

AMIBIOS ROM Utilities User Guide (Version 1.08)

American Megatrends, Inc.

6145-F Northbelt Parkway

Norcross, GA 30071, USA

This publication contains proprietary information, which is protected by copyright. No part of this publication can be reproduced, transcribed, stored in a retrieval system, translated to 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, modify this publication at any time, without notice.

Limited Warranty

No warranties are made, either express 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.

Limitations of Liability

In no event shall American Megatrends be held liable for any loss, expenses, or damages of any kind whatsoever, whether direct, special, indirect, incidental, or consequential, arising from or arising out of the use or inability to use the contents of this manual.

Trademarks

American Megatrends acknowledges the following trademarks:

Windows® 95, Windows® 98, Windows® ME, Windows® NT, Windows® 2000 and Windows® XP are trademarks of Microsoft® Corporation.

Other trademarks and trade names may be used in this document to refer to either the entities claiming the marks and names or their products. American Megatrends, Inc. disclaims any proprietary interest in trademarks and trade names other than its own.

Disclaimer

This manual describes the operation of the AMIBIOS ROM Utilities. Although efforts have been made to insure the accuracy of the information contained here, American Megatrends expressly disclaims liability for any error in this information, and for damages, whether direct, indirect, special, exemplary, consequential or otherwise, that may result from such error, including but not limited to the loss of profits resulting from the use or misuse of the manual or information contained therein (even if American Megatrends has been advised of the possibility of such damages). Any questions or comments regarding this document or its contents should be addressed to American Megatrends at the address shown on the cover.

American Megatrends provides this publication "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability or fitness for a specific purpose.

Some states do not allow disclaimer of express or implied warranties or the limitation or exclusion of liability for indirect, special, exemplary, incidental or consequential damages in certain transactions; therefore, this statement may not apply to you. Also, you may have other rights which vary from jurisdiction to jurisdiction.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. American Megatrends may make improvements and/or revisions in the product(s) and/or the program(s) described in this publication at any time. Requests for technical information about American Megatrends products should be made to your American Megatrends authorized reseller or marketing representative.

Revision History

Date	Ver.	Description	Editor			
08/23/2004	1.00	- Initial Document.	Danny Liu			
		- This version contains AMIMMDOS.EXE, AMIOLDOS.EXE and				
		ROMSETUP.EXE user guide.				
10/04/2004	1.01	- Add new user guides for AMIDEDOS.EXE, AMIDEWIN.EXE,	Danny Liu			
		DMIEDIT and OEMLOGO.EXE.				
11/03/2004	1.02	- Update for AMIMMDOS.EXE and AMIOLDOS.EXE.	Danny Liu			
11/16/2004	1.03	- Add new user guides for AFUDOS.EXE, AFUWIN.EXE	DannyLiu			
12/29/2004	1.04	- AMIDEDOS, AMIDEWIN and DMIEDIT supports SMBIOS	DannyLiu			
		spec up to 2.4.				
		- Add new user guides for AMISCE, AMISCEW, AMICMOS,				
		MMTOOL and AMIBCP.				
02/05/2005	1.05	- Correct command syntax for AMIMMDOS.	DannyLiu			
		- Update BIOS requirement for AMIDEWIN				
	- Update BIOS requirement for DMIEDIT.					
		- Add new example for AMISCE.				
		- Add new description for AFUDOS's "/Ln" option.				
	- 16. /	- Corrects rules, example and New Logo file Requirements for				
	<u>NIN</u>	AMIOLDOS. S KOM UTITIES				
		- Corrects New Logo file requirements for OEMLOGO.				
04/04/2005	1.06	- Add comments for MMTOOL – ROMInfo, ROMHole,	DannyLiu			
		CPUPatch.				
		- AMIMMDOS support /SM command to modify Sign-On				
		message.				
		- Add comment for AMIBCP - SETUP screen layout.				
		- Update comments for OEMLOGO.				
		- SMIFLASH eModule MUST be "8.00.00_SMIFlash-1.00.07"				
		label or later				
05/04/2005	1.07	- Add new comments for AMIMMWIN.EXE, AMIOLWIN.EXE.	DannyLiu			
06/04/2005	1.08	- Add user interface manual for AFUWIN.EXE.	DannyLiu			

	-
Chapter 1 AMIMMDOS/WIN v1.xx	9
Overview	9
Features	9
Requirements	9
Supported Operating System	9
BIOS Requirements	10
Getting Started	10
	10
Usage	10
	12
hapter 2 AMIOLDOS/WIN VI.xx	14
Overview	14
Features	14
Requirements	14
Supported Operating System	14
BIOS Requirements	14
New Logo File Requirements	13 1 <i>5</i>
	13 1 <i>5</i>
Installation	13 1 <i>5</i>
Usage	15 1 <i>6</i>
Examples	10
Overview	10
Eastures	10
Pequirements	10
Supported Operating System	10
BIOS Requirements	10
Getting Started	۱۵ ۱۷
Installation	۱۵ ۱۷
Running ROMSETUP program	۱۵ ۱۷
Shanter 4 AMIDEDOS v1 vy	10 71
	∠⊥ 21

Features	21
Requirements	21
Supported Operating System	21
BIOS Requirements	21
Getting Started	21
Installation	21
Usage	21
Examples	24
Chapter 5 AMIDEWIN v1.xx	26
Overview	26
Features	26
Requirements	26
Supported Operating System	26
BIOS Requirements	26
Operating System Driver Requirements	26
Getting Started	27
Installation	27
Usage & Example	27
Chapter 6 AFUDOS v4.xx	28
Overview	28
Features	28
Requirements	
Supported Operating System	
BIOS Requirements	
Getting Started	
Installation	
Usage	
Examples	30
Chapter 7 AFUWIN v4.xx	32
Overview	32
Features	32
Requirements	32
Supported Operating System	32
BIOS Requirements	32
Operating System Driver Requirements	32
Getting Started	
Installation	
Usage & Example for command line mode	
Main Window	
Buttons	34

Function Frame	
Functions	37
Saving system BIOS ROM image to file	
Flashing system BIOS with given file	
Chapter 8 AMISCE v1.xx/v2.xx	39
Overview	
Features	
Requirements	
Supported Operating System	
BIOS Requirements	
Getting Started	
Installation	
Usage	
Examples	43
Chapter 9 AMISCEW v1.xx/v2.xx	44
Overview	44
Features	44
Requirements	44
Supported Operating System	44
BIOS Requirements	44
Operating System Driver Requirements	44
Getting Started	45
Installation	45
Usage & Example	45
Chapter 10 AMICMOS v2.xx	46
Overview	46
Features	46
Requirements	46
Supported Operating System	46
BIOS Requirements	46
Getting Started	46
Installation	46
Llange	1.5
Usage	46
Examples	46 47

Part 2.	Graphical User Interface Mode	48

Chapter 1 OEMLOGO v3.xx	49
Overview	49
Features	49
© Copyright 2005 American Megatrends, Inc. All rights reserved.	- 5 -

Requirements	49
Supported Operating System	49
BIOS Requirements	49
New Logo File Requirements	49
Getting Started	
Installation	
Buttons	
Options	51
Functions	51
Changing OEM Logo	
Changing Small Logo	53
Chapter 2 MMTOOL v3.xx	55
Overview	55
Features	55
Requirements	55
Supported Operating System	55
BIOS Requirements	55
Getting Started	56
Installation	56
Main Window	56
Main Buttons	56
Function Frame	57
Module Info Frame	63
Functions	63
Inserting Generic Module	64
Inserting Linked Module	66
Replacing Module	66
Deleting Module	67
Extracting Module	68
Inserting image into a ROM Hole	69
Saving ROM Hole image to file	70
Deleting ROM Hole image	71
Inserting a patch data	71
Extracting a patch data	72
Deleting a patch data	73
Saving changes & Exiting	74
Chapter 3 AMIBCP v3.xx	75
Overview	75
Features	75
Requirements	75

Supported Operating System	75
BIOS Requirements	75
Getting Started	76
Installation	76
Main Window	76
Menu Bar	76
Toolbar	79
Body Frame	79
Status Bar	79
Functions	80
Setup Configuration Tab	81
Register Edit Tab	84
PCI IRQ Routing Tab	85
BIOS String Tab	87
BIOS Features Tab	89
DMI Tables Tab	91
Chapter 4 DMIEDIT v1.xx	93
Overview	93
Features	93
Requirements	93
Supported Operating System	93
BIOS Requirements	93
Operating System DLL/Driver Requirements	93
Getting Started	94
Installation	94
Main Window	94
Menu Bar	94
Toolbar	97
Type Frame	98
Info Frame	98
Status Bar	98
Functions	99
Browsing SMBIOS information	99
Saving SMBIOS information to file	99
Saving all SMBIOS information to file	100
Updating SMBIOS type	101
Updating all SMBIOS type	103
Undoing current type and Undoing all	104
Appendix A Module ID Codes	105
Appendix B AFUDOS v3.xx Commands	106

Part frican Command Mode

Chapter 1 AMIMMDOS/WIN v1.xx

Overview

AMIMMDOS is a module management tool with command line interface. Basically, it allows you to manage the BIOS modules that are contained in the BIOS ROM file.

Features

This utility offers the following features:

- Insert Module
- Replace Module
- Delete Module
- Extract Module
- Read/Write SLP String
- Fill ROM Hole with file
- Display/Save CPU MicroCode Patch information
- Delete CPU MicroCode Patch data
- Extract CPU MicroCode Patch data from ROM Image to file
- Insert new CPU MicroCode Patch data to ROM Image
- Read/Write Sign-On Message

User Guide

Supported Operating System

Requirements

AMIMMDOS is supported by the following operating systems:

- MS-DOS environment
- Microsoft® Windows® 98
- Microsoft® Windows® ME
- Microsoft® Windows® NT 4.0
- Microsoft® Windows® 2000
- Microsoft® Windows® XP

AMIMMWIN is supported by the following operating systems:

- Microsoft® Windows® 98
- Microsoft® Windows® ME
- Microsoft® Windows® NT 4.0
- Microsoft® Windows® 2000
- Microsoft® Windows® XP/XP64

Microsoft® Windows® PE •

BIOS Requirements

The loaded BIOS ROM file should have the followings:

- The file MUST be an AMIBIOS ROM file (Core version 8.xx.xx only)
- BIOS ROM file should be building via "8.00.08_AMITOOLS_17" label or above. •

Getting Started

Installation

Copies the AMIMMDOS.EXE executable file to any storage location accessible by the host system and then run AMIMMDOS in command prompt.

Usage

AMIMMDOS <BIOS ROM File Name> <Command> [Option 1] [Option2].....

BIOS ROM File Name

The mandatory field is used to specify path/filename of the BIOS ROM file with extension. **ROM** Utilities

Commands

.

The mandatory field is used to select an operation mode for module manipulation Or read/write SLP string. Following lists the valid commands and related format:

- /CP <Output File Name>
- /CPD <CMP ID>
- /CPE <CMP ID> <Output File Name>
- /CPI <New MicroCode File Name>
- /D <Module ID>
- /E <Module ID> <Output File Name>
- /I <Module ID> <Module File Name> .
- /INFO [Output File Name]
- /LM <L-VID> <L-DID> <VID> <DID>
- /R <Module ID> <Module File Name>
- /RHF <Hole Number> <Hole File Name> Fill ROM Hole with given file. .
 - /SLP [SLP String] /SM [Sign-On Message]

Read/Write SLP string from/to ROM Image. Read/Write Sign-On Message from/to ROM Image.

Display CPU MicroCode Patch

Delete a module.

Or save it to file.

Insert a linked-module.

information on screen Or save it to file.

Extract a CPU MicroCode Patch data to file.

Insert a new CPU MicroCode Patch data.

Extract a module as is in the ROM file.

Insert a module except for linked-module.

Display BIOS ROM Information on screen

Replace a module using uncompressed file.

Delete a CPU MicroCode Patch data.

Options

The optional field used to supply more information for individual operation mode. Following lists the supported optional parameters and format:

- /A <8-Bits Value>
- /C
- /M <M#1> <M#2>
- /N <NCB Name>
- /S <Start Address>
- /U

d optional parameters and format: Set alignment value. The module cannot be split.

Set destination address/PCI Device's

vendor/device ID. Set NCB region name.

Set start address

Specifies the module is to be placed as uncompressed.

	Parameters List					
Name	Description					
Module ID	2-digits hexadecimal Module ID. See <u>Appendix A Module ID Codes</u> for detail.					
Output File Name	This parameter is used to specify path/filename of the output file with extension.					
Module File Name	This parameter is used to specify path/filename of the Module file with extension.					
SLP String	If present, SLP WRITE function will be enabled.					
	If absent, SLP READ function will be enabled.					
Sign-On Message	If present, Sign-On Message WRITE function will be enabled.					
	If absent, Sign-On Message READ function will be enabled.					
Hole Number	This parameter MUST be a decimal value.					
Hole File Name	This parameter is used to specify path/filename of the Hole file with extension.					
CMP(CPU	This ID used for identifying CPU MicroCode Patch data in ROM Image. It has two					
MicroCde Patch)	expresions as below:					
ID	1. This is a number of CPU MicroCode Patch data in the module. 8-digits					
	decimal value. The number is starting from 1. The number can get by using					
	command.					
	2. This is CPU MicroCode Patch ID. It consists of a letter "" M " and 8-digits					
	hexadecimal ID as "Mxxxxxxx".					
	You can use command to get relative MicroCode number in ROM Image					
	Or ID code.					
New MicroCode	This parameter is used to specify path/filename of the new CPU MicroCode					
File Name	Patch file with extension.					
L-VID	4-digits hexadecimal linked-vendor ID.					
L-DID	4-digits hexadecimal linked-device ID.					
VID	4-digits hexadecimal vendor ID.					
DID	4-digits hexadecimal device ID.					
8-Bits Value	This parameter MUST be 2-digits hexadecimal value.					

Parameters List						
Name	Description					
M#1	This parameter MUST be 4-digits hexadecimal value.					
	For generic module, it is the module runtime Segment.					
	For module ID – 20h, it is the PCI ROM device ID.					
	For module ID – 21h, it is a Flags.					
M#2	This parameter MUST be 4-digits hexadecimal value.					
	For generic module, it is the module runtime Offset.					
	For module ID – 20h, it is the PCI ROM vendor ID.					
	For module ID – 21h, it is the Country Code.					
NCB Name	= EBB, The module placed in Extended Boot Block.					
	= NCB00, The module placed in first NCB region.					
	= NCB01, The module placed in second NCB region.					
	· ·					
	· ·					
	= NCBxx, The module placed in [xx]th NCB region.					
Start Address	8-digits hexadecimal starting address of the module in the ROM Image.					

Rules

- Any parameter encolsed by < > is a mandatory field.
- Any parameter enclosed by [] is an optional field.
- [/M] can be a condition to find out module in ROM Image exactly.
- [/A] & [/C] & [/N] & [/S] are valid only for </I> and </R> command.
- </D> command can use [/M] only.
- Both [/M] and [/U] are available for </E> command.
- [/N] should not co-exist with [/S]. By priority, [/N] < [/S].
- [/C] should not co-exist with [/S]. By priority, [/C] < [/S].
- All option is available only for $\langle D \rangle$, $\langle E \rangle$, $\langle I \rangle$ and $\langle R \rangle$ commands.

Note: Running AMIMMDOS under command prompt directly will display help message.

Examples

Examples on how to manipulate BIOS ROM image using the command prompt are shown in following:

- Display CPU MicroCode Patch module information
 AMIMMDOS <BIOS ROM File Name> /CP
- Save CPU MicroCode Patch module information to file
 AMIMMDOS <BIOS ROM File Name> /CP <Output File Name>

Delete a CPU MicroCode Patch data from ROM Image . AMIMMDOS <BIOS ROM File Name> /CPD <CMP ID> Extract a CPU MicroCode Patch data to file AMIMMDOS <BIOS ROM File Name> /CPE <CMP ID> <Output File Name > Insert a CPU MicroCode Patch data to ROM Image AMIMMDOS <BIOS ROM File Name> /CPI <New MicroCode File Name> **Deleting an existing module** AMIMMDOS <BIOS ROM File Name> /D <Module ID> [/M <M#1> <M#2>] Extracting a module AMIMMDOS <BIOS ROM File Name> /E <Module ID> <Output File Name> [/M <M#1> <M#2>] [/U]Inserting a new module AMIMMDOS <BIOS ROM File Name> /I <Module ID> <Module File Name> [/A] [/C] [/N <NCB *Name>*] [/*M* <*M*#1> <*M*#2>] [/*S* <*Start* Address>] [/*U*] **Inserting a linked-ID module** -AMIMMDOS <BIOS ROM File Name> /LM <L-VID> <L-DID> <VID> <DID> **Display BIOS ROM Information** ican AMIMMDOS <BIOS ROM File Name> /INFO Save BIOS ROM Information to file . AMIMMDOS <BIOS ROM File Name> /INFO <Output File Name> **Replacing an existing module** AMIMMDOS <BIOS ROM File Name> /R <Module ID> <Module File Name> [/A] [/C] [/N <NCB *Name>]* [/*M* <*M*#1> <*M*#2>] [/*S* <*Start* Address>] [/*U*] Insert a file to ROM Hole . AMIMMDOS <BIOS ROM File Name> /RHF <Hole Number> <Hole File Name> **Read SLP string from BIOS ROM File** AMIMMDOS <BIOS ROM File Name> /SLP Write SLP string to BIOS ROM File AMIMMDOS <BIOS ROM File Name> /SLP <"SLP string"> **Read Sign-On Message from BIOS ROM File** AMIMMDOS <BIOS ROM File Name> /SM Write Sign-On Message to BIOS ROM File AMIMMDOS <BIOS ROM File Name> /SM <"Sign-On Message">

Chapter 2 AMIOLDOS/WIN v1.xx

Overview

AMIOLDOS is a changing logo tool with command line interface. It allows you to replace the OEM Logo(Large) and OEM Logo(Small) module inside the BIOS ROM file with a new one.

Features

This utility offers following features:

- Change OEM/small logo.
- Remove OEM/small logo.

Requirements

Supported Operating System

AMIOLDOS Utility is supported by the following operating systems:

- MS-DOS environment
- Microsoft® Windows® 98
- Microsoft® Windows® ME
- Microsoft® Windows® NT 4.0
- Microsoft® Windows® 2000
- Microsoft® Windows® XP

AMIOLWIN is supported by the following operating systems:

- Microsoft® Windows® 98
- Microsoft® Windows® ME
- Microsoft® Windows® NT 4.0
- Microsoft® Windows® 2000
- Microsoft® Windows® XP/XP64
- Microsoft® Windows® PE

BIOS Requirements

The loaded BIOS ROM file should have the followings:

- The file MUST be an AMIBIOS ROM file (Core version 8.xx.xx only)
- BIOS ROM file should be building via "8.00.08_AMITOOLS_17" label or above.
- Large OEM Logo module (Module ID 0x0E) to be present

- Small OEM Logo module (Module ID 0x1A) to be present
- Quiet Boot function should be inside. It is recommended to use *DisplayLogo2 eModule* with "8.00.08_DISPLAYLOGO_05" label or later.

New Logo File Requirements

The Change OEM Logo Utility requires that the new Logo file fit the following format:

- 16-Color Bitmap format, even width, 640*480 pixels (Maximum)
- 256-Color Bitmap format, even width, 640*480 pixels (Maximum)
- 256-Color PCX format, even width, 640*480 pixels (Maximum)
- True-Color JPG format, even width, 640*480/800*600/1024*768 pixels (Maximum)

Note: Small OEM Logo does support only 640*80, 16-Color Bitmap format.

Getting Started

Installation

Copies the *AMIOLDOS.EXE* executable file to any storage location accessible by the host system and then run **AMIOLDOS** in command prompt.

Usage

AMIOLDOS <BIOS ROM File Name> <New Logo Image File Name> [Option]

Or

AMIOLDOS <BIOS ROM File Name>/D

BIOS ROM File Name

The mandatory field is used to specify path/filename of the BIOS ROM file with extension.

New Logo Image File Name

The mandatory field is used to specify path/filename of the new logo image file with extension.

Commands

The mandatory field is used to select an operation mode for manipulating logo image.

• /D Delete OEM Logo Module.

Options

The optional field used to supply more information for normal operation. Following

lists the supported optional parameters:

/FN

/S

- **/F** Force replacement even if the Logo format does NOT be matched.
 - Both [/F] and [/N] will be enabled.
- /N Insert 16-Color BMP without converting it to GRFX(AMI) format.
 - Change small OEM Logo.
- /XY <X> <Y> Set X-/Y-coordinate value.

For small logo...

X-coordinate ranged between 0 - 639. Default is 0.

Y-coordinate ranged between 0 - 79. Default is 0.

For large logo...

X-coordinate ranged between 0 - 639. Default is AUTO.

Y-coordinate ranged between 0 - 479. Default is AUTO.

ies

Rules

- Any parameter encolsed by < > is a mandatory field.
- Any parameter enclosed by [] is an optional field.
- Change large OEM Logo and check Logo format are default operation.
- Small Logo image will be converted to GRFX format automatically.

UD K

- [/D] MUST be used alone except [/S].
- [/N] and [/FN] option cannot co-exist with [/S].
- [/XY] is available only for GRFX logo format.

ъ

Note: Running AMIOLDOS under command prompt directly will display help message.

Examples

Examples on how to change large/small OEM Logo using the command prompt are shown in following:

- Replacement of large OEM Logo
 AMIOLDOS < BIOS ROM File Name> < New Logo Image File Name>
- Replacement of small OEM Logo
 AMIOLDOS <BIOS ROM File Name> <New Logo Image File Name> /S
- **Replacement of large OEM Logo(GRFX) with user defined X-/Y-coordinate** AMIOLDOS <BIOS ROM File Name> <New Logo Image File Name> /XY 10 10
- Replacement of small OEM Logo with user defined X-/Y-coordinate AMIOLDOS <BIOS ROM File Name> <New Logo Image File Name> /S /XY 10 10
- Deleting large OEM Logo
 AMIOLDOS <BIOS ROM File Name> /D
- Deleting small OEM Logo
 AMIOLDOS < BIOS ROM File Name> /D /S
- Force replacement of large OEM Logo

AMIOLDOS <BIOS ROM File Name> <New Logo Image File Name> /F

- Force replacement of small OEM Logo
 AMIOLDOS <BIOS ROM File Name> <New Logo Image File Name> /F /S
- Force replacement of large OEM Logo and do not convert to GRFX format AMIOLDOS <BIOS ROM File Name> <New Logo Image File Name> /FN



Chapter 3 ROMSETUP v1.xx

Overview

ROMSETUP can provide OEM customer an easy way to make SETUP manual for AMIBIOS projects. It allows the user to capture SETUP screen from any layer and save it with BMP file format.

Features

The utility offers the following features:

- Simulates BIOS POST to run SETUP. So everything you see is what you get.
- Good compatibility for file system. Any disk drive is usable.
- Simply control interface to save screen and break program.

Requirements

Supported Operating System

ROMSETUP Utility is now supported only in following operating system:

MS-DOS environment without EMM386 installed(non-V86 Mode)

BIOS Requirements User Guide

BIOS MUST have the followings:

- CORE0136 updated Or CORE version 8.00.12 or above.
- BIOS ROM file should be building via "8.00.08_AMITOOLS_17" label or above.

Getting Started

Installation

Copies the *ROMSETUP.EXE* executable file to any storage location accessible by the host system and then run **ROMSETUP** in command prompt.

Running ROMSETUP program

To use ROMSETUP, user can follow the steps as below:

- 1. Boot to DOS and make sure that EMM386 does not install.
- 2. Switch to the directory where ROMSETUP is existing.

 3. Type ROMSETUP behind command prompt symbol(\>) and press ENTER to run

 © Copyright 2005 American Megatrends, Inc. All rights reserved.

 - 18

ROMSETUP, afterward, the usage screen will be displayed:



Print ScreenCapture screen and save it as BMP file format to disk.Ctrl+BreakBreak program and back to DOS.

4. Press any key to continue. SETUP screen will be displayed:

			BIOS SET	UP UTILITY				
Main	Advanced	PCIPnP	Boot	Security	Chi	ipset	Power	Exit
System O	verview					Use or 19	ENTER	[TAB]
AMIBIOS Version Build Da ID	:08.00.12 te:08/31/04 :1AAAA000					Use confi	t a fiel (+) or [-) igure sys	d. 1 to tem Time.
Processo Type Speed Count	r :Intel(R) :2814MHz :1	Pentium (R) 4 CPU	2.80GHz				
System M Size	emory :224MB					↔ 1↓ +-	Select Select	Screen Item Field
System T System D	ime ate		[15:39 [Tue (9:20] 08/31/2004]		Tab F1 F10 ESC	Select General Save an Exit	Field Help d Exit
	v02.56 (C) Copyr ight	1985-20)04, American	n Meg	jatrend	ls, Inc.	

ties

Note: The SETUP screen may be different from above map.

It is depending on OEM's requirement.

5. Choose SETUP screen what you would like to capture and press **Print Screen**. to pop-up dialog box:



Chapter 4 AMIDEDOS v1.xx

Overview

AMIDEDOS is a Desktop Management Interface utility with command line interface. It allows you to modify strings associated with SMBIOS tables on *AMIBIOS* host system.

Features

The utility offers you to modify following SMBIOS table:

- System (Type 1)
- Base Board (Type 2)
- Chassis (Type 3)
- OEM String (Type 11)

Requirements

Supported Operating System

AMIDEDOS Utility is supported only in following operating system:

MS-DOS environment

BIOS Requirements User Guide

System BIOS should have the followings:

- AMIBIOS CORE version 8.xx.xx.
- *SMIFlash eModule* with "8.00.00_SMIFlash-1.00.07" label or later.
- *SMBIOS eModule* with "8.00.08_SMB-3.1.02_CORE_RC2" label or later.

Getting Started

Installation

Copies the *AMIDEDOS.EXE* executable file to any storage location accessible by the host system and then run **AMIDEDOS** in command prompt.

Usage



Configuration File Name

The input file included at least one SMBIOS Table entry. Each SMBIOS table entry contains the SMBIOS table type name followed by the strings to be edited. User can use a text editor Or use "/DMS" command to create an example file. Default file is named "CONFIG.DMS". Following lists the example of SMBIOS configuration file:

American

Megatrends

ROM Utilities

[System]

Manufacturer = AMI Product = COREVersion = 8.00SerialNum = 0123456789UUID = 0123456789ABCDEF0123456789ABCDEF SKUNum = 0123456789 Family = AMI

[BaseBoard]

Manufacturer = AMIProduct = COREVersion = 8.00SerialNum = 0123456789 AIV

[Chassis]

Manufacturer = AML Ser Guide Version = 8.00SerialNum = 0123456789TagNum = 0123456789ChassisType = 03ChassisOEM = 0123456789

[OEMString]

String = AMIString = WWW.AMI.COM String = BIOS vendor

Commands

User can order following commands to select the operation mode for read/write strings associated with SMBIOS tables, create configuration file...etc. The valid commands and related format as below:

Part 1. System (Type 1)

- /SM ["String"]
- /SP ["String"]
- /SV ["String"]
- /SS ["String"]
- /SU [16 Bytes]
- /SKU ["String"]
- /SF ["String"]

Part 2. Base Board (Type 2)

- /BM ["String"]
- /BP ["String"]
- /BV ["String"]
- /BS ["String"]

Part 3. Chassis (Type 3)

- /CM ["String"]
- /CT [8-Bits value]
- /CV ["String"]
- /CS ["String"]
- /CA ["String"]
- /CO [32-Bits value]

Read/Write system manufacturer.

Read/Write system product.

Read/Write system version.

Read/Write system serial number.

Read/Write system UUID.

Read/Write SKU number.

Read/Write family name.

Read/Write baseboard manufacturer. Read/Write baseboard product. Read/Write baseboard version. Read/Write baseboard serial number.

Read/Write chassis manufacturer. Read/Write chassis type. Read/Write chassis version. Read/Write chassis serial number. Read/Write chassis tag. Read/Write chassis OEM-defined value.

Part 4. OEM String (Type 11)

/OS [<Number> <"String">]

Read/Write #th OEM string.

Part 5. Others

- /ALL [Output File Name]
- /DMS [Output File Name]

Output information to screen Or file. Create configuration file. Default file name is "CONFIG.DMS".

Parameters List				
Name	Description			
String	NULL-Terminated ASCII string.			
8-Bits value	This parameter MUST be 2-digits hexadecimal value.			
32-Bits value	This parameter MUST be 8-digits hexadecimal value.			
16 Bytes	This parameter MUST be 32-digits hexadecimal value.			
Number	The decimal value ranges between 1 and 127.			
Output File Name	This parameter is used to specify path/filename of the output file with extension.			

Any parameter encolsed by < > is a mandatory field.

Any parameter enclosed by [] is an optional field.

• For command part 1-4, if parameter present, the WRITE function is going to update else READ function will be enabled.

Rules

 For command </ALL>, if Output File Name present, the SMBIOS information will be saved into the file else it will be displayed on screen.

 Using </DMS> without parameter can get "CONFIG.DMS" file in same directory, otherwise, the user-defined output file will contain the example syntax.

- Using </OS> without any parameter will display all OEM string on screen.
- READ function can be ignored if user run the utility followed by configuration file name.

Note: Running AMIDEDOS under command prompt directly will display help message.

trends

tilities

Examples

Examples on how to access SMBIOS data using the command prompt are shown in following:

- Create "CONFIG.DMS" file
 AMIDEDOS /DMS
- Create new configuration file
 AMIDEDOS /DMS OEM.DMS
- Display SMBIOS strings on screen
 AMIDEDOS /ALL
- Output SMBIOS strings to file
 AMIDEDOS /ALL SMBIOS.TXT
- Update SMBIOS strings by configuration file
 AMIDEDOS CONFIG.DMS
- Update system version, baseboard version and chassis version at once
 AMIDEDOS /SV 1.00 /BV 2.00 /CV 3.00
- Update system manufacturer string
 AMIDEDOS /SM AMI
- Update 1st OEM string
 AMIDEDOS /OS 1 AMIBIOS8
- Get chassis serial number
 AMIDEDOS /CS
- Get system UUID, baseboard version and chassis type information at once
 AMIDEDOS /SU /BV /CT
- Get OEM String
 AMIDEDOS /OS
- Update system manufacturer and get system UUID at once

AMIDEDOS /SM AMI /SU



Chapter 5 AMIDEWIN v1.xx

Overview

AMIDEWIN is a Desktop Management Interface utility with command line interface. It allows you to modify strings associated with SMBIOS tables on *AMIBIOS* host system.

Features

The utility offers you to modify following SMBIOS table:

- System (Type 1)
- Base Board (Type 2)
- Chassis (Type 3)
- OEM String (Type 11)

Requirements

Supported Operating System

AMIDEWIN Utility is supported in following operating system:

- Microsoft® Windows® 98
- Microsoft® Windows® ME
- Microsoft® Windows® 2000
- Microsoft® Windows® NT 4.0
- Microsoft® Windows® XP/XP64
- Microsoft® Windows® PE

BIOS Requirements

System BIOS should have the followings:

- AMIBIOS CORE version 8.xx.xx.
- *SMIFlash eModule* with "8.00.00_SMIFlash-1.00.07" label or later.
- SMBIOS eModule with "8.00.08_SMB-3.1.02_CORE_RC6" label or later.

Operating System Driver Requirements

Following drivers for different operation system are required by this utility:

- UCOREVXD.VXD Driver for Microsoft® Windows® 98/ME.
 - Driver for Microsoft® Windows® NT/2000/XP/PE.
 - UCOREW64.SYS Driver for Microsoft® Windows® XP64.

UCORESYS.SYS

Getting Started

Installation

Copies *AMIDEWIN.EXE*, *UCOREVXD.VXD*, *UCORESYS.SYS* and *UCOREW64.SYS* to any storage location accessible by the host system and then run **AMIDEWIN** in command prompt. Remember that three files MUST be in same directory.

Usage & Example

This utility is same as AMIDEDOS.EXE but running under Microsoft® Windows®. So you can see <u>Usage of AMIDEDOS</u> and <u>Example of AMIDEDOS</u> to learn more information.



Chapter 6 AFUDOS v4.xx

Overview

AFUDOS is an updating system BIOS utility with command line interface. It has no tedious and annoying parameters, just update your system BIOS. Hey!! Do not forget that target board MUST be *AMIBIOS* system.

Features

This utility offers the following features:

- Small executable file size
- Quickly update
- Clear updating information and status
- Fully compatible with previous version (See <u>Appendix B AFUDOS v3.xx Commands</u>)

rican

Requirements

Supported Operating System

This utility is supported by the following operating systems:

MS-DOS environment

User Guide

BIOS Requirements

System BIOS should have the followings:

- AMIBIOS CORE version 8.xx.xx.
- SMIFlash eModule with "8.00.00_SMIFlash-1.00.07" label or later.
- Token: SDSMGR_IN_RUNTIME = ON.
- Token: SMI_INTERFACE_FOR_SDSMGR_FUNC = ON.

Getting Started

Installation

Copies the *AFUDOS.EXE* executable file to any storage location accessible by the host system and then run **AFUDOS** in command prompt.

Usage

 For previous usage, see <u>Appendix B AFUDOS v3.xx Commands</u> to know details.

 © Copyright 2005 American Megatrends, Inc. All rights reserved.
 - 2

AFUDOS <BIOS ROM File Name> [Option 1] [Option 2]......

Or

AFUDOS <Output BIOS ROM File Name> <Commands>

Or

AFUDOS /M<MAC Address>

BIOS ROM File Name

/U

The mandatory field is used to specify path/filename of the BIOS ROM file with extension.

Commands

.

The mandatory field is used to select an operation mode.

- /O Save current ROM image to file
 - Get and display ROM ID from BIOS ROM file

Update BootBlock MAC address if exists

/M<MAC Address>

Options

The optional field used to supply more information for flashing BIOS ROM. Following lists the supported optional parameters and format:

/P Program main bios image **/B** Program Boot Block /N Program NVRAM /CDestroy CMOS after update BIOS done ser Gui /E Program Embedded Controller block if present **/K** Program all non-critical blocks /Kn Program n'th non-critical block only (n=0 - 7) **/O** Quiet mode enable . **/REBOOT** Reboot after update BIOS done /**X** Do not check ROM ID **/S** Display current system's BIOS ROM ID /Ln Load CMOS default (n=0 - 1) L0: Load current CMOS optimal settings L1: Load current CMOS failsafe settings L2: Load CMOS optimal settings from ROM file L3: Load CMOS failsafe settings from ROM file /M<MAC Address> Update BootBlock MAC address if exists **/R** Preserve all SMBIOS structures during NVRAM programming /Rn Preserve specific SMBIOS structure during NVRAM programming

	Rules
•	Any parameter encolsed by < > is a mandatory field.
-	Any parameter enclosed by [] is an optional field.
-	<commands> cannot co-exist with any [Options].</commands>
-	Main BIOS image is default flashing area if no any option present.
-	[/C], [/Q], [/REBOOT], [/X], [/Ln] and [/S] will enable [/P] function automatically.
-	If [/B] present alone, there is only the Boot Block area to be updated.
-	If [/N] present alone, there is only the NVRAM area to be updated.
•	If [/E] present alone, there is only the Embedded Controller block to be updated.
•	If [/Kn] present alone, there is only non-critical block to be updated.
-	When [/Ln] is co-exist with [/C], [/C] will be no function.
	[/M] can be used as a command for backward compatible.

Note: Running AFUDOS under command prompt directly will display help message.

Examples

Examples on how to update BIOS using the command prompt are shown in following:

- Save current BIOS ROM to file
 AFUDOS < BIOS ROM File Name > /0
- Get and display ROM ID from BIOS ROM file
 AFUDOS <BIOS ROM File Name> /U
- Update main BIOS image only
 AFUDOS <BIOS ROM File Name>

Or

AFUDOS <BIOS ROM File Name>/p

- Update Boot Block only
 AFUDOS <BIOS ROM File Name> /B
- Update NVRAM only
 AFUDOS <BIOS ROM File Name> /N
- Update Embedded Controller Block only
 AFUDOS <BIOS ROM File Name> /E
- Update 2nd non-critical block only *AFUDOS <BIOS ROM File Name> /K2*
- Update main BIOS image, Boot Block and NVRAM at once *AFUDOS <BIOS ROM File Name> /P /B /N*
- Update whole BIOS ROM
 AFUDOS <BIOS ROM File Name> /P /B /N /C /E /K
- Update whole BIOS ROM and load current CMOS optimal settings *AFUDOS <BIOS ROM File Name> /P /B /N /C /E /K /L0*

- Update whole BIOS without checking ROM ID *AFUDOS <BIOS ROM File Name> /P /B /N /C /E /K /X*
- Update whole BIOS with quiet execution *AFUDOS <BIOS ROM File Name> /P /B /N /C /E /K /Q*
- Update whole BIOS in quiet mode and REBOOT quietly *AFUDOS <BIOS ROM File Name> /P /B /N /C /E /K /Q /REBOOT*
- Update BootBlock MAC address
 AFUDOS /M<MAC Address>
- Update whole BIOS and BootBlock MAC address
 AFUDOS <BIOS ROM File Name> /P /B /N /C /E /K /M<MAC Address>
- Update whole BIOS except existing SMBIOS structures
 AFUDOS <BIOS ROM File Name> /P /B /N /C /E /K /R
- Update whole BIOS but preserve SMBIOS type 0 and 11 *AFUDOS <BIOS ROM File Name> /P /B /N /C /E /K /R0 /R11*



Chapter 7 AFUWIN v4.xx

Overview

AFUWIN is an updating system BIOS utility with command line and GUI interface. It has same parameters and behavior as AFUDOS, and further, GUI feature starting from v4.10 can provide you a friendly environment to visualize BIOS update procedure. By the way, do not forget that target board MUST be *AMIBIOS* system while using this utility.

Features

This utility offers the following features:

- Small executable file size
- Quickly update
- Clear updating information and status
- Fully compatible with previous version (See <u>Appendix B AFUDOS v3.xx Commands</u>)

Requirements

Supported Operating System

AFUWIN Utility is supported in following operating system:

- Microsoft® Windows® 98
- Microsoft® Windows® ME
- Microsoft® Windows® 2000
- Microsoft® Windows® NT 4.0
- Microsoft® Windows® XP/XP64
- Microsoft® Windows® PE

BIOS Requirements

System BIOS should have the followings:

- AMIBIOS CORE version 8.xx.xx.
- *SMIFlash eModule* with "8.00.00_SMIFlash-1.00.07" label or later.
- Token: SDSMGR_IN_RUNTIME = ON.
- Token: SMI_INTERFACE_FOR_SDSMGR_FUNC = ON.

Operating System Driver Requirements

Following drivers for different operation system are required by this utility:

- UCOREVXD.VXD
- UCORESYS.SYS
- UCOREW64.SYS

Driver for Microsoft® Windows® 98/ME. Driver for Microsoft® Windows® NT/2000/XP/PE. Driver for Microsoft® Windows® XP64.

Getting Started

Installation

Copies AFUWIN.EXE, UCOREVXD.VXD, UCORESYS.SYS and UCOREW64.SYS to any storage location accessible by the host system and then run AFUWIN in command prompt. Remember that three files MUST be in same directory. For launching GUI mode, you can just double-click on the icon.

Usage & Example for command line mode

This part is same as AFUDOS.EXE but running under Microsoft® Windows®. So you can see Usage of AFUDOS and Example of AFUDOS to learn more information.

Main Window

n Window	Megatren	ls
AFUWIN v4.10	×	~~
	Information Setup Progress	es
American Megatrends	System Information OS: Windows XP Chip: SST 39SF/VF040 BIOS Size: 524288 bytes Bootblock Size: 65536 bytes NVRAM Size: 65536 bytes Ocre Version: 08.00.12 Release Date: 03/11/05 System ROM ID: 1AAAA000 ROM File Information Input ROM File: C:\BIOS.ROM File ROM ID: File ROM ID: 1AAAA000	

Buttons

AFUWIN v4.10	
American Megatrends	Information Setup Progress System Information OS: System Information OS: Windows XP System Information Chip: SST 39SF/VF040 Size: BIOS Size: 524288 bytes South State Bootblock Size: 65536 bytes South State NVRAM Size: 65536 bytes South State Core Version: 08.00.12 Release Date: 03/11/05 System ROM ID: 1AAAA000 ROM File Information Input ROM File: CABIOS.ROM File ROM ID: 1AAAA000 Exit Save Exit
Open Flash Save	Click this button to search for BIOS ROM file from any disk drive. Click this button to starting update BIOS. Click this button to save BIOS ROM image to disk drive.
nction Frame	Click this button to exit this program.

Function Frame

Information Tab

This tab displays system BIOS information for your reference before flashing BIOS.

Information Setur	Progress
	System Information
OS :	Windows XP
Chip :	SST 39SF/VF040
BIOS Size :	524288 bytes
Bootblock Size :	65536 bytes
NVRAM Size :	65536 bytes
Core Version :	08.00.12
Release Date :	03/11/05
System ROM ID :	1AAAA000
	-ROM File Information
Input ROM File :	C:\BIOS.ROM
File ROM ID :	1AAAA000

Field

Name	Description
OS	This field displays current O/S version.

Name	Description
Chip	This field displays current flash part on the system.
BIOS Size	This field displays current BIOS ROM size.
BootBlock Size	This field displays current BIOS BootBlock size.
NVRAM Size	This field displays current BIOS NVRAM size.
Core Version	This field displays current AMIBIOS CORE version.
Release Date	This field displays current BIOS release date.
System ROM ID	This field displays current system BIOS ROM ID.
Input ROM File	This field displays BIOS ROM image file name/path where will be used to
	instead of old one.
File ROM ID	This field displays ROM ID in given BIOS ROM image file.

Setup Tab

This tab allows you to change the settings for flashing BIOS.



Field

Block Options		
Name	Description	
Program All Block	This option is used to enable all programmable blocks.	
Main BIOS Image	This option is used to determine if Main BIOS Image needs to update.	
Boot Block	This option is used to determine if Boot Blcok needs to update.	
NVRAM	This option is used to determine if NVRAM needs to update.	
EC Block	This option is used to determine if EC Block needs to update.	

CMOS Options						
Name	Description					
Nothing	Enable if you want to do nothing for CMOS after BIOS updated.					
Load Current Optimal	Enable if you do like to load CMOS optimal settings from current					
	system after BIOS updated.					
CMOS Options						
--------------------------	---	--	--	--	--	--
Name	Description					
Load Current Failsafe	Enable if you do like to load CMOS failsafe settings from current					
	system after BIOS updated.					
Load ROM File's Optimal	Enable if you do like to load CMOS optimal settings from current					
	system after BIOS updated.					
Load ROM File's Failsafe	Enable if you do like to load CMOS failsafe settings from current					
	system after BIOS updated.					
Destroy CMOS Checksum	Enable if you do like to destroy CMOS checksum after BIOS					
	updated. This is default setting in CMOS Options block.					

Non Critical Block				
Name	Description			
All	Enable if you want to update all Non Critical Blocks.			
1 – 8	Enable one of Non Critical Blocks if it needs to update.			

Miscellaneous							
Name	Description						
Do Not Check ROM ID	Enable if you do not want to check ROM ID before updating BIOS.						
Restart after Programming	Enable if you want to restart system after BIOS updated.						
Preserve SMBIOS Type	This field allows you to preserve SMBIOS types while BIOS						
	updating. The types string must be decimal-digit and separated by						
	a space(' ') character. For convenence, you can strike 'A' key as						
	first character to select all SMBIOS structures at once.						
Update MAC	This field is used to change BootBlock MAC address. It MUST be						
	hexadecimal-digit string.						

Progress Tab

This tab displays the updating status.



Field

Name	Description
ROM Map	This area displays current updating status.
Legend	This area illustrates the meaning of color in ROM MAP area.
Stage	This field displays the stage of updating BIOS.
Address	This field display the address where block is under working.

Functions

To launch into AFUWIN with GUI mode, you can double-click the executable file icon to open the operating window:

AFUWIN v4.10		×
	Information Setup Progress	
American Megatrends	System Information OS : Windows XP Chip : SST 39SF/VF040 BIOS Size : 524288 bytes Bootblock Size : 65536 bytes NVRAM Size : 65536 bytes Core Version : 08.00.12 Release Date : 03/11/05 System ROM ID : 1AAAA000 ROM File Information Input ROM File : Unknown File ROM ID : Unknown	
	Open <u>F</u> lash <u>S</u> ave <u>Exit</u>	

Usually, system BIOS information will be displayed first, but you may see a pop-up dialog if the system does not support AMIBIOS update function. After open this program successfully, you can refer to following steps to finish the operation what you need:

Saving system BIOS ROM image to file

- 1. Press <u>Save</u> button to open file dialog box.
- 2. Select path and input a file name.
- 3. Click on OK button to save system BIOS ROM image into specific file.
- 4. Press <u>Exit</u> button to exit this program.

Flashing system BIOS with given file

1. Press ______ button to search for BIOS ROM image file from any disk driver and

load it into memory.

- 2. Switch to *Setup Tab* to check and change necessary settings.
- 3. Press Flash button to start the operation.
- 4. *Progress Tab* will be switched automatically and display the programming status.
- 5. After BIOS updated, you can press $\underline{E_{xit}}$ button to exit this program or system

will restart automatically if the **Restart After Programming** option enabled.



Chapter 8 AMISCE v1.xx/v2.xx

Overview

AMISCE is an abStract CMOS Editor utility with command line interface. It can produce a script file that lists all the existing BIOS Setup Questions in the system where the utility is running. The script file will list all setup questions whether they actually show in BIOS Setup screens or not. This script file generated can also be modified and used as input to change the BIOS setup current values.

Features

The utility offers you following features:

• BIOS SETUP values can be edited under operation system by TEXT script file

erican

• Display, save and restore current CMOS contents

Requirements

Supported Operating System

AMISCE Utility is supported only in following operating system:

• MS-DOS environment

User Guide BIOS Requirements

System BIOS should have the followings:

- AMIBIOS CORE version 8.xx.xx.
- SMIFlash eModule with "8.00.00_SMIFlash-1.00.07" label or later.
- Token: SDSMGR_IN_RUNTIME = ON.
- Token: SMI_INTERFACE_FOR_SDSMGR_FUNC = ON.

Getting Started

Installation

Copies the *AMISCE.EXE* executable file to any storage location accessible by the host system and then run **AMISCE** in command prompt.

Usage

AMISCE <Command>

© Copyright 2005 American Megatrends, Inc. All rights reserved.

Commands

User can order following commands to select the operation mode for handling TEXT script file. The valid commands and related format as below:

- /O <Script File Name>
- /OX <Script File Name>
- /OC <Script File Name>
- /I <Script File Name>
- /CR [CMOS Image File Name]
- /CW <CMOS Image File Name>

Create TEXT script file with overwrite. Same as /**O** command but more information as CMOS Index register, Mask bits...etc. Parse TEXT script file and update CMOS. Display/Save CMOS contents. Restore CMOS contents.

Create TEXT script file without overwrite.

Parameters List						
Name	Description					
Script File Name	This parameter is used to specify path/filename of the TEXT script file with					
	extension.					
CMOS Image File	This parameter is used to specify path/filename of the CMOS Image file with					
Name	extension.					
	Iviegatrenus					

	Rules
•	Any parameter encolsed by < > is a mandatory field.
•	Any parameter enclosed by [] is an optional field.

Cruiue

Note: Running AMISCE under command prompt directly will display help message.

CI

Script Syntax

/O, **/OX** and **/OC** commands can generate a script file, which lists all the BIOS Setup questions for the system where the utility is running. The file consists of the following type of statements:

Comments

Comments are end-of-line comments and they start with the double slash "//". Any text will be ignored from the beginning of the "//" to the end of the line when parsing the script file.

Comments can be added anywhere in the file without affecting the behavior of the utility.

BIOS Setup Question

A BIOS Setup Question has five parts:

Setup Question Text

This is the first statement in the Setup Question and it displays the text that appears in the BIOS Setup Screen for that particular Setup Question.

Token

This field **MUST NOT** be modified.

BIOS Default

This is the BIOS Default setting for the current Setup Question. This field is for information only and modifying it has no effect.

MFG Default

This is the Manufacturing Default setting for the current Setup Question. This field is for information only and modifying it has no effect.

Options or Value

A Setup Question may have either one of these statements. These are the only modifiable fields in the Setup Questions.

Options

🔹 Regular

A list of all possible settings for the Setup Question appears following the "Options" statement. An "*" (asterisk) indicates the currently selected option. Change the setting by simply moving the asterisk to the desired option.

Do not change any of the text in the option list, specially the value inside the square brackets.

There must be only one asterisk in a particular Option Set.

• Child with One Option Set

After the "Options" statement, there will be a string enclosed in "< ...>" which tells what the "parent" question is and lists the options for the "parent" question.

Change the current option by just moving the asterisk to the desired option.

Do not change any of the text in the option list, specially the value inside the square brackets.

There must be only one asterisk in a particular Option Set.

Child with Multiple Option Set

Each Option Set will have a line enclosed in "<...>" which describes for which value or values of the Parent Question the following Option Set is valid.

Change the current option by just moving the asterisk to the desired option. Check the current setting of the "parent" question to see which of the Option Sets is valid and then move the asterisk to the desired option.

There must be only one asterisk in a particular Option Set and the value of the current setting must be the same in all Option Sets for a particular Setup Question.

■ Value

This "value" corresponds to the actual CMOS value of the CMOS bits reserved for the current Setup Question. There is no string to display the meaning of this setting. Changing this setting requires knowledge about the implementation details for the Setup Question.

BIOS Setup Question Examples

- Options
- Regular Setup Question Diskette A = Token 0000 // Do NOT change this line _ BIOS Default [04]1.44/1.25 MB 3¹/₂ = MFG Default [04]1.44/1.25 MB 3¹/2 = // Move "*" to the desired Option Options [00]Not Installed [01]360 KB 5¼ [02]1.2 MB 5¼ [03]720 KB 3½ *[04]1.44/1.25 MB 3½ e [05]2.88 MB 3½

Child with One Option Set Setup Question = USB KB/Mouse Legacy

Token	=	007C	//	Do	NOT	change	this	line
BIOS Default	=	[02]Aı	uto					
MFG Default	=	[01]Ke	eyboaı	rd				
// Move "*" to the .	desiı	ced Op	tion					
Options	=	<usb 1<="" td=""><td>Funct</td><td>ion</td><td>= D:</td><td>isabled,</td><td>Enab</td><td>oled></td></usb>	Funct	ion	= D:	isabled,	Enab	oled>
		[00]D:	isable	ed				
		[01]Ke	eyboaı	rd				

- *****[02]Auto
- [03]Keyb+Mouse

Child with Multiple Option Set

Setup Question	=	PCI0 Age	ent	То	Aper	ture A	ccess	
Token	=	0085	//	Do	NOT	change	this	line
BIOS Default	=	[00]N/A						
MFG Default	=	[00]N/A					,	12
© Copyright 2005 American Meganenus, Inc. An rights reserved	u.						- 4	

// Move "*" t	o t	he	desir	ed Option
Options			=	<pre><aperture access="" enable="Disabled"></aperture></pre>
			=	*[00]N/A
				<pre><aperture access="" enable="Enabled"></aperture></pre>
				*[00]Enabled
				[01]Disabled
Value				
Setup Question		=	L1/L	2 Cache
Token		=	006E	// Do NOT change this line
BIOS Default		=	[02]	WriteBack
MFG Default		=	[02]	WriteBack
Value		=	02	// Change to the desired value

Examples

Examples on how to process BIOS SETUP values using the command prompt are shown in following:

- Create TEXT script file but do not overwrite if the file existed
 AMISCE /O <Script File Name>
- Create TEXT script file and overwrite if the file existed
 AMISCE /OX <Script File Name>
- Create new TEXT script file to get CMOS index reg. and mask bits information
 AMISCE /OC <Script File Name>
- Display CMOS contents
 AMISCE /CR
- Save CMOS contents to file
 AMISCE /CR <CMOS Image File Name>
- Restore CMOS contents
 AMISCE /CW <CMOS Image File Name>
- Update CMOS contents by TEXT script file
 AMISCE /I <Script File Name >

Chapter 9 AMISCEW v1.xx/v2.xx

Overview

AMISCEW is an abStract CMOS Editor utility with command line interface. It can produce a script file that lists all the existing BIOS Setup Questions in the system where the utility is running. The script file will list all setup questions whether they actually show in BIOS Setup screens or not. This script file generated can also be modified and used as input to change the BIOS setup current values.

Features

This utility offers the following features:

• BIOS SETUP values can be edited under operation system by TEXT script file

Requirements

Supported Operating System

AMISCEW Utility is supported in following operating system:

- Microsoft® Windows® 98
- Microsoft® Windows® ME
- Microsoft® Windows® 2000
- Microsoft® Windows® NT 4.0
- Microsoft® Windows® XP/XP64
- Microsoft® Windows® PE

BIOS Requirements

System BIOS should have the followings:

- AMIBIOS CORE version 8.xx.xx.
- *SMIFlash eModule* with "8.00.00_SMIFlash-1.00.07" label or later.
- Token: SDSMGR_IN_RUNTIME = ON.
- Token: SMI_INTERFACE_FOR_SDSMGR_FUNC = ON.

Operating System Driver Requirements

Following drivers for different operation system are required by this utility:

- UCOREVXD.VXD Driver for Microsoft® Windows® 98/ME.
- UCORESYS.SYS Driver for Microsoft® Windows® NT/2000/XP/PE.

Getting Started

Installation

Copies *AMISCEW.EXE*, *UCOREVXD.VXD*, *UCORESYS.SYS* and *UCOREW64.SYS* to any storage location accessible by the host system and then run **AMISCEW** in command prompt. Remember that three files MUST be in same directory.

Usage & Example

This utility is same as AMISCE.EXE but running under Microsoft® Windows®. So you can see <u>Usage of AMISCE</u> and <u>Example of AMISCE</u> to learn more information.



Chapter 10 AMICMOS v2.xx

Overview

AMICMOS is a CMOS RAM contents processor with command line interface. It is useful in factory to produce CMOS RAM image on same case.

egatrends

Features

This utility offers the following features:

• Display CMOS RAM contents as table

Save/Restore current CMOS RAM contents

Requirements

Supported Operating System

This utility is supported by the following operating systems:

• MS-DOS environment.

BIOS Requirements IOS ROM Utilities

System BIOS should have the followings:

- AMIBIOS CORE version 8.xx.xx.
- SMIFlash eModule with "8.00.00_SMIFlash-1.00.07" label or later.
- Token: SDSMGR_IN_RUNTIME = ON.
- Token: SMI_INTERFACE_FOR_SDSMGR_FUNC = ON.

Getting Started

Installation

Copies the *AMICMOS.EXE* executable file to any storage location accessible by the host system and then run **AMICMOS** in command prompt.

Usage

AMICMOS <Command>

Commands

The mandatory field used to select an operation mode for processing CMOS RAM © Copyright 2005 American Megatrends, Inc. All rights reserved. - 46 - contents. Following lists the supported commands and format:

- /CR [CMOS Image File Name]
- Display/Save CMOS contents.
- /CW <CMOS Image File Name>

Restore	CMOS	contents.

Parameters List					
Name	Description				
CMOS Image File	This parameter is used to specify path/filename of the CMOS contents file with				
Name	extension.				

	Rules
-	Any parameter encolsed by < > is a mandatory field.
-	Any parameter enclosed by [] is an optional field.

Note: Running AMICMOS under command prompt directly will display help message.

Examples

Examples on how to display, save and restore CMOS RAM contents using the command prompt are shown in following:

- Display CMOS contents on screen
 AMICMOS /CR
- Save CMOS contents as CMOS Image file
 AMICMOS /CR <CMOS Image File Name>
- Restore CMOS contents
 AMICMOS /CW < CMOS Image File Name>

Part 2. Graphical User Interface Mode

Chapter 1 OEMLOGO v3.xx

Overview

OEMLOGO is a changing logo tool with graphical user interface. It allows you to replace the OEM Logo(Large) and OEM Logo(Small) module inside the BIOS ROM file with a new one.

Features

This utility offers following features:

- Change OEM/small logo.
- Remove OEM/small logo.
- Check logo image format automatically to make sure the logo works with target BIOS.

Requirements

Supported Operating System

OEMLOGO Utility is supported in the following operating systems:

- Microsoft® Windows® 98
- Microsoft® Windows® ME
- Microsoft® Windows® NT 4.0
- Microsoft® Windows® 2000
- Microsoft® Windows® XP/XP64
- Microsoft® Windows® PE

BIOS Requirements

The loaded BIOS ROM file should have the followings:

- The file MUST be an AMIBIOS ROM file (Core version 8.xx.xx only)
- BIOS ROM file should be building via "8.00.08_AMITOOLS_17" label or above.
- Large OEM Logo module (Module ID 0x0E) to be present
- Small OEM Logo module (Module ID 0x1A) to be present
- Quiet Boot function should be inside. It is recommended to use *DisplayLogo2 eModule* with "8.00.08_DISPLAYLOGO_05" label or later.

New Logo File Requirements

The Change OEM Logo Utility requires that the new Logo file fit the following format:

- 16-Color Bitmap format, even width, 640*480 pixels (Maximum)
- 256-Color Bitmap format, even width, 640*480 pixels (Maximum)
- 256-Color PCX format, even width, 640*480 pixels (Maximum)
- True-Color JPG format, even width, 640*480/800*600/1024*768 pixels (Maximum)

Note: Small OEM Logo does support only 640*80, 16-Color Bitmap format.

Getting Started

Installation

Copies the *OEMLOGO.EXE* executable file to any storage location accessible by the host system and then double-click **OEMLOGO** icon Or type **OEMLOGO** in command prompt to run.

Buttons merica _ 🗆 × Load ROM Select the logo module ● OEM Logo ID = 0x0E you want to replace. Save Image Select 640*480, 16-color BMP file; 640x480, 256-color BMP file; Browse 640x480, 800x600, 1024x768 JPEG file; or 640x480, 256-color PCX file Replace Image New logo is created. Save ROM As Close Load ROM Click this button to search for BIOS ROM file from any disk drive.

Save Image Click this button to extract Logo Image from ROM to any disk drive

by given PATH/FileName.

Erowse Click this button to search for new Logo Image file from any disk drive.

 Replace Image
 Click this button to replace an existing BIOS Logo module inside the

 BIOS ROM file.
 ROM file.

Save ROM As Click this button to save the changes that you have made to the BIOS

ROM file. You can also specify the location and to change the existing file name.

Click this button to exit the program.

Options

Close

ở OEM LOGO v3.06			
Load AMIBIOS 8.xx ROM	file to modify		Load ROM
D:\Project\1AAAA000.rom			
Select the logo module you want to replace.	⊙ OEM Logo ID = 0x0E □ 1 C Small Logo ID = 0x1A	Do not convert GRFX	Save Image
Select 640*480, 16-color Bl	MP file; 640x480, 256-color BMP file;		Browse
640x480, 800x600, 1024x7	68 JPEG file; or 640x480, 256-color P	CX file	
D:\Project\OEMLOGO.jpg			
Replace Image New logo	is created.	<u>S</u> ave ROM As	Close
OEM Logo ID=0x0E	Choose this option can cha	inge OEM logo me	odule.
Do not convert GRF	This option is visible only	for when OEM Lo	ogo is enabled.
	If this option to be enabled	, the 16-Color BM	IP image will NO
AMIR	convert to GRFX format.	Titiliti	es
Small Logo ID=0x1A	Choose this option can cha	nge small logo mo	odule.
	User Guid	le	

Functions

To use OEMLOGO, you can double-click the executable file icon to open the operating window:

2 OEM LOGO v3.00		
Load AMIBIOS 8.xx ROM file to modify		Load ROM
J	Γ	Save Image
	Ī	Browse
Replace Image	Save ROM As	<u>C</u> lose

And refer to the following steps to change new logo:

Changing OEM Logo

1. Click Load ROM button to load the BIOS ROM file which contained OEM Logo

Module(0Eh) from disk drive.

al OEM LOGO v3.06		<u>-0×</u>
Load AMIBIOS 8.xx ROM	file to modify	Load ROM
Select the logo module	 OEM Logo ID = 0x0E Do not convert GRFX 	
Select 640*480, 16-color E	Small Logo ID = 0x1A MP file: 640x480, 256-color BMP file:	Save Image Browse
640x480, 800x600, 1024x	768 JPEG file; or 640x480, 256-color PCX file	
[
<u>Replace Image</u>	Save ROM A	

2. Click <u>Browse</u> button to search for new Logo Image file from disk drive.

沈 OEM LOGO v3.06		
Load AMIBIOS 8.xx ROM	file to modify	Load ROM
D:\Project\1AAAA000.ron	1	
Select the logo module you want to replace.	⊙ OEM Logo ID = 0x0E	GRFX Save Image
Select 640*480, 16-color E	MP file; 640x480, 256-color BMP file;	Browse
640x480, 800x600, 1024x	768 JPEG file; or 640x480, 256-color PCX file	
D:\Project\OEMLOGO.jpg		
<u>R</u> eplace Image	Save R	OM As

3. Click Replace Image button to change logo module. If the function works fine, *New*

logo is created message will be displayed behind the button.

AL OEM LOGO v3.06			
Load AMIBIOS 8.xx ROM	file to modify		Load ROM
D:\Project\1AAAA000.rom	1		
Select the logo module you want to replace.	⊙ OEM Logo ID = 0x0E □ D ⊙ Small Logo ID = 0x1A	o not convert GRFX	Save <u>I</u> mage
Select 640*480, 16-color B	MP file; 640x480, 256-color BMP file;		Browse
640x480, 800x600, 1024x7	768 JPEG file; or 640x480, 256-color PC	X file	
D:\Project\OEMLOGO.jpg			
Replace Image New logo	o is created	Save ROM As	Close
. Click Save ROM A:	^s button to save new BIOS RO	DM file to disk dı	rive.
. Click <u>Close</u>	button to exit program.		
. Click <u>Close</u>	☐ button to exit program.	•	
. Click <u>Close</u>	button to exit program.	rica	n
. Click <u>Close</u> nging Small Logo . Click <u>Losd ROM</u>	 button to exit program. button to load the BIOS RO 	M file which con	tained Smal
 Click <u>Close</u> nging Small Logo Click Load ROM Module(1Ah) fro 	 button to exit program. button to load the BIOS RO button disk drive and choose Small 	M file which con	tained Smal
 Click <u>Close</u> Click <u>Losd ROM</u> Click <u>Losd ROM</u> Module(1Ah) fro MOGUNE(1Ah) COM 	button to exit program. button to load the BIOS RO button to load the BIOS RO button disk drive and choose Small	M file which con	tained Smal option.
 Click <u>Close</u> Aging Small Logo Click Load ROM Module(1Ah) fro ACEM LOGO ∨3.06 	button to exit program. button to load the BIOS RO om disk drive and choose Small	M file which con	tained Smal option.
 Click <u>Close</u> Click <u>Load ROM</u> Click <u>Load ROM</u> Module(1Ah) fro 21 OEM LOGO v3.06 Load AMIBIOS 8.xx ROM 	button to exit program. button to load the BIOS RO om disk drive and choose Small	M file which con	tained Smal option.
 Click <u>Close</u> Click Load ROM Click Load ROM Module(1Ah) fro OEM LOGO v3.06 Load AMIBIOS 8.xx ROM D:Project%1AAAA0000.rom 	button to exit program. button to load the BIOS RO om disk drive and choose Small file to modify	M file which con	tained Smal option.
 Click <u>Close</u> Click Load ROM Click Load ROM Module(1Ah) fro OEM LOGO v3.06 Load AMIBIOS 8.xx ROM D:Project\1AAAA000.rom Select the logo module you want to replace. 	button to exit program. button to load the BIOS RO om disk drive and choose Small file to modify COEM Logo ID = 0x0E Small Logo ID = 0x1A	M file which con	tained Smal option. Load ROM
 Click <u>Close</u> Click <u>Load ROM</u> Click <u>Load ROM</u> Click <u>Load ROM</u> Module(1Ah) fro OEM LOGO v3.06 Load AMIBIOS 8.xx ROM D:Project%1AAAA000 rom Select the logo module you want to replace. Select 640*80, 16-color BM 	button to exit program.	M file which con	tained Small option. Load ROM Save Image Browse
Click <u>Close</u> Ging Small Logo Click Load ROM Module(1Ah) fro OEM LOGO v3.06 Load AMIBIOS 8.xx ROM D:Project\1AAAA000.rom Select the logo module you want to replace. Select 640*80, 16-color BM	button to exit program.	M file which con	tained Small option. Load ROM Save Image Browse
Click <u>Close</u> Ging Small Logo Click Load ROM Module(1Ah) fro OEM LOGO ∨3.06 Load AMIBIOS 8.xx ROM D:\Project\1AAAA000.rom Select the logo module you want to replace. Select 640*80, 16-color BN Replace Image	button to exit program.	M file which con Logo ID=0x1A	tained Sma option. Load ROM Save Image Browse

2. Click **Browse** button to search for new Logo Image file from disk drive.

file to modify		Load ROM
1		
○ OEM Logo ID = 0x0E ⓒ Small Logo ID = 0x1A		Save <u>I</u> mage
4P file	[Browse
p		
	Save ROM As	<u>C</u> lose
	file to modify OEM Logo ID = 0x0E Small Logo ID = 0x1A dP file	file to modify OEM Logo ID = 0x0E Small Logo ID = 0x1A dP file P <u>Save ROM As</u>

3. Click Replace Image button to change logo module. If the function works fine, *New*

logo is created message will be displayed in right of the button.

file to modify		Load ROM
○ OEM Logo ID = 0x0E ○ Small Logo ID = 0x1A		Save <u>I</u> mage
P file		Browse
is created	Save ROM As	<u>C</u> lose
f	ile to modify © OEM Logo ID = 0x0E © Small Logo ID = 0x1A P file is created	ile to modify © OEM Logo ID = 0x0E © Small Logo ID = 0x1A P file is created <u>Save ROM As</u>

- 4. Click <u>Save ROM As</u> button to save new BIOS ROM file to disk drive.
- 5. Click <u>Close</u> button to exit program.

Chapter 2 MMTOOL v3.xx

Overview

MMTOOL is a Module Manipulation Tools with graphical user interface. It allows you to manage the BIOS modules that are contained in the BIOS ROM file.

erican

Features

This utility offers following features:

- Insert Module.
- Replace Module.
- Extract Module.
- Delete Module.
- Display ROM Information
- Display/Change ROM Hole content
- Display NCB Information
- Edit CPU Micro Code Patches module

Requirements TRIOS ROV Utilities

Supported Operating System

MMTOOL Utility is supported in the following operating systems:

- Microsoft® Windows® 98
- Microsoft® Windows® ME
- Microsoft® Windows® NT 4.0
- Microsoft® Windows® 2000
- Microsoft® Windows® XP/XP64
- Microsoft® Windows® PE

BIOS Requirements

The loaded BIOS ROM file should have the followings:

- The file MUST be an AMIBIOS ROM file (Core version 8.xx.xx only)
- BIOS ROM file should be building via "8.00.08_AMITOOLS_17" label or above.

Getting Started

Installation

Copies the *MMTOOL.EXE* executable file to any storage location accessible by the host system and then double-click **MMTOOL** icon Or type **MMTOOL** in command prompt to run.

Main Window

Load ROM Save ROM we ROM as <u>C</u> lose	Module Module Offset/V Seg/DI	Replace De file: ID: ID: ID: D.	For Adapte: For Adapte: Link P Link Vend	r ROM only resent or ID:	lole NC	ert Compress M Insert Uncor omRegion	ATCH Browse fodule mpressed
) Name		RomLoc	Source size	Insert Size in Rom	%%	RunLoc	NCB

Main Buttons



existing file name.



Click this button to exit the program.

Function Frame

Insert Module

The function allows you to add a new BIOS module into the BIOS ROM file.

Insert Replace Delete	Extract ROM Info RomHole	NCB CPU PATCH
Module file: Module ID: Offset/VID: Seg./DID:	For Adapter ROM only Link Present Link Vendor ID: Link Device ID:	Browse Insert Compress Module Insert Uncompressed RomRegion
	Insert	

Name	Description
Module File	This field is used to specify path/filename of new module file with extension.
Module ID	2-digits hexadecimal Module ID. See <u>Appendix A Module ID Codes</u> for detail.
Offset/VID	This filed is used to enter a new module runtime Offset. This field is optional
	except when inserting an Adapter ROM or Multilanguage modules. You must
	enter the Vendor ID for the Adapter ROM.
	The default value for the Offset field is equal to zero. The value indicates that
	runtime location is dynamic.
Segment/DID	This filed is used to enter a new module runtime Segment. This field is optional
	except when inserting an Adapter ROM or Multilanguage modules. You must
	enter the Device ID for the Adapter ROM.
	The default value for the Segment field is equal to zero. The value indicates
	that runtime location is dynamic.
Link Vendor ID	This field is used to enter the PCI vendor ID for the PCI device that uses the
	option ROM.
	Note:
	This filed must be filled only if Link Present check box is selected. You
	must enter the vendor ID of the PCI device that shares the same option ROM
	with an existing device.
Link Device ID	This field is used to enter the PCI device ID for the PCI device that uses the
	option ROM.
	Note:
	This filed must be filled only if Link Present check box is selected. You

Name	Description
	must enter the device ID of the PCI device that shares the same option ROM
	with an existing device.
RomRegion	This field is used to insert the module into a Non-Critical region. User must be
	sure that region name is present in current BIOS ROM file, otherwise, the
	module will still insert to Main BIOS Image. You can type region name directly
	Or select an available region from drop-down menu.
	Note:
	Non-Critical Block contains BIOS modules that do not prevent BIOS
	POST from completing its execution. Examples of Non-Critical Blocks are
	option ROM for onboard devices, logos, language modules, setup clients and
	user defined modules.

Note: MMTOOL does not check to see if the module file is valid.

Buttons

lcon	Description	
Browse	This button is used to search for a new module file from any storage location.	
Insert	This button is used to launch the insert module operation.	

Options

Options Megatrends		
Name	Description	
Link Present	This option(check box) is used to support multiple PCI devices with a single	
	PCI adapter ROM. This option is only for PCI adapter ROM, thus, the Module	
	ID is always fixed at 20h. User can input Linked Vendor ID/Device ID to share	
	PCI adapter ROM with an existing one.	
Compress	This option is used to insert the new module in compact form.	
Module	Note:	
	Some modules MUST be uncompressed, for example: BootBlock-	
	Runtime interface, CPU MicroCode Or ROMID.	
Insert	This option is used to insert the module in its original form.	
Uncompressed		

Replace Module

This function allows you to substitute an existing BIOS module into BIOS ROM file with a new one.

Insert	Replace	Delete Extract ROM Info Ron	nHole NCB CPU PATCH
Module	file:	[Browse
Module Offset: Segmer	ID: at:		
		Replace	

Field

Name	Description	
Module File	This field is used to specify path/filename of new module file with extension.	
Module ID	2-digits hexadecimal Module ID. See <u>Appendix A Module ID Codes</u> for detail.	
Offset	This filed is used to enter a new module runtime Offset. This field is optional	
Vendor ID	except when replacing an Adapter ROM or Multilanguage modules. You must	
	enter the Vendor ID for the Adapter ROM.	
	The default value for the Offset field is equal to zero. The value indicates that	
	runtime location is dynamic.	
Segment	This filed is used to enter a new module runtime Segment. This field is optional	
Device ID	except when replacing an Adapter ROM or Multilanguage modules. You must	
	enter the Device ID for the Adapter ROM.	
	The default value for the Segment field is equal to zero. The value indicates	
	that runtime location is dynamic.	

Buttons

lcon	Description	
Browse	This button is used to search for a new module file from any storage location.	
Replace	This button is used to launch the replace module operation.	

Delete Module

This function allows you to remove BIOS module from the BIOS ROM file.

Insert Replace	Delete Extract ROM Info RomHole NCB CPU PATCH	
Module ID:		
Vendor ID:		
Device ID:		
You need to spec module ID 0x20	y Vendor ID and Device ID only if deleting PCI Adapter ROM with Module with ID 0x80 is generated automatically.	
	Delete	

Note: A delete module is no longer available in the BIOS ROM file and cannot be

recovered by using MMTOOL..

Field

Name	Description	
Module ID	2-digits hexadecimal Module ID. See <u>Appendix A Module ID Codes</u> for detail.	
Offset	This filed is used to enter a new module runtime Offset. This field is optional	
Vendor ID	except when deleting an Adapter ROM or Multilanguage modules. You must	
	enter the Vendor ID for the Adapter ROM.	
Segment	This filed is used to enter a new module runtime Segment. This field is optional	
Device ID	except when deleting an Adapter ROM or Multilanguage modules. You must	
	enter the Device ID for the Adapter ROM.	

Buttons

lcon	Description		
Delete This button is used to launch the delete module operation.			
Note: The	original BIOS ROM file is not modified unless you use Save ROM		
but	ton or the Save ROM As button to save changes.		

Extract Module

This function allows you to copy any BIOS module from the BIOS ROM file.

Insert Replace	Delete Extra	t ROM Info RomHole NCB	CPU PATCH]
Module file:			Browse
Module ID: Offset: Segment:		Extract Module C As is in the ROM file C In uncompressed form	
		Extract	

Note: The BIOS Module is saved to selected file.

Name	Description	
Module File	This field is used to specify path/filename of new module file with extension.	
Module ID	2-digits hexadecimal Module ID. See <u>Appendix A Module ID Codes</u> for detail	
Offset	This filed is used to enter a new module runtime Offset. This field is optiona	
Vendor ID	except when extracting an Adapter ROM or Multilanguage modules. You must	
	enter the VendorID for the Adapter ROM.	
Segment	This filed is used to enter a new module runtime Segment. This field is optiona	
Device ID	except when extracting an Adapter ROM or Multilanguage modules. You must	
	enter the DeviceID for the Adapter ROM.	

Buttons

lcon	Description	
Browse	This button is used to search for a new module file from any storage location.	
Extract	This button is used to launch the extract module operation.	

Options

Name	Description
As is in ROM	This option is used to extract module in the same way the module is present in
File	BIOS ROM file.
In	This option is used to extract the module in its original form.
Uncompressed	
Form	

Note: Extracting a BIOS module will not affect the BIOS ROM file.

ROM Information

nsert Replace Delete	Extract ROM	Info RomHole	NCB CPU	PATCH

Note: This sheet displays ROM related information except ROM Hole and NCB.

ROM Hole

Insert Replace Delet	e Extract ROM	Info RomHole	NCB CPU	PATCH]
Name	Location	Size	Image	
				_
Module File		1		Browse
	Insert Image	Save Im	age	

Name	Description
Name	This field displays ROM Hole number in BIOS ROM file.
Location	This field displays the start address of ROM Hole in BIOS ROM file.
Size	This filed display the ROM Hole size.

Name	Description		
Image	If a ROM Hole contains data, this field displays "Yes", or it will be "No".		

Buttons

lcon	Description		
Browse	This button is used to search for a new image file from any storage location.		
Insert Image	This button is used to insert a new image into the marked ROM Hole.		
Save Image	This button is used to save marked ROM Hole content into file.		

NCB Information

Name	Туре	Modules	Start address	Size	Free Space

	Mogetrends
Field	Description
Name	This field displays the NCB Region Name for idenification.
Туре	This field displays the region type. Usually, it is either Extended Boot Block
	region or Generic region.
Modules	This field displays how many modules inside this region.
Start Address	This field displays region's start address in BIOS ROM file.
Size	This field displays the region size in unit of byte.
Free Space	This field diaplays remaining size of the region.

CPU Patch

Insert Repla	ce Delete Extract ROM Info RomHole NCB	CPU PATCH
Patch File		Browse
No.	Option—	
Vender :	🔿 Insert a Patch File	
Total No. :	🔿 Extract a Patch File	Apply
Total Size	🔿 Delete a Patch File	
1010110126		

Name	Description
Patch File	This field is used to specify path/filename of new patch file with extension.

Name	Description		
No.	2-digits decimal patch data number in the CPU Micro Code Patches Module.		
Vendor	Display the CPU manufacturer's name. This is a non-editable field.		
Total No.	Display total patch data number. This is a non-editable field.		
Total Size	Display total patch data size. This is a non-editable field.		

Buttons

lcon	Icon Description		
Browse	This button is used to search for a new patch file from any storage location.		
Apply	This button is used to insert a new image into the marked ROM Hole.		

Module Info Frame

ID	Name	RomLoc	Source size	Size in Rom	9696	RunLoc	NCB
				•			

Field	Description
ID	2-digits hexadecimal Module ID. See <u>Appendix A Module ID Codes</u> for detail.
Name	BIOS module name. See Appendix A Module ID Codes for detail.
RomLoc	Module data location in BIOS ROM image.
Source Size	Original module data source size in unit of bytes
Size in Rom	Actual module data size in unit of bytes in BIOS ROM image.
%%	This field displays the module's compressed ratio. Usually, 0.00 means the module is
	uncompressed and means it is a linked module.
RunLoc	This field displays the address where the module will be uncompressed.
	For PCI Adapter ROM Module(20h), it is VendorID and DeviceID.
	For Multilanguage Module(21h), it is language ID and flags.
NCB	This unique name identifies the Non-Critical Block. If present, the module will be
	inserted to the region.

Functions

To use MMTOOL, you can double-click the executable file icon to open the operating

window and press Load ROM button to load a BIOS ROM file:

Lo	ad ROM	insert []	Replace Delete	Extract ROM	1 Info RomHole	INCB	Teroran	I
Sa	ve ROM	Module f	ile:				Е	lrowse
Save ROM as		Module ID:		For Adapter ROM only		-Insert © Compress Module		
		Seg./DID	:	Link Device II);]	RomRegion		T
					Insert			
ID	Name		RomLoc	Source size	Size in Rom	%%	RunLoc	NCB
ID 08	Name Bootblock	Runtime	RomLoc D000:F848	Source size 0798(01944)	Size in Rom 07AC(01964)	%% 0.00	RunLoc Dynamic	NCB
ID 08 04	Name Bootblock Setup Clien	Runtime	RomLoc D000:F848 D000:B8F4	Source size 0798(01944) 69A2(27042)	Size in Rom 07AC(01964) 3F54(16212)	%% 0.00 40.05	RunLoc Dynamic Dynamic	NCB -
ID 08 04 0C	Name Bootblock Setup Clien ROMID	Runtime t	RomLoc D000:F848 D000:B8F4 D000:B8D8	Source size 0798(01944) 69A2(27042) 0008(00008)	Size in Rom 07AC(01964) 3F54(16212) 001C(00028)	%% 0.00 40.05 0.00	RunLoc Dynamic Dynamic Dynamic	NCB - -
ID 08 04 0C 0E	Name Bootblock Setup Clien ROMID OEM Logo	Runtime	RomLoc D000:F848 D000:B8F4 D000:B8D8 D000:A808	Source size 0798(01944) 69A2(27042) 0008(00008) ADBA(44474)	Size in Rom 07AC(01964) 3F54(16212) 001C(00028) 10D0(04304)	%% 0.00 40.05 0.00 90.32	RunLoc Dynamic Dynamic Dynamic Dynamic	NCB - - -
ID 08 04 0C 0E 1A	Name Bootblock Setup Clien ROMID OEM Logo Small Logo	Runtime t	RomLoc D000:F848 D000:B8F4 D000:B8D8 D000:A808 D000:9810	Source size 0798(01944) 69A2(27042) 0008(00008) ADBA(44474) 3D34(15668)	Size in Rom 07AC(01964) 3F54(16212) 001C(00028) 10D0(04304) 0FF8(04088)	%% 0.00 40.05 0.00 90.32 73.91	RunLoc Dynamic Dynamic Dynamic Dynamic Dynamic	NCB - - - -
ID 08 04 0C 0E 1A 18	Name Bootblock - Setup Clien ROMID OEM Logo Small Logo Display Ma	Runtime t (s) nager	RomLoc D000:F848 D000:B8F4 D000:B8D8 D000:A808 D000:9810 D000:80B8	Source size 0798(01944) 69A2(27042) 0008(00008) ADBA(44474) 3D34(15668) 4019(16409)	Size in Rom 07AC(01964) 3F54(16212) 001C(00028) 10D0(04304) 0FF8(04088) 1758(05976)	%% 0.00 40.05 0.00 90.32 73.91 63.58	RunLoc Dynamic Dynamic Dynamic Dynamic Dynamic Dynamic	NCB - - - - -
ID 08 04 0C 0E 1A 18 19	Name Bootblock - Setup Clien ROMID OEM Logo Small Logo Display Ma Font Modu	Runtime t (s) nager e	RomLoc D000:F848 D000:B8F4 D000:B8D8 D000:A808 D000:9810 D000:80B8 D000:7B08	Source size 0798(01944) 69A2(27042) 0008(00008) ADBA(44474) 3D34(15668) 4019(16409) 1304(04868)	Size in Rom 07AC(01964) 3F54(16212) 001C(00028) 10D0(04304) 0FF8(04088) 1758(05976) 05B0(01456)	%% 0.00 40.05 0.00 90.32 73.91 63.58 70.09	RunLoc Dynamic Dynamic Dynamic Dynamic Dynamic Dynamic Dynamic	NCB - - - - - - -
ID 08 04 0C 0E 1A 18 19 1B	Name Bootblock - Setup Clien ROMID OEM Logo Small Logo Display Ma Font Modul Single Link	Runtime t (s) nager e Arch BIOS	RomLoc D000:F848 D000:B8F4 D000:B8D8 D000:A808 D000:9810 D000:9810 D000:80B8 D000:7B08 B000:60AC	Source size 0798(01944) 69A2(27042) 0008(00008) ADBA(44474) 3D34(15668) 4019(16409) 1304(04868) 42914(272660)	Size in Rom 07AC(01964) 3F54(16212) 001C(00028) 10D0(04304) 0FF8(04088) 1758(05976) 05B0(01456) 21A5C(1378	%% 0.00 40.05 0.00 90.32 73.91 63.58 70.09 49.45	RunLoc Dynamic Dynamic Dynamic Dynamic Dynamic Dynamic Dynamic Dynamic	NCB - - - - - - - - - - - - - -
ID 08 04 0C 0E 1A 18 19 1B 21	Name Bootblock - Setup Clien ROMID OEM Logo Small Logo Display Ma Font Modu Single Link Multi Lang	Runtime t (s) nager e Arch BIOS Jage	RomLoc D000:F848 D000:B8F4 D000:B8D8 D000:A808 D000:9810 D000:9810 D000:30B8 D000:7B08 B000:60AC B000:3A34	Source size 0798(01944) 69A2(27042) 0008(00008) ADBA(44474) 3D34(15668) 4019(16409) 1304(04868) 42914(272660) 5583(21891)	Size in Rom 07AC(01964) 3F54(16212) 001C(00028) 10D0(04304) 0FF8(04088) 1758(05976) 05B0(01456) 21A5C(1378 2678(09848)	%% 0.00 40.05 0.00 90.32 73.91 63.58 70.09 49.45 55.01	RunLoc Dynamic Dynamic Dynamic Dynamic Dynamic Dynamic Dynamic US	NCB - - - - - - - - - - - - - -
ID 08 04 0C 0E 1A 18 19 1B 21 20	Name Bootblock - Setup Clien ROMID OEM Logo Small Logo Display Ma Font Modu Single Link Multi Lang PCI Option	Runtime t (s) nager e Arch BIOS nage ROM	RomLoc D000:F848 D000:B8F4 D000:B8D8 D000:A808 D000:9810 D000:9810 D000:30B8 D000:7B08 B000:60AC B000:3A34 A000:A7F8	Source size 0798(01944) 69A2(27042) 0008(00008) ADBA(44474) 3D34(15668) 4019(16409) 1304(04868) 42914(272660) 5583(21891) E000(57344)	Size in Rom 07AC(01964) 3F54(16212) 001C(00028) 10D0(04304) 0FF8(04088) 1758(05976) 05B0(01456) 21A5C(1378 2678(09848) 923C(37436)	%% 0.00 40.05 0.00 90.32 73.91 63.58 70.09 49.45 55.01 34.72	RunLoc Dynamic Dynamic Dynamic Dynamic Dynamic Dynamic Dynamic US 1002:5834	NCB - - - - - - - - - - - - - - - - -
ID 08 04 0C 0E 1A 18 19 1B 21 20 11	Name Bootblock - Setup Clien ROMID OEM Logo Small Logo Display Ma Font Modu Single Link Multi Lang PCI Option P6 Micro C	Runtime t (s) nager e Arch BIOS nage ROM ode	RomLoc D000:F848 D000:B8F4 D000:B8D8 D000:A808 D000:9810 D000:30B8 D000:7B08 B000:60AC B000:3A34 A000:A7F8 A000:3FD0	Source size 0798(01944) 69A2(27042) 0008(00008) ADBA(44474) 3D34(15668) 4019(16409) 1304(04868) 42914(272660) 5583(21891) E000(57344) 6808(26632)	Size in Rom 07AC (01964) 3F54 (16212) 001C (00028) 10D0 (04304) 0FF8 (04088) 1758 (05976) 05B0 (01456) 21A5C (1378 2678 (09848) 923C (37436) 681C (26652)	%% 0.00 40.05 0.00 90.32 73.91 63.58 70.09 49.45 55.01 34.72 0.00	RunLoc Dynamic Dynamic Dynamic Dynamic Dynamic Dynamic Dynamic US 1002:5834 Dynamic	NCB - - - - - - - - - - - - - - - - - - -
ID 08 04 0C 0E 1A 18 19 1B 21 20 11 06	Name Bootblock - Setup Clien ROMID OEM Logo Small Logo Display Ma Font Modu Single Link Multi Lang PCI Option P6 Micro C SMBIOS	Runtime t (s) nager e Arch BIOS nage ROM ode	RomLoc D000:F848 D000:B8F4 D000:B8D8 D000:A808 D000:9810 D000:30B8 D000:7B08 B000:60AC B000:3A34 A000:A7F8 A000:3FD0 A000:3C60	Source size 0798(01944) 69A2(27042) 0008(00008) ADBA(44474) 3D34(15668) 4019(16409) 1304(04868) 42914(272660) 5583(21891) E000(57344) 6808(26632) 07AD(01965)	Size in Rom 07AC (01964) 3F54 (16212) 001C (00028) 10D0 (04304) 0FF8 (04088) 1758 (05976) 05B0 (01456) 21A5C (1378 2678 (09848) 923C (37436) 681C (26652) 0370 (00880)	%% 0.00 40.05 0.00 90.32 73.91 63.58 70.09 49.45 55.01 34.72 0.00 55.22	RunLoc Dynamic Dynamic Dynamic Dynamic Dynamic Dynamic Dynamic US 1002:5834 Dynamic Dynamic	NCB - - - - - - - - - - - - - - - - - - -

And refer the following steps to manipulate modules:

Inserting Generic Module

You can insert new BIOS module by following steps:

1. Switch to *Insert* tab and click Browse button to specify the new module file location

lities

Or type the path and the file name in the *Module file* field.

Insert Repl	ace Delete Extract ROM Info RomHole	NCB CPU PATCH
Module file: Module ID: Offset/VID: Seg./DID:	D:\Project\VGA.BIN For Adapter ROM only Link Present Link Vendor ID: Link Device ID:	Browse Insert C Compress Module Insert Uncompressed RomRegion
	Insert	

2. Type the new module ID into the *Module ID* field.

Insert Repl	ace Delete Extract ROM Info RomHole NCB CPU PATCH
Module file: Module ID: Offset/VID: Seg/DID:	D:\ProjectWGA.BIN For Adapter ROM only Insert Link Present Link Vendor ID: Link Device ID: RomRegion
	Insert

3. Enter values in the *Offset/VID* and *Segment/DID* fields. These fields are optional except when inserting an Adapter ROM. You must enter the Vendor ID/Device ID for the Adapter ROM. (If the specific module file is compliant with PCI Adapter ROM specification, MMTOOL will find out relative Vendor ID/Device ID and fill in the fields as default value). The default value for *Offset/VID* and *Segment/DID* field is equal to zero. It indicates that runtime location is dynamic.

Insert Replace Delete Extract ROM Info RomHole NCB CPU PATCH
Module file: D:\ProjectWGA.BIN Browse Module ID: 20 For Adapter ROM only Insert © Compress Module Offset/VID: 1111 Link Present © Compress Module © Insert Uncompressed Seg./DID: 2222 Insert Insert Insert

4. Select one of option buttons(*Compress Module* Or *Insert Uncompressed*) to decide how the new module is to be inserted. The default option is *Compress Module*. If you want to insert the module in a Non-Critical region, you can click it to open RomRegion Box and choose one of valid IDs.



- 5. Click Insert button to insert the new module into the BIOS ROM image.
- Note: All fields in the *Insert Module* tab must be filled in properly before the *Insert* button is pressed.

Inserting Linked Module

You can insert new linked module by following steps:

1. Switch to *Insert* tab and click on *Link Present* option button to enter linked module mode.

Insert Replace Delete	Extract ROM Info RomHole	NCB CPU PATCH
Module file:		Browse
Module ID:	For Adapter ROM only Link Present Link Vendor ID: Link Device ID:	Insert C Compress Module Insert Uncompressed RomRegion
Seg./DID:	Insert	

2. Enter values in the *Link Vendor ID* and *Link Device ID* fields. The IDs means the PCI device that shares the same option ROM with an existing device.

Insert Repl	Delete Extract ROM Info RomHole NCB CPU PATCH
Module file: Module ID: Offset/VID: Seg./DID:	For Adapter ROM only For Adapter ROM only Link Present Link Vendor ID: 1111 Link Device ID: 2222 RomRegion
	Insert

3. Enter Vendor ID and Device ID in the *Offset/VID* and *Segment/DID* fields to share the option ROM from existing device.

Insert Repl	ace Delete Extract ROM Info RomHole NCB CPU PATCH
Module file:	Browse
Module ID:	20 For Adapter ROM only Insert 20 Compress Module
Offset/VID:	3333 Link Vendor ID: 1111 C Insert Uncompressed
Seg./DID:	4444 Link Device ID: 2222 RomRegion
	Insert

4. Click Insert button to insert the new module into the BIOS ROM image.

Replacing Module

You can replace BIOS module by following steps:

1. Switch to *Replace* tab and click Browse button to specify the new module file

location Or type the path and the file name in the Module file field.

Insert R	eplace	Delete Extract ROM Info RomHole NCB CPU P	ATCH
Module fil	e:	D:\Project\NewVGA.bin	Browse
Module ID VendorID DeviceID:): :		
		Replace	

2. Type the new module ID into the *Module ID* field Or select the module to be deleted from the module info frame.

Insert Replace	Delete Extract ROM Info RomHole NCB CPU PATCH
Module file:	D:\Project\New\VGA.bin Browse
Module ID: VendorID: DeviceID:	
	Replace

3. If you select the module to be replaced from the module info frame, just ignore this step. Otherwise, enter values in the *Offset/VID* and *Segment/DID* fields. These fields are optional except when replacing an Adapter ROM. You must enter the Vendor ID/Device ID for the Adapter ROM. The default value for *Offset/VID* and *Segment/DID* field is equal to zero. It indicates that runtime location is dynamic.

Insert Replac	Delete Extract ROM Info RomHole NCB CPU PATCH
Module file:	D:\Project\NewVGA.bin Browse
Module ID:	20
VendorID:	1111
DeviceID:	2222
	Replace

4. Click Replace button to replace the existing module with new module file. The new module will be inserted into the BIOS ROM image.

Deleting Module

You can delete BIOS module by following steps:

1. Switch to *Delete* tab and type the module ID into the *Module ID* field Or select the module to be deleted from the module info frame.

Insert Replace	Delete Extract ROM Info RomHole NCB CPU PATCH
Module ID:	20
VendorID:	
DeviceID:	
You need to spec module ID 0x20.	ify Vendor ID and Device ID only if deleting PCI Adapter ROM with Module with ID 0x80 is generated automatically.
	Delete

2. If you select the module to be deleted from the module info frame, just ignore this step. Otherwise, enter values in the *Offset/VID* and *Segment/DID* fields. These fields are optional except when deleting an Adapter ROM. You must enter the Vendor ID/Device ID for the Adapter ROM.

Insert Replace	Delete Extract ROM Info RomHole NCB CPU PATCH				
Module ID:	20				
VendorID:	1111				
DeviceID:	2222				
You need to spec module ID 0x20.	y Vendor ID and Device ID only if deleting PCI Adapter ROM with Module with ID 0x80 is generated automatically.				
	Delete				
TING WE WANT					

- 3. Click ______ button to remove the module from the BIOS ROM image.
- Note: Deleting a BIOS module can cause critical BIOS errors. It can also cause the BIOS to halt the system.

Extracting Module

You can extract BIOS module by following steps:

1. Switch to *Extract* tab and click Browse button to select output module file location

Or type the path and the file name in the *Module file* field.

Insert Replac	e Delete Ext r	act ROM Info RomHole NCB CPU PATCH
Module file:	D:\Project\Save	VGA.bin Browse
Module ID: VendorID: DeviceID:		Extract Module C As is in the ROM file C In uncompressed form
		Extract

2. Type the new module ID into the *Module ID* field Or select the module to be extracted from the module info frame.

© Copyright 2005 American Megatrends, Inc. All rights reserved.

Insert Replace	Delete Extract	ROM Info RomHole NCB	CPU PATCH
Module file:	D:\Project\SaveVG	A.bin	Browse
Module ID: VendorID: DeviceID:		Extract Module C As is in the ROM file In uncompressed form Extract	

3. If you select the module to be deleted from the module info frame, just ignore this step. Otherwise, enter values in the *Offset/VID* and *Segment/DID* fields. These fields are optional except when replacing an Adapter ROM. You must enter the Vendor ID/Device ID for the Adapter ROM. The default value for *Offset/VID* and *Segment/DID* field is equal to zero. It indicates that runtime location is dynamic.

Insert Replace	Delete	act ROM Info RomHole NCB CPU PATCH
Module file:	D:\Project\Save	VGA.bin Browse
Module ID: VendorID: DeviceID:	20 11111 2222	Extract Module C As is in the ROM file C In uncompressed form Extract

- 4. Select one of option buttons(*As is in the ROM file* Or *In uncompressed form*) to decide how the module is to be extracted. The default option is *In uncompressed form*.
- 5. Click Extract button to extract the existing module.
- Note: It is recommended to extract the module in uncompressed form. BIOS module must not be compressed twice.

Inserting image into a ROM Hole

You can insert new image into ROM Hole by following steps:

1. Switch to *ROMHole* tab and select a target ROM Hole on the list.

Insert Replace Delete	Extract ROM	Info RomHole	NCB CPUI	PATCH
Name	Location	Size	Image	
RomHole #0	A0000	8000	NO	
RomHole # 1	A8000	8000	NO	
	-	1	1	
Image File				Browse
Insert Image	Delete Im	age	Save Image	

2. Click Browse button to select input image file location Or type the path and the file name in the *Image File* field.

Insert Replace Delete	Extract ROM	Info RomHole	NCB CPU P.	ATCH	
Name	Location	Size	Image		
RomHole #0	A0000	8000	NO		
RomHole #1	A8000	8000	NO		
Image File DAProject/OptROM.bin Browse					
Insert Image	Delete Ima	ige	Save Image		

3. Click Insert Image button to insert new image into target ROM Hole. If the operation is successful, the *Image* field will be display "YES".

insert Replace Delete	Extract ROM	Info RomHole	NCB CPU	PATCH
Name	Location	Size	Image	
RomHole #0	A0000	8000	NO	
RomHole # 1	A8000	8000	YES	
Image File				Browse
Insert Image	Delete Ims	age	Save Image	

Saving ROM Hole image to file Guide

You can save ROM Hole image to file by following steps:

1. Switch to *ROMHole* tab and select a target ROM Hole on the list.

Insert Replace Delete	Extract ROM	Info RomHole	NCB CPU P	ATCH
Name	Location	Size	Image	
RomHole #0	A0000	8000	NO	
RomHole # 1	A8000	8000	YES	
	1	1	1	
Image File				Browse
Insert Image	Delete Ima	age	Save Image	

2. Click Browse button to select output image file location Or type the path and the file name in the *Image File* field.

Insert Replace Delete	Extract ROM	Info RomHole	NCB CPU F	PATCH]	
Name	Location	Size	Image		
RomHole #0	A0000	8000	NO		
RomHole # 1	A8000	8000	YES		
Image File D:\Project'SaveROM.bin Browse					
Insert Image Delete Image Save Image					

3. Click Save Image button to save image to file.

Deleting ROM Hole image

You can delete ROM Hole image by following steps:

4. Switch to *ROMHole* tab and select a target ROM Hole on the list.

Insert Replace Delete	Extract ROM	Info RomHole	NCB CPU	PATCH
Name	Location	Size	Image	
RomHole #0	A0000	8000	NO	
RomHole # 1	A8000	8000	YES	
Image File				Browse
Insert Image	Delete Ima	age	Save Image	

5. Click Delete Image button to delete target ROM Hole image. If the operation is

successful, the Image field will be display "NO".

Insert Replace Dele	ete Extract ROM	Info RomHole	NCB CPU PATO	CH
Name	Location	Size	Image	
RomHole #0	A0000	8000	NO	
RomHole # 1	A8000	8000	NO	
Image File			Br	owse
Insert Image	Delete Im	age .	Save Image	

Inserting a patch data

You can insert a patch data into CPU MicroCode Patch module by following steps:

1. Switch to *CPUPatch* tab and enable *Insert a Patch Data* at Option block.
| Insert Repl | ace Delete Extract | t ROM Info RomHole NCB | CPU PATCH |
|--------------|--------------------|------------------------------|-----------|
| Patch File | | | Browse |
| No. | | _ Option | 1 |
| Vender : | Intel | Insert a Patch Data | 4 la - 1 |
| Total No. : | 12 | | Appiy |
| Total Size : | 00006800 | C Delete a Patch Data | |

2. Click Browse button to select input patch file location Or type the path and the file name in the *Patch File* field.

Insert Replace Delete Extract ROM Info RomHole NCB CPU PATCH D:\Project\M0123456.pdb Browse Patch File Option -No. 💿 Insert a Patch Data Vender Intel C Extract a Patch Data Apply Total No. : 12 🔿 Delete a Patch Data Total Size : 00006800 Apply Click button to insert the patch data. Itilities

Extracting a patch data

3.

You can extract a patch data from CPU MicroCode Patch module by following steps:

e

1. Switch to CPUPatch tab and enable Extract a Patch Data at Option block.

1



2. Click Browse button to select output patch file location Or type the path and the file name in the *Patch File* field.

Insert Repl	lace Delete Extract	ROM Info RomHole NCB	CPU PATCH
Patch File	D:\Project\M654321(D.pdb	Browse
No.		Option	1
Vender :	Intel	🔿 Insert a Patch Data	
Total No. :	12	 Extract a Patch Data Daluta - Patch Data 	Apply
Total Size :	00006800	UDelete a Fatch Data	

3. Type the new patch number into *No*. field Or select patch data from the CPU MicroCode Patch info frame.

No. [4	01	Option	Apply
Vender : In	ntel	C Insert a Patch Data	
Total No. : 1	2	C Extract a Patch Data	
Total Size : 0	00006800	C Delete a Patch Data	

Deleting a patch data IOS ROM Utilities

You can delete a patch data from CPU MicroCode Patch module by following steps:

1. Switch to *CPUPatch* tab and enable *Delete a Patch Data* at Option block.



2. Type the new patch number into *No*. field Or select patch data from the CPU MicroCode Patch info frame.

4.

Patch File			Browse
No.	01	- Option	1
Vender :	Intel	C Insert a Patch Data	ý pply
Total No. :	12		

3. Click Apply button to delete target patch data.

Saving changes & Exiting

After all necessary operations finished, press Save ROM Or Save ROM es.. button to save new BIOS ROM image to file or all changes will be ignored.



Chapter 3 AMIBCP v3.xx

Overview

AMIBCP stands for American Megatrends BIOS Configuration Program. It provides you, the OEM or system integrator, with an easy way to customize some of the AMIBIOS features without coding. This means that you do not have to contact American Megatrends every time a minor change has to be made in your system's AMIBIOS.

With AMIBCP, you can customize your AMIBIOS. This can speedup system development and allow you a greater degree of freedom in adding or changing system features.

Features

This utility offers following features:

- Configure the AMIBIOS System Setup.
- Edit the Registers Tables.
- Edit PCI IRQ Routing Table.
- View AMIBIOS Features, CPU Microcode Patches, Edit Minor Version Number, Sign on message and OEM data

ican

- View and Edit AMIBIOS String.
- View and Modify AMIBIOS DMI Tables.
- SETUP screen layout Ser Guide

Requirements

Supported Operating System

AMIBCP Utility is supported in the following operating systems:

- Microsoft® Windows® 98
- Microsoft® Windows® ME
- Microsoft® Windows® NT 4.0
- Microsoft® Windows® 2000
- Microsoft® Windows® XP/XP64
- Microsoft® Windows® PE

BIOS Requirements

The loaded BIOS ROM file should have the followings:

• The file MUST be an AMIBIOS ROM file (Core version 8.xx.xx only)

Getting Started

Installation

Copies the *AMIBCP.EXE* executable file to any storage location accessible by the host system and then double-click **AMIBCP** icon Or type **AMIBCP** in command prompt to run.

Main Window

nce Al	MIBCP	v3.00					
File	$\underline{\mathtt{V}}iew$	<u>W</u> indow	<u>A</u> bout				
6		8					
Ready	,					NUM	

Menu Bar

User Guide

The *Menu bar* is located at the top of the AMIBCP window. The *Menu bar* contains the following:

- File drop-down menu
- View drop-down menu
- Window drop-down menu. This menu can be shown only when BIOS ROM loaded.
- About

RCP V3.00		_ 🗆 🗵
<u>File V</u> iew <u>W</u> indow <u>A</u> bout		
🖻 🔒 💡		
J Read v		NUM

© Copyright 2005 American Megatrends, Inc. All rights reserved.

File drop-down menu options

When you click on *File*, the *File* menu drops down as shown in the following:

ROR AMIBCP v3.00			
<u>File V</u> iew <u>W</u> indow	About		
<u>O</u> pen <u>S</u> ave Save <u>A</u> s <u>R</u> eport	Ctrl+O Ctrl+S		
Exit			
			NUM

The *File* drop down menu item are explained in the following table:

File Menu Item List			
Name	Description		
Open	Open an AMIBIOS ROM file.		
Save	Save any changes you have made to the AMIBIOS ROM file.		
Save As	Same feature as Save menu item. In addition, it also allows you to specify		
	the location and to change the existing file name.		
Report	Generates a report for current AMIBIOS ROM file. All of the BIOS		
	information will write to specific path/filename.		
Exit	Quit program.		

View drop-down menu options

When you click on *View*, the *View* menu drops down as shown in the following:

Rog AMIBCP v3.00		
<u>File <u>V</u>iew About</u>		
 ✓ <u>I</u>oolbar ✓ <u>S</u>tatus Bar 		
		NUM //

The *View* drop down menu item are explained in the following table:

View Menu Item List

Name	Description		
Toolbar	Display or hide the <i>Toolbar</i> . The <i>Toolbar</i> is displayed under the <i>Menu bar</i> .		
Status Bar	Display or hide the Status Bar. The Status Bar is displayed at the bottom of the		
	AMIBCP window.		

Window drop-down menu options

When you click on Window, the Window menu drops down as shown in the following:

Real AMIBCP	v3.00		_ 🗆 🗵
<u>File V</u> iew	<u>W</u> indow <u>A</u> bout		
	<u>N</u> ew Window <u>C</u> ascade <u>T</u> ile <u>A</u> rrange Icons		
			NUM /

The Window drop down menu item are	e explained in the following table:
------------------------------------	-------------------------------------

Window Menu Item List				
Name	Description			
New Window	Open current AMIBIOS ROM to a new window.			
Cascade	Arrange the AMIBIOS ROM windows so that they overlap one another.			
Tile	Display AMIBIOS ROM windows at the same time.			
Arrange Icons	Automatically arrange the icons.			

About

When you click on *About*, AMIBCP copyrights information will be shown as below:



Toolbar

The Toolbar is located under the Menu bar. It contains three icens:

nde AMIBCP v3.00		
<u>File V</u> iew <u>A</u> bout		
🖻 🖬 🤶		
J		
Ready		NUM //

The *Toolbar* icons are explained in the following table:

	Toolbar Icon List
lcon	Description
M	Open an AMIBIOS ROM file.
	Save any changes you have made to the AMIBIOS ROM file.
8	Display AMIBCP copyrights information.
_	The automus

Body Frame AIBIOS ROM Utilities

The *Body Frame* is the main frame of AMIBCP. It is located under the *Menu bar* and *Toolbar*. The AMIBCP main functions are displayed in the body frame screen.

AMIBCP v3.00		
<u>File V</u> iew <u>A</u> bout		
🖻 🗄 🤶		
Ready		NUM

Status Bar

The *Status bar* is located under *Body Frame*. The left area of the *Status Bar* describes actions of menu items as you use the arrow keys to navigate through menus. The right area of the *Status Bar* indicates if any of the following keys are latched:

© Copyright 2005 American Megatrends, Inc. All rights reserved.

ltem	Description
CAP	The Caps Lock key is latched down.
NUM	The Num Lock key is latched down.
SCRL	The Scroll Lock key is latched down.

AMIBCP v3.00		_ 🗆 🗵
<u>File V</u> iew <u>A</u> bout		
Ready]	NUM //.

Functions

To use AMIBCP, you can double-click the executable file icon to open Main Window and press 🖾 on *Toolbar* to open an AMIBIOS ROM file. 13

AMIBCP allows you to view and modify the AMIBIOS ROM file image. You can perform various actions using the following configuration tabs: ies 1

- Setup Configuration Tab • ser Guide
- Register Editing Tab •
- PCI IRQ Routing Tab
- **BIOS** Features Tab •
- **BIOS Strings Tab**
- DMI Tables Tab •

AMIBCP v3.00 - [1AAAA000.rom]						
[∑] <u>F</u> ile <u>V</u> iew <u>W</u> indow <u>A</u> bout						_ 2
2 🖬 💡						
etup Configuration Register Edit PCI IF	Q Routing BIOS	Strings BIOS Features DM	II Tables			
✓ (019D) Main ✓ (019E) Advanced ✓ (019F) FCIPnP ✓ (01A0) Boot ✓ (01A1) Security ♥ (01A2) Chipset ✓ (01A3) Power ✓ (01A4) Exit	Handle	Control Group Structures	Show	Access/Use	Failsafe	Optimal
Menu Help String Control Help String						Undo
ady					Γ	NUM

© Copyright 2005 American Megatrends, Inc. All rights reserved.

Setup Configuration Tab

The *Setup Configuration* tab allows you to change the settings for AMIBIOS setup options. To change settings, just double-click on the field/check box what you want to do. Currently, you can perform the following actions:

- Edit the control group structure names
- Show Or hide setup screen and setup questions
- Modify access levels and usage
- Edit failsafe and optimal values
- Change SETUP screen layout

An example of the Setup Configuration tab is shown below:

AMIBCP v3.00 - [1AAAA000.rom]						
[⊈] <u>F</u> ile <u>V</u> iew <u>W</u> indow <u>A</u> bout						
≆ 🖬 💡						
etup Configuration Register Edit PCI IRQ Ro	uting BIOS	Strings BIOS Features DMI	Tables			
🔽 🚞 (019D) Main	Handle	Control Group Structures	Show	Access/Use	Failsafe	Optimal
	(01A3)	Power ACPL Eurotion	Ves	User	E 11 1	- Enabled
	(0065) (0042) (0062)	ACPI Stand by State Re-call VGA BIOS from S3 Suspend Time Out(Minute)	Yes Yes Yes	User User User User	Enabled Disabled Enabled	Disabled Disabled
(01A2) Chipset (01A3) Power (01A3) Power (01A4) Exit	(0063) (0064)	Power Button Function Restore on AC Power Loss	Yes Yes	User User	Power Off Off	Power Off Off
Menu Help String						_
-Control Help String						7
						<u>U</u> ndo
adar.						MIIM

Fields

The Setup Configuration fields are explained in the following table:

Field	Description
Handle	This field displays the setup item's string number in BIOS strings.
	This is a read-only field.
Control Group Structures	This field allows you to modify the setup item's name that appears in
	the AMIBIOS setup screen.
Show	This field allows you to display or hide a particular setup item from
	the AMIBIOS setup screen.
Access/Use	This field allows you to control the access levels and usage of setup
	item.
Failsafe	This field allows you to program the setup item with the safest
	possible settings that can be used if the system behaves erratically.
Optimal	This field allows you to program the setup item with the best system
	performance settings.

Buttons

The Setup Configuration button is explained in the following table:

lcon	Description
<u>U</u> ndo	This button allows you to restore the original setup settings.
	Note:
	This Undo button is used the same way throughout the AMIBCP program.

Menu Help String

This area displays the help string for individual setup menu. When you select a menu item on Setup Menu list, the help message will be shown here.

Control Help String

This area displays the help string for individual setup item. When you select an item on Control Group Structures list, the help message will be shown here.

Setup Structures

The Setup Structures consists of setup controls, such as questions, date, time, password, and setup control group items.

Field/Check Box	Description
Setup Controls	For setup questions, you can modify the show, access/use, failsafe
	and optimal fields. Setup questions strings can be edited.
	For the date, time and password controls, you can modify the show
	and access/use fields.
	Note:
	Setup question strings can be edited or replaced in the BIOS
	Strings tab
Control Group Items	For the separator control group item, you can modify the show field
	and type of separator to display (blank line/single, thin line/double or
	thick line).
	For the static-text control group item, you can modify the show field
	and usage (normal/title).
	For the dynamic-text control group item, you can modify the show
	field and refresh on/off value of the dynamic text (refresh or no
	refresh).
	Note:
	If the refresh option is selected, the text refreshes once per
	second.

Example of Setup Structures are shown in the following table:

Using the Setup Configuration Tab

You can use the Setup Configuration tab by following steps:

1. Select a setup screen and a sub setup screen.

For Example: Advanced -> Super IO Configuration

- Note: Some of the setup screen does not have sub setup screen.
- 2. Click on Show, Access level, Fail-safe or Optional fields to modify setup options.
- When modifying Failsafe/Optiomal fields, some items does have only list of numbers. These items are implemented using AMIBIOS external functions.
 Its required executing AMIBIOS code is used to define the list of all possible options for setup items. These fields are filled with numeric identifiers because external functions are not available after booting.
- 3. To save the changes you have made to the AMIBIOS ROM file, click on the *File* menu bar and select *Save* menu item. You can also click 🖬 icon on *Toolbar* to save the changes you have made.
 - Note: You can lick on the *Undo* button to restore the original setup settings before saving any new changes.

Changing SETUP screen layout in Setup Configuration Tab

You can change SETUP screen layout in Setup Configuration tab by following steps:

Menu Item layout

1. Choose a Menu Item you want to move on left side of *Setup Configuration tab*.

- 2. Click and hold on left button of mouse.
- 3. Drop the chosen Menu Item to new place.

While the dropping operation is under running, you may see following icons:

lcon	Description
	This icon indicates the chosen Menu Item will be subordinate to the focus
տվե	item.
ſ	This icon indicates the chosen Menu Item will be having same rank with the
13	focus item.
0	This icon indicates the chosen Menu Item may not move to the place where
9	you like.

Question Item layout in same Menu

- 1. Choose a Menu Item on left side of *Setup Configuration tab*.
- 2. Choose a Question Item you want to move on right side of *Setup Configuration tab*
- 3. Click and hold on left button of mouse
- 4. Drop the chosen Question Item to new place.

Question Item layout between Menus

- 1. Choose a Menu Item on left side of *Setup Configuration tab*.
- 2. Choose a Question Item you want to move on right side of *Setup Configuration tab.*
- 3. Click right button of mouse to *Cut* the target Question Item.
- 4. Choose the Menu Item you like on left side of *Setup Configuration tab*.
- 5. Click right button of muse on right side of *Setup Configuration tab* to *Paste* the cut Question Item.

Register Edit Tab

The *Register Edit* tab allows you to edit the AMIBIOS register tables. Each table contains register, data, or other fields that can be edited.

Note: Depending on a particular BIOS table, more than two columns can be used.

An example of the *Register Edit* tab is shown below:

Mar AMIBCP v3.00 - [1AAAA000.rom]			<u>_ X</u>
Eile ⊻iew <u>W</u> indow <u>A</u> bout			_ 8 ×
🖻 🖬 🤶			
Setup Configuration Register Edit PCI IRQ Ro	outing BIOS Strings	BIOS Features DMI	Tables
POST/RUNTIME SIO Table	SIO Register	Value	
Onboard PCI IDE Dev#&Func#, Bus# Onboard PCI IDE Init Values	00	00	
BootBlock SIO Table			
V 11 E 15			17-1-
Module ID : 1B			Undo
Ready			NUM

Fields

The *Register Edit* fields are explained in the following table:

Field	Description
Register	This field allows you to change the register address of a particular
	device or chipset.
Data	This field allows you to change the values to be programmed into
	the device or chipset.
Undo	This button allowas you to restore the original register values.

Buttons

The *Register Edit* button is explained in the following table:

Icon	Description
<u>U</u> ndo	This button allows you to restore the original setup settings.

lcon	Description
	Note:
	This Undo button is used the same way throughout the AMIBCP program.

Note: It is not recommended to change chipset registers values without working knowledge about that specific chipset.

Using the Register Edit Tab

You can use the *Register Edit* tab by following steps:

- 1. Click on the *Register* table to be edited from the list of that appear on the left side of screen. A list of register values or other data is displayed on the right side of the *Register Edit* tab screen.
- 2. To edit the fields, simply double-click on the field you want to modify and type in new value.
- 3. To save the changes you have made to the AMIBIOS ROM file, click on the *File* menu bar and select *Save* menu item. You can also click icon on *Toolbar* to save the changes you have made.

Note: You can lick on the *Undo* button to restore the original setup settings before saving any new changes.

PCI IRQ Routing Tab

The *PCI IRQ Routing* tab allows you to view and modify the *PCI IRQ Routing* table that is used by AMIBIOS during POST and runtime.

165

You can view and modify the following fields:

- PCI Bus
- Dev.#
- Int A-B-C-D Reg
- Int A-B-C-D Bitmap
- Phys.Slot

An example of the *PCI Routing* tab is shown below:

PCI Bus Dev.# IntA Reg IntA Bitmap IntB Reg IntB Bitmap IntC Reg IntC Bitmap IntD Reg IntD B 00 88 00 0000 00 0000 03 DCF8 00 0000 00 80 01 DCF8 02 DCF8 03 DCF8 05 DCF8 00 78 02 DCF8 03 DCF8 01 DCF8 00 68 01 DCF8 02 DCF8 03 DCF8 05 DCF8 00 68 01 DCF8 02 DCF8 01 DCF8 00 68 01 DCF8 05 DCF8 01 DCF8 02 DCF8 00 58 03 DCF8 00 0000 00 0000 000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 00000 00000	map Phys.Slot 00 00
00 88 00 0000 00 0000 03 DCF8 00 0000 00 80 01 DCF8 02 DCF8 03 DCF8 05 DCF8 00 78 02 DCF8 03 DCF8 05 DCF8 00 68 01 DCF8 02 DCF8 03 DCF8 01 DCF8 00 68 01 DCF8 02 DCF8 03 DCF8 05 DCF8 00 90 03 DCF8 05 DCF8 01 DCF8 02 DCF8 00 90 03 DCF8 05 DCF8 01 DCF8 02 DCF8 00 58 03 DCF8 00 0000 00 000 000 000 000 000 000 000 000 000 000 000 000 000 000 0000 000	00
00 80 01 DCF8 02 DCF8 03 DCF8 05 DCF8 00 78 02 DCF8 03 DCF8 05 DCF8 01 DCF8 00 68 01 DCF8 02 DCF8 03 DCF8 01 DCF8 10 68 01 DCF8 02 DCF8 03 DCF8 05 DCF8 10 90 03 DCF8 05 DCF8 01 DCF8 02 DCF8 10 90 03 DCF8 05 DCF8 01 DCF8 02 DCF8 10 58 03 DCF8 00 0000 00 0000 000 00000 0000	00
00 78 02 DCF8 03 DCF8 05 DCF8 01 DCF8 00 68 01 DCF8 02 DCF8 03 DCF8 05 DCF8 05 DCF8 05 DCF8 05 DCF8 05 DCF8 00 DCF8 05 DCF8 01 DCF8 02 DCF8 01 DCF8 02 DCF8 01 DCF8 02 DCF8 01 DCF8 02 DCF8 00 0000 00 0000 000 00000 0000 00000	00
00 68 01 DCF8 02 DCF8 03 DCF8 05 DCF8 10 90 03 DCF8 05 DCF8 01 DCF8 02 DCF8 10 90 03 DCF8 05 DCF8 01 DCF8 02 DCF8 10 58 03 DCF8 00 0000 00 0000 00 0000	00
00 90 03 DCF8 05 DCF8 01 DCF8 02 DCF8 00 58 03 DCF8 00 0000 00 0000 00 0000 00 0000 </td <td>00</td>	00
0 58 03 DCF8 00 0000 00 0000 00 0000 00 0000 00 0000	00
0 28 01 DCF8 02 DCF8 03 DCF8 05 DCF8 0 30 02 DCF8 03 DCF8 05 DCF8 01 DCF8	00
0 30 02 DCF8 03 DCF8 05 DCF8 01 DCF8	01
	02
U 38 U5 DCF8 U1 DCF8 U2 DCF8 U3 DCF8	03
0 40 03 DCF8 05 DCF8 01 DCF8 02 DCF8	04
0 50 02 DCF8 03 DCF8 05 DCF8 01 DCF8	05
0 60 02 DCF8 03 DCF8 05 DCF8 01 DCF8	05
0 08 01 DCF8 02 DCF8 00 0000 00 0000	00
0 00 00 0000 00 0000 00 0000 00 0000 00 0000	00

۰

Fields

٠

The *PCI IRQ Routing* fields are explained in the following table:

Field	Description
PCI Bus	This field displays the PCI bus that the device/slot is on.
	Note:
	Most boards contain a single PCI bus, so this field is usually set
	to 0.
Dev.#	This field displays the PCI device/slot number.
	The value of this field is set to a slot or device address on the PCI
	bus shifted left by three bits (the device number must be in bits 7:3
	and bits 2:0 must be 000).
Int A-B-C-D Reg	This field displays the chipset register number that controls the PCI
	slots (or device) Int A, B, C, and D Pin. The value in this field is
	basically arbitrary. Slots and devices that share the same chipset
	interrupt signal must have the same value for this field.
	For example:
	If Slot 1 Int A pin and Slot 2 Int B pin are both connected to the same
	chipset interrupt signal, then the chipset register value for Slot 1 Int
	A must match Slot 2 Int B.
	If a slot or device has nothing connected to its Int A pin, then this
	field must be set to 0.
	If a slot or device has its Int A pin hardwired directly to an IRQ, then
	this field is set to 0Fxh (where x is 0-F for IRQ 0 - IRQ 15). This is
	useful if a motherboard has a PCI IDE chip that has its Int A pin

Field	Description
	hardwired to IRQ 14.
Int A-B-C-D Bitmap	This field displays the IRQ that the chipset is capable of routing to in
	the slots (or device) Int A, B, C, and D pin.
	Note: If the value for Chipset Register is set to 0, then all bits in this
	field are set to 0.
	Note:
	If the value for chipset register was set to 0Fxh to indicate a
	hardwired connection to a certain IRQ, then only one bit
	corresponding to that IRQ must be set in this field.
Phys.Slot	This field displays the slot number of a PCI slot as it appears to the
	end user. Numbers like 1, 2, 3, and 4 must be used.
	Note:
	Onboard PCI devices such as PCI IDE chips must have this
	field set to 0 to indicate that the device is not a removable PCI
	adapter card.

Buttons

The *PCI IRQ Routing* button is explained in the following table:

lcon	Description
<u>U</u> ndo	This button allows you to restore the original setup settings.
	Note:
	This Undo button is used the same way throughout the AMIBCP program.

merican

Using the PCI IRQ Routing Tab

You can use the PCI IRQ Routing tab by following steps:

- 1. To modify any field in the *PCI IRQ Routing* table, simply double-click on the any field you want to modify and type in new value.
 - Note: You can add a new PCI entry into the table if all fields in the entry are set to zeros.
- 2. To save the changes you have made to the AMIBIOS ROM file, click on the *File* menu bar and select *Save* menu item. You can also click 🖬 icon on *Toolbar* to save the changes you have made.

Note: You can lick on the *Undo* button to restore the original setup settings before saving any new changes.

BIOS String Tab

The *BIOS Strings* tab allows you to view and edit AMIBIOS strings. An example of the *BIOS Strings* tab is shown below:

Setup Configuration Register Edit PCI IRQ Routing EIOS Strings DMI Tables Token String	de AMIE 💬 Eile 🧀 📑	CP v3 ⊻iew	.00 - [1. <u>W</u> ind	0AAAA0 .ow <u>A</u> 1	00.rom bout]											
Dx00 4D 75 6C 74 69 2D 42 69 74 20 45 43 43 20 45 A 0x10 72 6F 72 0 <th>Setup C Token 0x000 0x00</th> <th>String 1 Pa 0 M 1 Pa 2 F 3 N 4 Ha 5 Se 6 Da 7 1.2 8 Ba 9 No A 1.2 D Ra D Ra String String</th> <th>ration ring ulti-Bit uity Em BootSec /IRUS: ard Disk elect Boo nknown 2MB F ooting fo o Emula 44MB H 88MB H stem Ha eboot ar RAM Fr</th> <th>Register or tor Writ Continu Image of Image loppy Ir rom CD tion Im Joppy I Joppy I Joppy J Jated d Select requency</th> <th>c Edit ror e II ie (Y/N) : a : mage: ROM w age : mage: mage: mage: t proper</th> <th>PCI IR()? vith Mul</th> <th>Q Routi tiple Bo evice or</th> <th>ot Imag</th> <th>OS Strin œ Boot Me</th> <th>edia in s</th> <th>OS Feat</th> <th>tures I Boot de</th> <th>DMI Tal</th> <th>bles l press a</th> <th>key</th> <th></th> <th></th>	Setup C Token 0x000 0x00	String 1 Pa 0 M 1 Pa 2 F 3 N 4 Ha 5 Se 6 Da 7 1.2 8 Ba 9 No A 1.2 D Ra D Ra String String	ration ring ulti-Bit uity Em BootSec /IRUS: ard Disk elect Boo nknown 2MB F ooting fo o Emula 44MB H 88MB H stem Ha eboot ar RAM Fr	Register or tor Writ Continu Image of Image loppy Ir rom CD tion Im Joppy I Joppy I Joppy J Jated d Select requency	c Edit ror e II ie (Y/N) : a : mage: ROM w age : mage: mage: mage: t proper	PCI IR()? vith Mul	Q Routi tiple Bo evice or	ot Imag	OS Strin œ Boot Me	edia in s	OS Feat	tures I Boot de	DMI Tal	bles l press a	key		
0x60 0x70 0x70	0x00 0x10 0x20 0x30 0x40 0x50 0x60 0x70	4D 72	75 6F	6C 72	74	69	2D	42	69	74	20	45	43	43	20	45	Find Next

Fields

Megatrends The *BIOS String* fields are explained in the following table:

Field	Description
Token	The Token field displays the string handle that is used by AMIBIOS
	to reference the string.
String	The String field displays the AMIBIOS string as it appears in the
	AMIBIOS setup or POST screen. This field is editable.
Find String	This field allows you to find a specific string in BIOS Strings list. The
	string is case-insensitive.

. .

Buttons

The *BIOS String* button is explained in the following table:

lcon	Description
Find Next	This button allows you to find next string in BIOS Strings list.
Undo	This button allows you to restore the original setup settings.
	Note:
	This Undo button is used the same way throughout the AMIBCP program.

Using the BIOS String Tab

You can use the *BIOS String* tab by following steps:

- 1. To modify any string, double-click on it and type in the new string.
- To save the changes you have made to the AMIBIOS ROM file, click on the File 2. menu bar and select *Save* menu item. You can also click **I** icon on *Toolbar* to save the changes you have made.

Note: You can lick on the Undo button to restore the original setup settings before saving any new changes.

BIOS Features Tab

The BIOS Features tab allows you to view and configure some of the AMIBIOS features. You can view the following fields:

merican

egatrends

- 11

Itilities

- **BIOS** Date •
- **BIOS** Name
- Processor •
- Major Version •
- ID String 1 •
- **BIOS Size** •
- **BIOS** Tag .
- **Reference** Number •
- CPU Microcode Update Patchs •
- Sign On Message
- OEM Data(If available)

An example of the BIOS Features tab is shown below: ۳.___ -0 Т

- 1

<u>File V</u> iew <u>W</u> indow <u>A</u> bou	ut					_ 5
3 🖬 💡						
tup Configuration Register E	dit PCI IRQ Routing	BIOS Strings BIOS	Features DMI Tab	les		
IOS Date : 12/25/04	BIOS Tag :	1AAAA000		Major Version :	01	
IOS Name : Demo	Processor :	6		Minor Version :	00	
IOS Size : 512Kb	Reference Nu	mber: 000001				
) String 1 : 63-0100-000001-	00101111-122504-Dei	mo				
CPU Microcode Update patche	\$					
4TT - J. L. D 0 1 C J. L. J	DOM DOOD A ODIT	D DIE D1.# T				1000002
Sign On Message	Treaten Pint					
Sign On Message AMIBIOS(C)2003 Ame	erican Megatre	nds, Inc.				
Sign On Message	erican Megatre	nds, Inc.				
Sign On Message AMIBIOS(C)2003 Ame BIOS Date: 12/25/0	erican Megatre 04 17:24:30 Ve	nds, Inc. r: 08.00.10 				
Sign On Message [AMIBIOS(C)2003 Ame [BIOS Date: 12/25/0 OEM Data	erican Megatre	nds, Inc. r: 08.00.10 				
Sign On Message AMIBIOS(C)2003 Ame BIOS Date: 12/25/0 OEM Data	erican Megatre 04 17:24:30 Ve	nds, Inc. r: 08.00.10 				
Sign On Message AMIBIOS(C)2003 Ame BIOS Date: 12/25/0 OEM Data	erican Megatre 04 17:24:30 Ve	nds, Inc. r: 08.00.10	00 00 00	00 00	00	
Sign On Message AMIBIOS(C)2003 Ame BIOS Date: 12/25/0 OEM Data x00 00 00 00 x00 00 00 00	04 17:24:30 Ve	nds, Inc. r: 08.00.10		00 00		
Sign On Message AMIBIOS(C)2003 Ame BIOS Date: 12/25/0 OEM Data x00 00 00 x00 00 00 00 x00 00 00 00 x00 00 00 00	erican Megatre 04 17:24:30 Ve 00 00 00 00 00 00 00 00 00 00 00	nds, Inc. r: 08.00.10		00 00 00 00 00 00 00 00 00 00 00 00 00		Undo

© Copyright 2005 American Megatrends, Inc. All rights reserved.

Fields

The BIOS Features fields are explained in the following table:

Field	Description
BIOS Date	This field displays the date when the AMIBIOS ROM file was built.
	The value in this field cannot be changed.
BIOS Name	This field displays the name associated with the AMIBIOS ROM file.
	The value in this field cannot be changed.
Processor	This field displays the number that is used to define the processor
	type. The value in this field cannot be changed.
Major Version	This field displays the main AMIBIOS revision number that is used
	with the AMIBIOS release. The value in this field cannot be
	changed.
ID String 1	This field displays the ID string that is associated with the AMIBIOS
	ROM file. The value in this field cannot be changed.
BIOS Size	This field displays the actual size of the AMIBIOS ROM file. The
	value in this field cannot be changed.
BIOS Tag	This field displays the eight-character tag that is associated with the
	AMIBIOS ROM file. The value in this field cannot be changed.
Minor Version	This field displays the minor AMIBIOS revision number that is used
	with the AMIBIOS release. This field is editable.
CPU Microcode Update	This field displays processor patches that are contained in the
Patches	AMIBIOS ROM file. The values in this field cannot be changed.
Sign On Message	This field displays the AMIBIOS sign-on message that is displayed
	during POST. The AMIBIOS copyright string is not editable.
	Note:
	This field must not be more than 175 characters in length.
OEM Data	This field displays the OEM data that is provided by the AMIBIOS.
	This field is 62-bytes long and displayed in two modes (text and
	hexadecimal). This field is editable.
	Note:
	The OEM data area is supported on the AMIBIOS 8.00.08 core
	and later releases.

Buttons

The *BIOS Features* button is explained in the following table:

lcon	Description
<u>U</u> ndo	This button allows you to restore the original setup settings.
	Note:
	This Undo button is used the same way throughout the AMIBCP program.

Using the BIOS Feature Tab

You can use the BIOS Feature tab by following steps:

- 1. To modify any editable field, click on it and type in the new value.
- 2. To save the changes you have made to the AMIBIOS ROM file, click on the *File* menu bar and select *Save* menu item. You can also click 🖬 icon on *Toolbar* to save the changes you have made.

Note: You can lick on the *Undo* button to restore the original setup settings before saving any new changes.

DMI Tables Tab

The *DMI Tables* tab allows you to view and modify AMIBIOS DMI Tables such as BIOS information, system information, baseboard, and so on.

An example of the DMI Tables tab is shown below:

Real AMIBCP v3.00 - [1AAAA000.rom]															_ 🗆 🗵
Eile <u>V</u> iew <u>W</u> indow <u>A</u> bout															
🖻 🔒 💡															
Setup Configuration Register Edit PCI IRQ Ro	outing I	BIOS	Strings	BIC)S Fea	tures	DMI	Tables	1						
0 BIOS Information (Type 0)	System	Info	mation	1											
1 System Information (Type 1) Base Board (or Module) Informatio	Format	tted ar	ea:												
3 System Enclosure or Chassis (Type 4 Processor Information (Type 4) 5 Cache Information (Type 7) 6 Cache Information (Type 7) 7 Cache Information (Type 7) 8 Memory Controller Information (C) 9 Memory Controller Information (C)	0x00 0x10	01 00	19 06	01 00	00	01 00	02 08	03	04 09	00	02	00	03	00	04
9 Memory Module Information (Ty) 10 Memory Module Information (Ty) 11 Memory Module Information (Ty) 12 Memory Module Information (Ty) 13 Port Connector Information (Typ) 14 Port Connector Information (Typ) 15 Port Connector Information (Typ) 16 Port Connector Information (Typ) 17 Port Connector Information (Typ) 18 Port Connector Information (Typ) 20 Port Connector Information (Typ) 21 Port Connector Information (Typ) 22 Port Connector Information (Typ) 23 Port Connector Information (Typ) 24 Port Connector Information (Typ) 25 Port Connector Information (Typ) 26 Port Connector Information (Typ) 27 Port Connector Information (Typ) 28 Port Connector Information (Typ) 29 Port Connector Information (Typ) 20 Port Connector Information (Typ)	Text S	Trings Trc Trc Trc	Be Fi Be Fi Be Fi Be Fi	lled By lled B lled B lled B	yOE. yOE. yOE.	<u>М.</u> М. М. М.									indo
Ready														NUM	

Fields

The DMI Tables fields are explained in the following table:

Field	Description
Formatted area	This field displays the DMI Tables values (in hex). You can modify
	all DMI table values except the first two bytes.
	Note: The first two bytes of the DMI Tables are used to define the
	table type and size.

© Copyright 2005 American Megatrends, Inc. All rights reserved.

Field	Description
Test Strings	This field displays the DMI Tables strings. You can modify these
	strings but you cannot change the number of strings.

Buttons

The DMI Tables button is explained in the following table:

lcon	Description
<< <u>B</u> ack	This button allows you to go to the previous DMI table in the table list.
<u>N</u> ext >>	This button allows you to go to the next DMI table in the table list.
<u>U</u> ndo	This button allows you to restore the original setup settings.
	Note:
	This Undo button is used the same way throughout the AMIBCP program.

Using the DMI Tables Tab

You can use the DMI Tables tab by following steps:

- 1. From the displayed *DMI Tables* list, select the table you want to view.
- 2. To edit the table, simply type in new values in the formatted or text strings areas.
- 3. To save the changes you have made to the AMIBIOS ROM file, click on the *File* menu bar and select *Save* menu item. You can also click icon on *Toolbar* to save the changes you have made.

Note: You can lick on the *Undo* button to restore the original setup settings before saving any new changes.

User Guide

Chapter 4 DMIEDIT v1.xx

Overview

DMIEDIT is a Desktop Management Interface utility with graphical user interface. It provides you an easy way to process SMBIOS data on current host system.

Features

This utility offers the following features:

- Easy to browse all SMBIOS information (Non-AMIBIOS system support).
- Save SMBIOS information to file (Non-AMIBIOS system support).
- Modify and Update SMBIOS information (AMIBIOS system only).

Requirements

Supported Operating System

DMIEDIT Utility is supported in following operating system:

- Microsoft® Windows® 98
- Microsoft® Windows® ME
- Microsoft® Windows® 2000
- Microsoft® Windows® NT 4.0
- Microsoft® Windows® XP/XP64
- Microsoft® Windows® PE

BIOS Requirements

System BIOS should have the followings:

- AMIBIOS CORE version 8.xx.xx.
- *SMIFlash eModule* with "8.00.00_SMIFlash-1.00.07" label or later.
- SMBIOS eModule with "8.00.08_SMB-3.1.02_CORE_RC6" label or later.

Operating System DLL/Driver Requirements

Following files are required by this utility:

- UCOREDLL.DLL
- UCOREVXD.VXD
- UCORESYS.SYS
- UCOREW64.SYS
- AMIBIOS Utility CORE APIs DLL.
 - Driver for Microsoft® Windows® 98/ME.
 - Driver for Microsoft® Windows® NT/2000/XP/PE.
- Driver for Microsoft® Windows® XP64.

Getting Started

Installation

Copies *DMIEDIT.EXE*, *UCOREDLL.DLL*, *UCOREVXD.VXD* and *UCORESYS.SYS* to any storage location accessible by the host system and then double-click **DMIEDIT** icon Or type **DMIEDIT** in command prompt to run. Remember that four files MUST be in same directory.

Main Window

🏡 DMI EDITOR v1.00			_ 🗆 ×	3
<u>File V</u> iew <u>U</u> ndo U <u>p</u> date <u>A</u> bout				
SMBIOS Header	Name	Data Type	Data 🔶	4
🔄 🔚 [Type 0] BIOS Information	SMBIOS Signature	4 BYTEs	_SM_	
[] [Type 1] System Information	SMBIOS Checksum	BYTE	C8h	
📕 🔚 [Type 2] Base Board/Module Information	SMBIOS Table Length	BYTE	31 bytes	
I I (Type 3) System Enclosure or Chassis	SMBIOS Version	WORD	2.3	
I Type 41 Processor Information	SMBIOS Max. Struc. Size	WORD	182	
I Tupe 5] - Memory controller Information	SMBIOS Point Revision	BYTE	00h	
	SMBIOS Formatted Area	5 BYTEs	00 00 00 00 00h	
El ype 6 - Memory Module Information	DMI Signature	5 BYTEs	_DMI_	1
			► I	
Ready			NUM /	1

Menu Bar

The *Menu bar* is located at the top of the DMI Editor window. The *Menu bar* contains the following:

Jser Guide

- File drop-down menu
- View drop-down menu
- Undo drop-down menu
- Update drop-down menu
- About

<u>File View Undo Update About</u>				
SMBIOS Header	Name			-
🛛 🔤 [Type 0] BIOS Information	SMBIOS Signature	4 BYTEs	_SM_	
- E [Type 1] - System Information	SMBIOS Checksum	BYTE		
	SMBIOS Table Length	BYTE	31 bytes	
ITupe 31 Sustem Enclosure or Chassis	SMBIOS Version		2.3	
Tupe 41 - Processor Information	SMBIOS Max. Struc. Size		182	
Tupo El Menory controller Information	SMBIOS Point Revision	BYTE		
	SMBIOS Formatted Area	5 BYTEs		- 23
н Пуре Бј Memory Module Information	DMI Signature	5 BYTEs	_DMI_	-
				<u>۶</u>

File drop-down menu options

赨 DMI EDITO	DR v1.00				
<u>File</u> ⊻iew <u>U</u> r	ndo U <u>p</u> date <u>A</u> bout				
🔒 <u>S</u> ave	Ctrl+S				
🗊 Save All		Name	Data Type	Data	
Exit (Type (Type (Type (Type (Type (Type (Type (Type	nformation 2] Base Board/Module Information 3] System Enclosure or Chassis 4] Processor Information 5] Memory controller Information 6] Memory Module Information	SMBIOS Signature SMBIOS Checksum SMBIOS Table Length SMBIOS Version SMBIOS Max. Struc. Size SMBIOS Point Revision SMBIOS Formatted Area DMI Signature	4 BYTEs BYTE WORD WORD BYTE 5 BYTEs 5 BYTEs	_SM_ C8h 31 bytes 2.3 182 00h 00 00 00 00 00 _DMI_	h •
				NUM	

File Menu Item List					
Name Description					
Save	Save current type information to specific path/filename.				
Save All	Save all type information to specific path/filename.				
Exit	Quit program.				

View drop-down menu options

🎪 DMI EDITOR v1.00					- 🗆 ×
<u>File <u>V</u>iew Undo Update About</u>					
📄 💟 Ioolbar 🦾 🎆					
Status Bar		Name	Data Type	Data	
Refresh Information [Type 2] - Base Board/Module Information [Type 3] - System Enclosure or Chassis [Type 4] - Processor Information [Type 5] - Memory controller Information [Type 6] - Memory Module Information	-	SMBIOS Signature SMBIOS Checksum SMBIOS Table Length SMBIOS Version SMBIOS Max. Struc. Size SMBIOS Point Revision SMBIOS Formatted Area DMI Signature	4 BYTEs BYTE WORD WORD BYTE 5 BYTEs 5 BYTEs	_SM_ C8h 31 bytes 2.3 182 00h 00 00 00 00 00 _DMI_	lh
				NUM	

View Menu Item List					
Name	Description				
Toolbar	Display or hide the Toolbar. The Toolbar is displayed under the Menu bar.				
Status Bar	Display or hide the Status Bar. The Status Bar is displayed at the bottom of the				
	DMI Editor window.				
Refresh	Reload all SMBIOS information from actual BIOS ROM.				

Undo drop-down menu options

File Yiew Undo Update About Image: SME SME ALL Image: SME SME Data Type Data Image: SME SME SME SME Image: SME System Information SME SME Image: Type System Information SME SME Image: Type System Enclosure or Chassis SME SME Image: Type System Enclosure or Chassis SME SME Image: Type System Enclosure or Chassis SME SME SME Image: Type System Enclosure or Chassis SME SME SME SME Image: Type System Enclosure or Chassis SME SME SME SME SME Image: Type System Enclosure or Chassis SME SME SME SME SME SME SME SME<	📩 DMI EDITOR v1.00			_	. 🗆 🗙
Image: Solution of the system information Name Data Type Data All Image: Solution of the system information Image: Solution of the system i	<u>File View</u> <u>Undo</u> U <u>p</u> date <u>A</u> bout				
SME Current Type Image: Type 0] BIOS Information Image: SME Image: Type 0] BIOS Information SMBIOS Signature 4 BYTEs Image: Type 1] System Information SMBIOS Checksum BYTE C8h Image: Type 2] Base Board/Module Information SMBIOS Checksum BYTE C8h Image: Type 3] System Enclosure or Chassis SMBIOS Version WORD 2.3 Image: Type 4] Processor Information SMBIOS Version WORD 2.3 Image: Type 5] Memory controller Information SMBIOS Point Revision BYTE 00h Image: Type 6] Memory Module Information Image: Type 5 BYTEs 00 00 00 00 00 00 00 00 00 MI	🕞 🧊 [🚮 ALL				
Image: Type 0] BIOS Information Image: Type 1] System Information Image: Type 2] Base Board/Module Information Image: Type 3] System Enclosure or Chassis Image: Type 4] Processor Information Image: Type 5] Memory controller Information Image: Type 6] Memory Module Information Image: Type 6] Memory Module Information Image: Type 6] Memory Module Information Image: Type 6] Memory Module Information Image: Type 6] Memory Module Information Image: Type 6] Memory Module Information Image: Type 6] Memory Module Information Image: Type 6] Memory Module Information Image: Type 6] Memory Module Information Image: Type 6] Memory Module Information Image: Type 6] Memory Module Information Image: Type 6] Memory Module Information	SME SME Current Type	Name	Data Type	Data	
	[Type 0] BIOS Information [Type 1] System Information [Type 2] Base Board/Module Information [Type 3] System Enclosure or Chassis [Type 4] Processor Information [Type 5] Memory controller Information [Type 6] Memory Module Information	SMBIOS Signature SMBIOS Checksum SMBIOS Table Length SMBIOS Version SMBIOS Max. Struc. Size SMBIOS Point Revision SMBIOS Formatted Area DMI Signature	4 BYTEs BYTE WORD WORD BYTE 5 BYTEs 5 BYTEs	_SM_ C8h 31 bytes 2.3 182 00h 00 00 00 00 00 _DMI_	h •

Undo Menu Item List			
Name	Description		
ALL	Reload all SMBIOS information from buffer.		
Current Type Reload current type information from buffer.			
USCI GUIUC			

Update drop-down menu options

📩 DMI EDITOR 🖬 .00				. 🗆 🗙
File View Undo Update About				
🖃 🕼 🖄 📶 🔏 ALL				
SMBIOS He The Current Type	Name	Data Type	Data	
[Type 0] BIOS Information	SMBIOS Signature	4 BYTEs	_SM_	
	SMBIOS Checksum	BYTE	C8h	
	SMBIOS Table Length	BYTE	31 bytes	
I Type 31 System Enclosure or Chassis	SMBIOS Version	WORD	2.3	
I Tupe 41 - Processor Information	SMBIOS Max. Struc. Size	WORD	182	
I [Type 4] Processor monitation	SMBIOS Point Revision	BYTE	00h	
	SMBIOS Formatted Area	5 BYTEs	00 00 00 00 00	h
📳 🗒 [Type 6] - Memory Module Information 📃	DMI Signature	5 BYTEs	_DMI_	-
				Þ
			NIL INC.	

Undo Menu Item List			
Name	Description		
ALL	Write all SMBIOS information to actual BIOS ROM.		
Current Type	Write current type information to actual BIOS ROM.		

© Copyright 2004 American Megatrends, Inc. All rights reserved.

About

The About is used to display AMIBCP copyrights information.

🚵 DMI EDITOR v1.00	
<u>File View Undo</u> Update <u>About</u>	
SMBIOS Head About DMI Editor	A
[Type 0] BIC	
I I I I I I I I I I I I I I I I I I I	
Copyright (C) 2004 American Megatrends Inc.	
[
Type 5] Me	1.006
Type 6] Memory Module Information TOMI Signature DMI Signature DMI) UUN
Ready	4

Toolbar

The Toolbar is located under the Menu bar.

📩 DMI EDITOR v1.00			_ [
<u>F</u> ile ⊻iew <u>U</u> ndo U <u>p</u> date <u>A</u> bout				
🖬 🗊 😰 🕺 👬 👬				
SMBIOS Header	Name	Data Type	Data	
[Type 0] BIOS Information [Type 1] System Information [Type 2] Base Board/Module Information [Type 3] System Enclosure or Chassis [Type 4] Processor Information [Type 5] Memory controller Information	SMBIOS Signature SMBIOS Checksum SMBIOS Table Length SMBIOS Version SMBIOS Max. Struc. Size SMBIOS Point Revision SMBIOS Formatted Area	4 BYTEs BYTE BYTE WORD WORD BYTE 5 BYTEs	_SM_ C8h 31 bytes 2.3 182 00h 00 00 00 00 00h	
	MI Signature	5 BYTEs	_DMI_	× ►
Ready			NUM	

There are a number of *Toolbar* icons. These icons allows you easy access to some standard tools used in DMI Editor. The following table describes the *Toolbar* icons in detail.

	Toolbar Icon List			
lcon	Description			
	Save current type information to specific path/filename.			
	Save all type information to specific path/filename.			
¢.	Reload all SMBIOS information from actual BIOS ROM.			
ALL	Reload all SMBIOS information from buffer.			
TYPE	Reload current type information from buffer.			
ALL	Write all SMBIOS information to actual BIOS ROM.			
TYPE	Write current type information to actual BIOS ROM.			

Type Frame

This frame is located under *Menu Bar* and *Toolbar*. It displays identifiable SMBIOS structure types. If a type is unidentifiable, it will display as "[**Type XXX**] – **Unknown Type**". Drop the scroll bar to see more types.

<u>, ()</u> (MI EDIT	'OR v1.00			-	
<u>F</u> ile	⊻iew L	Indo Update <u>A</u> bout				
	a	ALL TYPE ALL THE				
[[B) SMBIC)S Header	Name	Data Type	Data	A
	🗐 (Type	0] BIOS Information	SMBIOS Signature	4 BYTEs	SM	
	🗐 (Type	1] System Information	SMBIOS Checksum	BYTE	C8h	
l	🗐 (Type	2] Base Board/Module Information	SMBIOS Table Length	BYTE	31 bytes	
l		31 System Enclosure or Chassis	SMBIOS Version	WORD	2.3	
		41 Processor Information	SMBIOS Max. Struc. Size	WORD	182	
		5] - Memory controller Information	SMBIOS Point Revision	BYTE	00h	
		C1 Manage Madula Information	SMBIOS Formatted Area	5 BYTEs	00 00 00 00 00 00h	
<u>+</u>		6j Memory Module Information	DMI Signature	5 BYTEs	_DMI_	T
•						Þ
Read	ły				NUM	
			merio	ar	1	

Info Frame

This frame is located under *Menu Bar* and *Toolbar*. It displays current type's information. Drop the scroll bar to see more information.

			1.79	
💑 DMI EDITOR v1.00			_ [
<u>File View Undo Update About</u>				
🖬 🗊 😰 🕅 Ki tipe Ži tipe				
SMBIOS Header	Name	Data Type	Data	
IType 0] BIOS Information	SMBIOS Signature	4 BYTEs	_SM_	
	SMBIOS Checksum	BYTE	C8h	
	SMBIOS Table Length	BYTE	31 bytes	
I Type 31 - System Enclosure or Chassis	SMBIOS Version	WORD	2.3	
I Tupe 41 - Processor Information	SMBIOS Max. Struc. Size	WORD	182	
ITupo 51 Memory controller Information	SMBIOS Point Revision	BYTE	00h	
	SMBIOS Formatted Area	5 BYTEs	00 00 00 00 00 00h	
El Type 6 - Memory Module Information	DMI Signature	5 BYTEs	_DMI_	-
	•			•
Ready			NUM	

Status Bar

The *Status bar* is located under *Type Frame* and *Info Frame*. The left area of the *Status Bar* describes actions of menu items as you use the arrow keys to navigate through menus. The right area of the *Status Bar* indicates if any of the following keys are latched:

ltem	Description
CAP	The Caps Lock key is latched down.

Item	Description			
NUM	The Num Lock key is latched down.			
SCRL	The Scroll Lock key is latched down.			

📩 DMI EDITOR 🖬 1.00					. 🗆 🗙
<u>File View Undo Update About</u>					
🖬 🗊 🖻 🕅 Ki 👬					
SMBIOS Header		Name	Data Type	Data	
[Type 0] BIOS Information [Type 1] System Information [Type 2] Base Board/Module Information [Type 3] System Enclosure or Chassis [Type 4] Processor Information [Type 5] Memory controller Information [Type 6] Memory Module Information	-	SMBIOS Signature SMBIOS Checksum SMBIOS Table Length SMBIOS Version SMBIOS Max. Struc. Size SMBIOS Point Revision SMBIOS Formatted Area DMI Signature	4 BYTEs BYTE BYTE WORD WORD BYTE 5 BYTEs 5 BYTEs	_SM_ C8h 31 bytes 2.3 182 00h 00 00 00 00 00 _DMI_	h
					Þ
Ready				NUM	///

Functions

To use DMIEDIT, you can double-click the executable file icon to open Main Window.

.

For non-AMIBIOS system, **Undo** and **Update** menu will be hidden as below:

🚓 DMI EDITOR v1.00			_ 🗆 ×
<u>File V</u> iew <u>A</u> bout			
SMBIOS Header	Name	Data Type	Data 🔺
- 🗐 [Type 0] BIOS Information	SMBIOS Signature	4 BYTEs	_SM_
🛛 🔚 [Type 1] System Information	SMBIOS Checksum	BYTE	C8h
📕 🗐 [Type 2] Base Board/Module Information	SMBIOS Table Length	BYTE	31 bytes
I Type 31 System Enclosure or Chassis	SMBIOS Version	WORD	2.3
I I Upe 41 Processor Information	SMBIOS Max. Struc. Size	WORD	182
I Tupe 5] - Memory controller Information	SMBIOS Point Revision	BYTE	00h
	SMBIOS Formatted Area	5 BYTEs	00 00 00 00 00h
	DMI Signature	5 BYTEs	_DMI_
	•) I
Ready			NUM ///

Browsing SMBIOS information

Choose a type with single-click on *Type Frame* and then the related information will be displayed at *Info Frame* immediately. Drop the scroll bars to see more types and information.

Saving SMBIOS information to file

- 1. Choose a type what you do like to save on *Type Frame*.
- 2. Open *File* drop-down menu and select *Save* item Or single click 📕 icon on *Toolbar*.

Life Yrew Undo Opdate About Save Ctrl+S XL XL XL Save All Information Name Data Type Data Exit Name Data Type Data Amount If type Opdate Amount String String String Exit Name Data Type Data Amount Amount Amount Exit Name Data Type Data Amount String String String If type Opdate Amount String String	MI EDITOR v1.00			-	
Name Data Type Data Exit Information SMBIOS Signature 4 BYTEs SM_ Image: System Enclosure or Chassis SMBIOS Checksum BYTE C8h Image: Type 31 - System Enclosure or Chassis SMBIOS Version WORD 2.3 Image: Type 51 - Memory controller Information SMBIOS Point Revision BYTE 00h Image: Type 51 - Memory Module Information SMBIOS Formatted Area 5 BYTEs 00 00 00 00h Image: Type 61 - Memory Module Information Image: Type 5 BYTEs 0 00 00 00 00h	File View Undo Update About				
Exit nformation Information Information Image: Type 2] Base Board/Module Information SMBIOS Checksum BYTE C8h Image: Type 3] System Enclosure or Chassis SMBIOS Checksum BYTE 31 bytes Image: Type 4] Processor Information SMBIOS Version WORD 2.3 Image: Type 5] Memory controller Information SMBIOS Point Revision BYTE 00h Image: Type 6] Memory Module Information	😭 Save Ali	Name	Data Type	Data	
	Exit nformation IType 2] Base Board/Module Information IType 3] System Enclosure or Chassis IType 4] Processor Information IType 5] Memory controller Information IType 6] Memory Module Information	SMBIOS Signature SMBIOS Checksum SMBIOS Table Length SMBIOS Version SMBIOS Max. Struc. Size SMBIOS Point Revision SMBIOS Formatted Area DMI Signature	4 BYTEs BYTE WORD WORD BYTE 5 BYTEs 5 BYTEs	_SM_ C8h 31 bytes 2.3 182 00h 00 00 00 00 00h _DMI_	

3. Input path/file name on dialog box and press <u>Save</u> to get information file Or

.

press _	to igi	nore the fund	ction.		
Save As				1	2 ×
Save in: 📝	Desktop	-	💼 🛃		
🚚 My Compu	ter				
My Docum	ents				
Unline Ser	vices				
					- us
					1.2
					_ ties
File <u>n</u> ame:	SMBIOS Header.txt			<u>S</u> ave	
Save as tune:	Document(txt)		-	Cancel	
outo da geo.			لنب		
	Dpen as read-only	1			11.

Saving all SMBIOS information to file

Cancel , , , , , ,

1. Open *File* drop-down menu and select *Save All* item Or single click icon on *Toolbar*.

File View Undo Update About Save Ctrl+S Image: Save All	赨 DMI EDIT	OR v1.00				-	
Save Ctrl+S Xin Mame Data Type Data Exit nformation Name Data Type Data A Exit nformation Information SMBIOS Signature 4 BYTEs SM_ Image: Type 2] Base Board/Module Information SMBIOS Checksum BYTE C8h Image: Type 3] System Enclosure or Chassis SMBIOS Version WORD 2.3 Image: Type 4] Processor Information SMBIOS Max. Struc. Size WORD 2.3 Image: Type 5] Memory controller Information SMBIOS Formatted Area 5 BYTEs 00h Image: Type 6] Memory Module Information Image: Type 5 BYTEs 0MI	<u>F</u> ile ⊻iew U	Indo Update <u>A</u> bout					
Save All Name Data Type Data Exit nformation Information SMBIOS Signature 4 BYTEs SM_ Image: Signature 1 Information SMBIOS Checksum BYTE C8h Image: Signature 1 Information SMBIOS Checksum BYTE C8h Image: Signature 1 Structure 1 Str	📙 <u>S</u> ave	Ctrl+S ZL The					
Exit nformation Image: Imag	🚮 Save All			Name	Data Type	Data	A
	Exit □ (Type □ (Type □ (Type □ (Type □ (Type □ (Type □ (Type	nformation 2] Base Board/Module Information 3] System Enclosure or Chassis 4] Processor Information 5] Memory controller Information 6] Memory Module Information	_	SMBIOS Signature SMBIOS Checksum SMBIOS Table Length SMBIOS Version SMBIOS Max. Struc. Size SMBIOS Point Revision SMBIOS Formatted Area DMI Signature	4 BYTEs BYTE WORD WORD BYTE 5 BYTEs 5 BYTEs	_SM_ C8h 31 bytes 2.3 182 00h 00 00 00 00 00 00h _DMI_	

© Copyright 2004 American Megatrends, Inc. All rights reserved.

2. Input path/file name on dialog box and press <u>Save</u> to get information file Or

press <u>Cancel</u> to ignore the function. The dialog box is same as <u>above</u>.

Updating SMBIOS type

This function is valid only on AMIBIOS system.

1. As SMBIOS Specification, not of all type can be edited. So you have to know which type contains editable item first.

📩 DMI EDITOR v1.00			
<u>File View Undo Update About</u>			
SMBIOS Header	Name	Data Type	Data 🔺
[Type 000] BIOS Information	Struc. Length	BYTE	27 bytes
	Struc. Handle	WORD	0001h
. [Type 002] Base Board/Moc-	Manufacturer	STRING	AMI
I Type 0031 System Enclosur	Product Name	STRING	BIOS
I Tupe 0041 Processor Inform	Version	STRING	8.00
ITupo 0051 Memory controlle	Serial Number	STRING	0123456789
	UUID	16 BYTEs	00 02 00 03 00 04 1
I um [I ype UU6] Memory Module	Wake-up Type	BYTE	06h 💌
₹	•		•
Ready			NUM //

2. Select an item that will be modified.

🝌 DMI EDITOR v1.00				×
<u>File View Undo Update About</u>				
🔲 🕼 🙆 🖍 🖬 📶 👘				
SMBIOS Header	Name	Data Type	Data	
[Type 000] BIOS Information	Struc. Length	BYTE	27 bytes	
[] [Type 001] System Informatio	Struc. Handle	WORD	0001h	
	Manufacturer	STRING	AMI	
IType 003] System Enclosur	Product Name	STRING	BIOS	
I Type 0041 Processor Inform	Version	STRING	8.00	
ITupe 0051 - Memory controlls	Serial Number	STRING	0123456789	
	UUID	16 BYTEs	00 02 00 03 00 04	I
🕒 🖽 [Type UU6] Memory Module 👻	Wake-up Type	BYTE	06h	Ŧ
			Þ	
Ready			NUM	

3. Double-click on the item to enter edit dialog box. The dialog box displays selected item with related information and allows you to modify the value field. After change

the string value, you can press	OK	to submit the change Or	Cancel	to
ignore.				

A DMIEDITOR v1.0 File ⊻iew Undo Up	DD gdate About			
B SMBIOS Heac Image: Type 000] B Image: Type 000] B Image: Type 002] B Image: Type 002] B	DMI Editor [Type 1 Name : Manufacturer String Value : OEM] System Information Type : STRING	OK Cancel	Data 27 bytes 0001h AMI BIOS 8.00 0123456789 20 04 04
	Wake	-ир Туре	BYTE	066h

Open Update drop-down menu and select Current Type item Or single click 👬 icon 4. on *Toolbar*.

📩 DMI EDITOR v1.00				. 🗆 🗙
File View Undo Update About				
	Name	Data Type	Data	
[Type U] BIUS Information [Type 1] System Information	SMBIOS Signature SMBIOS Checksum	4 BYTEs BYTE	_SM_ C8h	
[] [Type 2] Base Board/Module Information [] [Type 3] System Enclosure or Chassis	SMBIUS Table Length SMBIOS Version	WORD	31 bytes 2.3	
[Type 4] Processor Information [Type 5] Memory controller Information	SMBIOS Max. Struc. Size SMBIOS Point Revision SMBIOS Formatted Area	BYTE	00h	h
[Type 6] - Memory Module Information	DMI Signature	5 BYTEs	_DMI_	-
			NUM	

5. Press Yes to confirm the update instruction Or <u>No</u> to ignore.

🚓 DMI EDITOR ¥1.00		<u>- </u>
<u>File View Undo Update About</u>		
🖬 🗇 🖻 🔝 ਜ 👬 🦾 👬	-1	
SMBIOS Header DMI EDITOR) ata Type	Data 🔺
I I I I I I I I I I I I I I I I I I I	BYTE	27 bytes
Type 001] System Informatio 🛛 🖡 🛛 Are you sure to update data ?	WORD	0001h
📘 🔚 [Type 002] Base Board/Moc	TRING	OEM
[Type 003] System Enclosur	TRING	BIOS
ITupe 004] Processor Inform Yes No	\$TRING	8.00
	\$TRING	0123456789
	6 BYTEs	00 02 00 03 00 04 1
U Wake-up Type	BYTE	06h 💌
		•
Ready		NUM //

6. Now is updating data when the instruction is confirmed.

📥 DMI EDITOR v1.00			_ 🗆 🗵
<u>File View Undo Update About</u>			
🖬 🛍 ដ 🖬 📶 👘			
SMBIOS Header	Name	Data Type	Data 🔺
📕 🗐 [Type 000] BIOS Information 🛄	MIEDITOR	BYTE	27 bytes
		WORD	0001h
	Updating	STRING	OEM
Type 003] System Enclosur	op and given	STRING	BIOS
I I Voe 0041 Processor Inform		STRING	8.00
I Tupe 0051 - Memory controlls	Serial Number	STRING	0123456789
	UUID	16 BYTEs	00 02 00 03 00 04 📖
	Wake-up Type	BYTE	06h 🗾
	•		•
Ready			NUM ///

7. Update done. You will get the notice dialog box. Just press to finish the update operation.

🝌 DMI EDITOR v1.00			_ 🗆 ×	1
<u>Eile ⊻iew Undo Update About</u>				
🖬 🕼 🙆 🛍 👬 👬 📶				
SMBIOS Header	DMI EDITOR	Data Type	Data 📩]
📳 🗐 [Type 000] BIOS Information	9	BYTE	27 bytes	
	S Updating DMI data done!	WORD	0001h	
. [Type 002] Base Board/Moc	F I	STRING	OEM	
I Type 0031 System Enclosur	F OK	STRING	BIOS	
I I I I I I I I I I I I I I I I I I I		STRING	8.00	
I Tupe 0051 - Memory controlls	9	STRING	0123456789	
	UUID	16 BYTEs	00 02 00 03 00 04 (
I Home I I ype UU6] - Memory Module	Wake-up Type	BYTE	06h 💌	
			•	
Ready			NUM /	11

Updating all SMBIOS type

This function is valid only on AMIBIOS system.

- 1. Repeat step.1-3 of <u>Update SMBIOS Type</u> to modify items.
- 2. Open *Update* drop-down menu and select *All* item Or single click **X** icon on *Toolbar*.

🚓 DMI EDITOR v1.00			-	
<u>File View Undo</u> Update About				
🔲 🗊 🖄 👷 📶 🕰				
SMBIOS He 🎋 Current Type	Name	Data Type	Data	
[Type 0] BIOS Information [Type 1] System Information [Type 2] Base Board/Module Informatio [Type 3] System Enclosure or Chassis [Type 4] Processor Information [Type 5] Memory controller Information [Type 6] Memory Module Information	SMBIOS Signature SMBIOS Checksum SMBIOS Table Length SMBIOS Version SMBIOS Max. Struc. Size SMBIOS Point Revision SMBIOS Formatted Area DMI Signature	4 BYTEs BYTE WORD WORD BYTE 5 BYTEs 5 BYTEs	_SM_ C8h 31 bytes 2.3 182 00h 00 00 00 00 00h _DMI_	
				Þ
			NEM	

© Copyright 2004 American Megatrends, Inc. All rights reserved.

3. See step.5-7 of <u>Update SMBIOS Type</u> to finish update operation.

Undoing current type and Undoing all

Undo function can restore the original value before you execute the update operation. To undo current type value, open *Undo* drop-down menu and select *Current Type* item Or single click **w** icon on *Toolbar*.

📩 DMI EDITOR v1.00					_ 🗆 🗙
<u>File View</u> <u>Undo</u> U <u>p</u> date <u>A</u> bout					
SME SME Current Type		Name	Data Type	Data	A
 [Type 0] - BIOS Information [Type 1] - System Information [Type 2] - Base Board/Module Information [Type 3] - System Enclosure or Chassis [Type 4] - Processor Information [Type 5] - Memory controller Information [Type 6] - Memory Module Information 	-	SMBIOS Signature SMBIOS Checksum SMBIOS Table Length SMBIOS Version SMBIOS Max. Struc. Size SMBIOS Point Revision SMBIOS Formatted Area DMI Signature	4 BYTEs BYTE WORD WORD BYTE 5 BYTEs 5 BYTEs	_SM_ C8h 31 bytes 2.3 182 00h 00 00 00 00 00 _DMI_	Dh
				NUM	

To undo all type value, open *Undo* drop-down menu and select *All* item Or single click icon on *Toolbar*.

💑 DMI EDITOR v1.00					. 🗆 🗙
<u>File View Undo</u> U <u>p</u> date <u>A</u> bout					
SME SME Current Type		Name	Data Type	Data	
🛛 🔤 [Type 0] BIOS Information		SMBIOS Signature	4 BYTEs	_SM_	
		SMBIOS Checksum	BYTE	C8h	
[Type 2] Base Board/Module Information [Type 3] System Enclosure or Chassis [Type 4] Processor Information		SMBIOS Table Length	BYTE	31 bytes	
		SMBIOS Version	WORD	2.3	
		SMBIOS Max. Struc. Size	WORD	182	
[Type 4] A Hocesser monitation		SMBIOS Point Revision	BYTE	00h	
Type Sja Menoly Controller Information		SMBIOS Formatted Area	5 BYTEs	00 00 00 00 00	h
Em I ype 6 - Memory Module Information	Υ.	DMI Signature	5 BYTEs	_DMI_	-
					Þ
				NUM	

Appendix A Module ID Codes

These are the Module IDs currently used by AMIBIOS8. Note: Module IDs 00, 01, 02, 03, 05, 07, 09, 0A, 0B, 0D, 0F, 12, 13, 14, 16, and 17 were used by previous version of AMIBIOS and must not be assigned to any new module for AMIBIOS8.

Module ID (hex)	Description
04	Setup Engine (Client).
06	DMI Data (SMBIOS Data).
08	BootBlock-POST Interface module.
0C	ROM ID Module (BIOS Tag).
0E	OEM Logo (large) for Silent Boot.
10	ACPI AML.
11	CPU Microcode patches.
15	External Memory Detection module.
18	ADM.
19	ADM Font.
1A	OEM Small Logo.
1B	Main BIOS (SLAB).
1C	BCP Information Module (Created by AMIBCP).
1D	DUAL Logo
1E	INTEL OSB (On Screen Branding)
1F	Currently unassigned
20	PCI Addon ROM (Same for all PCI Option ROMs).
21	Language Module (Same for all languages).
22 - 25	Currently unassigned
26	Source Level Debugger.
27	Source Level Debugger transport layer.
28	BMC Output Redirection Module.
29	MBI File.
2A	MBI Test Pattern.
2B	More than 4GB memory test.
2C – 2D	Currently unassigned
2E	PXE Base ROM.
2F	Serial Redirection module.
30	Parties Logo.
31	NEC CIM Module (Used by AMI Taiwan).
32	NEC battery refresh support (Used by AMI Taiwan).
38	Auto flash EC firmware (Used by AMI Taiwan).
80	BIOS Information Module.
F0-FF	OEM Modules.

Appendix B AFUDOS v3.xx Commands

Usage: AFUDOS /i<ROM File Name> [/o<Save ROM File Name>] [/n] [/p[b][n][c][e]] [/s] [k[N]] [/c[N]] [/q] [/h]

[/t] [/u<ROM File Name>]

Following table lists the description of previous version of AFUDOS commands.

Command	Description
/n	Do not check ROM ID
/pbnce	p – Program main BIOS
	b – Program Boot Block
	n – Program NVRAM
	c – Destroy system CMOS
	e – Program Embedded Controller Block
/k	Program all Non-Critical Block only
/kN	Program N'th Non-Critical Block only (From K0 upto K7)
/s	Leaves signature in BIOS
/q	Silent execution
/h	Print help
/t	Display current system's ROM ID string
/c	Program Main BIOS and all Non-Critical Blocks
/cN	Program Main BIOS and N'th Non-Critical Block(From C0 upto C7)
/srb	Force REBOOT after programming done
/d	Compare ROM file (Skips flashing)
/o <rom file="" name=""></rom>	Save current system BIOS ROM into disk
/u <rom file="" name=""></rom>	Display ROM file's ROM ID string