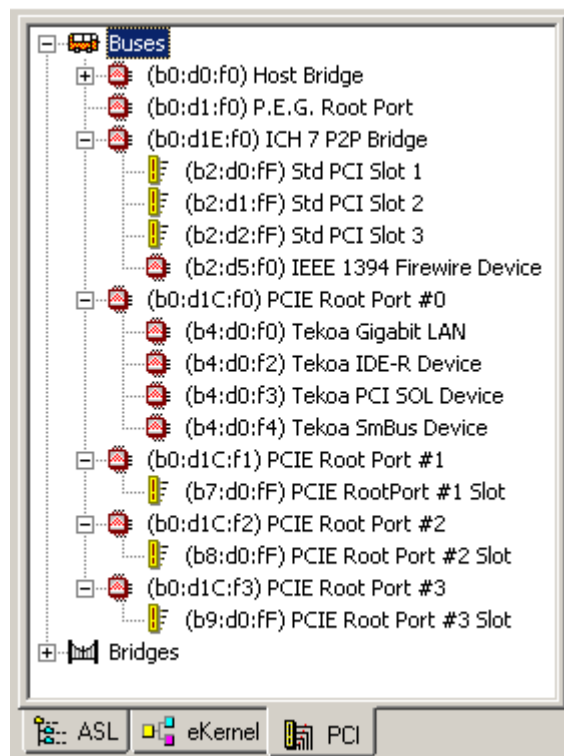
VeB includes a built in editor. VeB's editor supports syntax highlighting and many other features available with popular source code editors. The user may configure VeB to use its internal editor or to launch an external editor of the user's choice when opening a file. The VeB editor may be customized to use any hot key combination for common editing tasks. This allows the editor to emulate other popular editors.

4.1.5 PCI Wizard

VeB includes the PCI Wizard for graphical IRQ routing. This gives engineers working with schematics a more intuitive way to customize the IRQ routing for a platform. This screen also allows hardware engineers to verify the IRQ routing information without having to understand the syntax of IRQ routing tables.

The information that is gathered by the IRQ Routing Wizard generates several sections of source code:

- ❖ PCI BIOS IRQ routing table
- ❖ Microsoft IRQ routing table
- ❖ ACPI IRQ routing information
- ❖ MultiProcessor (MP) table IRQ information



VeB also allows IRQ information to be organized by bus hierarchy using the PCI tab of the Tools panel. This view is useful for systems with complex PCI bus configurations.

4.1.6 Project Parameters & System Description Language (SDL)

VeB includes a screen that allows the engineer to easily view and edit the setting of BIOS parameters. BIOS parameters are analogous to the make flags used by previous AMIBIOS® releases. Each BIOS component includes an SDL file. SDL files define the parameters that are relevant to each BIOS component. Using VeB's BIOS parameter screen, it is not necessary to directly modify make files.

4.1.7 Project Build

A simple mouse click from within VeB builds the BIOS and displays any error messages that result from the build. Build output is displayed in VeB's lower pane. Double

clicking on an error message in the window causes the related file to be opened in VeB's editor.

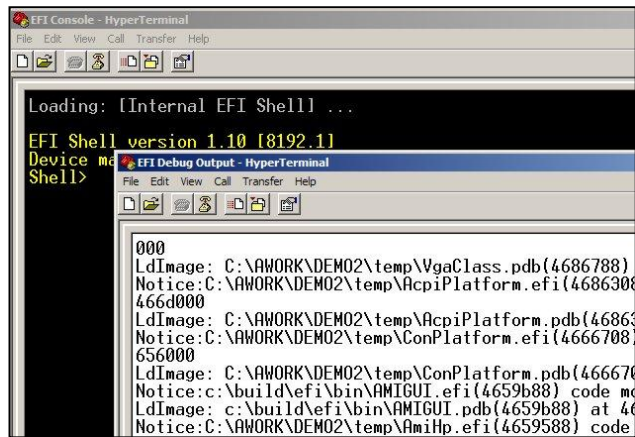
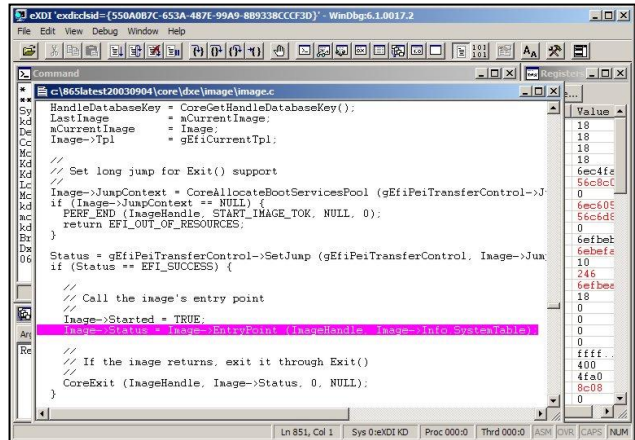
4.1.8 Context Sensitive Help

VeB includes a comprehensive help system for all chipset and OEM hooks. Pressing 'F1' while the cursor is located on the name of a hook function activates help for that function. Help includes a description of what the hook should do, who calls it, inputs and outputs, etc. Over time help may also include helpful hints or common pitfalls related to particular chipset or OEM hooks.

4.2 AMI Debug for UEFI

AMI Debug for UEFI uses a flexible host/target configuration. The debug target communicates with a Microsoft Windows host application via RS-232 serial or USB 2.0 debug port. Originally developed as a debugger for EFI Shell applications, AMI Debug has been extended to support UEFI debugging in PEI, DXE, SMM & EFI Shell environments.

- ❖ Symbolic source-level debugging
- ❖ View & edit memory/PCI/IO
- ❖ Add breakpoints
- ❖ Redirect console
- ❖ Redirect debug messages
- ❖ Remove when not required
- ❖ Does not use ICE/ITP/HDT/JTAG
- ❖ Lower cost to deploy
- ❖ Use with production hardware
- ❖ Use with any processor or chipset



4.3 AMI Debug Rx



AMI Debug Rx™ is the first of its kind: a low-cost debug tool built around the debug port feature common to today's USB 2.0 EHCI controllers. Perfect for today's embedded & netbook platforms, this product is targeted to power users, quality assurance labs & service technicians. Diagnosing small form factor platforms with AMI Debug Rx is non-intrusive, allowing technicians to access checkpoints without opening the case.

AMI Debug Rx is designed as replacement for the PCI ‘POST Checkpoint Card,’ which is becoming less useful in the PC market as newer systems omit the traditional PCI expansion slot.

Based on patent-pending technology, AMI Debug Rx replaces the POST checkpoint card’s 7-segment LED with a small LCD screen. This debug method produces more descriptive debugging messages than the checkpoint card, along with including boot performance timing and UEFI debug message redirection.

AMI Debug Rx Feature Set:

- ❖ Low-cost alternative to the PCI “POST Checkpoint Card”
- ❖ Low cost device for platform development or field diagnostics
- ❖ Device records checkpoints, UEFI debug messages and timing data for measuring boot performance
- ❖ Session data can be captured and stored to one of four “sessions” for later review
- ❖ Display descriptive text for each checkpoint: based on AMIBIOS8, Aptio 4.x or a user-provided string table
- ❖ AMI Debug Rx protocol can be ported into existing AMIBIOS8 and Aptio 4.x BIOS projects
- ❖ Works with AMI Debug for UEFI™ to enable source-level BIOS debugging
- ❖ Designed for BIOS developers, quality assurance testing and field diagnostics.



AMI Debug Rx is scheduled to ship in the second half of 2009. For more information, contact an AMI Sales Associate.

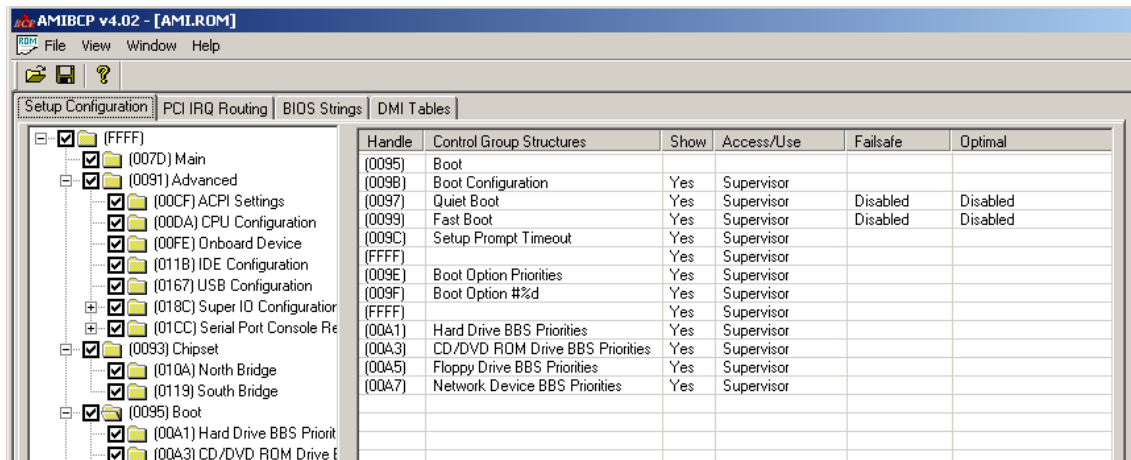
5 Aptio 4.x Deployment Utilities

AMI designs a variety of deployment utilities for Aptio, simplifying management of customer firmware images.

5.1 AMI BIOS Configuration Program (AMIBCP)

AMIBCP allows customers to modify common options in firmware ROM images without rebuilding the project from source code:

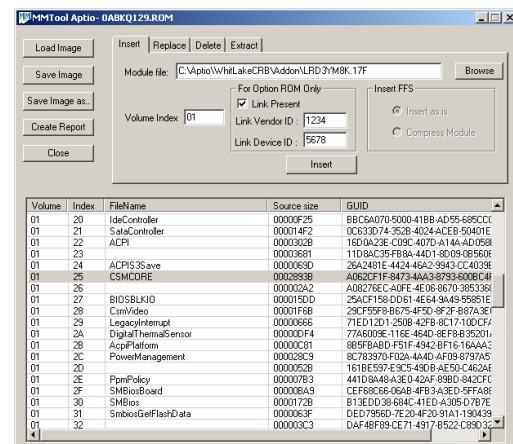
- ❖ Modify setup option defaults, multi-language strings & static SMBIOS Data
- ❖ Change setup option default values
- ❖ Change Unicode strings in setup & sign-on messages
- ❖ Edit static SMBIOS data



5.2 Module Management Tool (MMTool)

MMTool allows customers to manage the drivers and modules that construct an Aptio firmware image.

- ❖ Add, delete, replace & extract components without rebuilding from source
- ❖ Operates on PEI/DXE drivers & legacy Option ROM
- ❖ Command line interface available for automation in manufacturing environments



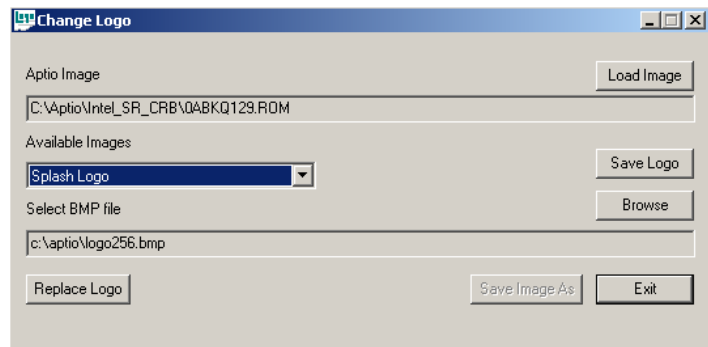
5.3 DMIEdit

DMIEdit is designed to modify platform-specific SMBIOS data in manufacturing environments. SMBIOS UUID, serial numbers, asset tags & OEM strings can be

modified using this script-driven utility. DMIEdit runs in DOS, Microsoft Windows & EFI Shell environments (32-bit & 64-bit).

5.4 ChangeLogo

ChangeLogo is Aptio's OEM logo customization utility. A platform's look & feel can quickly be changed by altering the "silent boot" logo embedded in the ROM image. ChangeLogo can also be used to extract logos from other Aptio firmware images.



5.5 AMI Firmware Update (AFU)

AFU allows the system ROM to be upgraded from multiple operating environments.

- ❖ Supported in DOS, EFI Shell, Windows, Linux and FreeBSD
- ❖ Command line utility - x86 & x64
- ❖ GUI version available for Microsoft Windows