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:


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

<table>
<thead>
<tr>
<th>Date</th>
<th>Ver.</th>
<th>Description</th>
<th>Editor</th>
</tr>
</thead>
</table>
| 08/23/2004  | 1.00 | - Initial Document.  
- This version contains AMIMMDOS.EXE, AMIOLDOS.EXE and ROMSETUP.EXE user guide.                                                                                                                      | Danny Liu|
| 10/04/2004  | 1.01 | - Add new user guides for AMIDEDOS.EXE, AMIDEWIN.EXE, DMIEDIT and OEMLOGO.EXE.                                                                                                                                | Danny Liu|
| 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.                                                                                                                                                              | Danny Liu|
| 12/29/2004  | 1.04 | - AMIDEDOS, AMIDEWIN and DMIEDIT supports SMBIOS spec up to 2.4.  
- Add new user guides for AMISCE, AMISCEW, AMICMOS, MMTOOL and AMIBCP.                                                                                                                                  | DannyLiu |
| 02/05/2005  | 1.05 | - Correct command syntax for AMIMMDOS.  
- Update BIOS requirement for AMIDEWIN  
- Update BIOS requirement for DMIEDIT.  
- Add new example for AMISCE.  
- Add new description for AFUDOS’s “Ln” option.  
- Corrects rules, example and New Logo file Requirements for AMIOLDOS.  
- Corrects New Logo file requirements for OEMLOGO.                                                                                           | DannyLiu |
| 04/04/2005  | 1.06 | - Add comments for MMTOOL – ROMInfo, ROMHole, 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                                                                                                                                   | DannyLiu |
| 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 |
# Table of Contents

## Part 1. Command Mode

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AMIMMDOS/WIN v1.xx</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>Overview</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>Features</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>Requirements</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>Supported Operating System</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>BIOS Requirements</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Getting Started</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Installation</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Usage</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Examples</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>AMIOOLDOS/WIN v1.xx</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>Overview</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>Features</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>Requirements</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>Supported Operating System</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>BIOS Requirements</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>New Logo File Requirements</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>Getting Started</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>Installation</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>Usage</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>Examples</td>
<td>16</td>
</tr>
<tr>
<td>3</td>
<td>ROMSETUP v1.xx</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>Overview</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>Features</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>Requirements</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>Supported Operating System</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>BIOS Requirements</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>Getting Started</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>Installation</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>Running ROMSETUP program</td>
<td>18</td>
</tr>
<tr>
<td>4</td>
<td>AMIDEDOS v1.xx</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>Overview</td>
<td>21</td>
</tr>
</tbody>
</table>
# Chapter 5... AMIDEWIN v1.xx

**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

**Overview** .......................................................... 28

**Features** ........................................................... 28

**Requirements** .................................................... 28

- Supported Operating System .................................. 28
- BIOS Requirements ............................................... 28
- Operating System Driver Requirements .................... 28

**Getting Started** .................................................. 28

- Installation ................................................................ 28
- Usage ....................................................................... 28
- Examples................................................................... 30

---

# Chapter 7... AFUWIN v4.xx

**Overview** .......................................................... 32

**Features** ........................................................... 32

**Requirements** .................................................... 32

- Supported Operating System .................................. 32
- BIOS Requirements ............................................... 32
- Operating System Driver Requirements .................... 32

**Getting Started** .................................................. 33

- Installation ................................................................ 33
- Usage & Example for command line mode .................. 33
- Main Window ......................................................... 33
- Buttons .................................................................... 34
Function Frame

Functions

Saving system BIOS ROM image to file

Flashing system BIOS with given file

Chapter 8: AMISCE v1.xx/v2.xx

Overview

Features

Requirements

Supported Operating System

BIOS Requirements

Getting Started

Installation

Usage

Examples

Chapter 9: AMISCEW v1.xx/v2.xx

Overview

Features

Requirements

Supported Operating System

BIOS Requirements

Operating System Driver Requirements

Getting Started

Installation

Usage & Example

Chapter 10: AMICMOS v2.xx

Overview

Features

Requirements

Supported Operating System

BIOS Requirements

Getting Started

Installation

Usage

Examples

Part 2. Graphical User Interface Mode

Chapter 1: OEMLOGO v3.xx

Overview

Features
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

© Copyright 2005 American Megatrends, Inc. All rights reserved. - 6 -
Part 1.
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

Requirements

Supported Operating System

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

Usage

AMIMMOS <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.

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>
  Display CPU MicroCode Patch information on screen or save it to file.
- /CPD <CMP ID>
  Delete a CPU MicroCode Patch data.
- /CPE <CMP ID> <Output File Name>
  Extract a CPU MicroCode Patch data to file.
- /CPI <New MicroCode File Name>
  Insert a new CPU MicroCode Patch data.
- /D <Module ID>
  Delete a module.
- /E <Module ID> <Output File Name>
  Extract a module as is in the ROM file.
- /I <Module ID> <Module File Name>
  Insert a module except for linked-module.
- /INFO [Output File Name]
  Display BIOS ROM Information on screen or save it to file.
- /LM <L-VID> <L-DID> <VID> <DID>
  Insert a linked-module.
- /R <Module ID> <Module File Name>
  Replace a module using uncompressed file.
- /RHF <Hole Number> <Hole File Name>
  Fill ROM Hole with given file.
- /SLP [SLP String]
  Read/Write SLP string from/to ROM Image.
- /SM [Sign-On Message]
  Read/Write Sign-On Message from/to ROM Image.
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>`: Set alignment value.
- `/C`: The module cannot be split.
- `/M <M#1> <M#2>`: Set destination address/PCI Device’s vendor/device ID.
- `/N <NCB Name>`: Set NCB region name.
- `/S <Start Address>`: Set start address
- `/U`: Specifies the module is to be placed as uncompressed.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module ID</td>
<td>2-digits hexadecimal Module ID. See Appendix A, Module ID Codes for detail.</td>
</tr>
<tr>
<td>Output File Name</td>
<td>This parameter is used to specify path/filename of the output file with extension.</td>
</tr>
<tr>
<td>Module File Name</td>
<td>This parameter is used to specify path/filename of the Module file with extension.</td>
</tr>
<tr>
<td>SLP String</td>
<td>If present, SLP WRITE function will be enabled. If absent, SLP READ function will be enabled.</td>
</tr>
<tr>
<td>Sign-On Message</td>
<td>If present, Sign-On Message WRITE function will be enabled. If absent, Sign-On Message READ function will be enabled.</td>
</tr>
<tr>
<td>Hole Number</td>
<td>This parameter MUST be a decimal value.</td>
</tr>
<tr>
<td>Hole File Name</td>
<td>This parameter is used to specify path/filename of the Hole file with extension.</td>
</tr>
<tr>
<td>CMP(CPU MicroCde Patch) ID</td>
<td>This ID used for identifying CPU MicroCode Patch data in ROM Image. It has two expresions as below:</td>
</tr>
<tr>
<td>New MicroCode File Name</td>
<td>This parameter is used to specify path/filename of the new CPU MicroCode Patch file with extension.</td>
</tr>
<tr>
<td>L-VID</td>
<td>4-digits hexadecimal linked-vendor ID.</td>
</tr>
<tr>
<td>L-DID</td>
<td>4-digits hexadecimal linked-device ID.</td>
</tr>
<tr>
<td>VID</td>
<td>4-digits hexadecimal vendor ID.</td>
</tr>
<tr>
<td>DID</td>
<td>4-digits hexadecimal device ID.</td>
</tr>
<tr>
<td>8-Bits Value</td>
<td>This parameter MUST be 2-digits hexadecimal value.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| 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 enclosed 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.
- </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 </D>, </E>, </I> and </R> commands.

**Examples**

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

- **Display CPU MicroCode Patch module information**
  
  AMIMMMDOS <BIOS ROM File Name> /CP

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

---

Note: Running AMIMMMDOS under command prompt directly will display help message.
• **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**

• **Inserting a linked-ID module**
  AMIMMDOS <BIOS ROM File Name> /LM <L-VID> <L-DID> <VID> <DID>

• **Display BIOS ROM Information**
  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**

• **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:

- /F
  Force replacement even if the Logo format does NOT be matched.
- /FN
  Both [/F] and [/N] will be enabled.
- /N
  Insert 16-Color BMP without converting it to GRFX(AMI) format.
- /S
  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.

### Rules

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

### 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

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
ROMSETUP, afterward, the usage screen will be displayed:

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

**Note:** The SETUP screen may be different from above map.
5. Choose SETUP screen what you would like to capture and press **Print Screen** to pop-up dialog box:

<table>
<thead>
<tr>
<th>System Overview</th>
<th>Use (ENTER), (TAB) or (SHIFT-TAB) to select a field.</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMIBIOS</td>
<td></td>
</tr>
<tr>
<td>User</td>
<td></td>
</tr>
<tr>
<td>Boot</td>
<td></td>
</tr>
<tr>
<td>Security</td>
<td></td>
</tr>
<tr>
<td>Chioset</td>
<td></td>
</tr>
<tr>
<td>Power</td>
<td></td>
</tr>
<tr>
<td>Exit</td>
<td></td>
</tr>
</tbody>
</table>

![BIOS SETUP UTILITY](image)

6. Input PATH and file name manually on dialog box:

<table>
<thead>
<tr>
<th>Input File Name:</th>
</tr>
</thead>
<tbody>
<tr>
<td>C:\00000010.BMP</td>
</tr>
</tbody>
</table>

![Input File Name](image)

7. Press **ENTER** to save captured screen Or **ESC** to cancel.
8. Repeat step 5 – 7 to get pictures Or press **Ctrl+Break** to quit program.
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

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

AMIDEDOS <Configuration File Name>

Or

AMIDEDOS <Command 1> [Command 2] [Command 3].....
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:

**[System]**
Manufacturer = AMI  
Product = CORE  
Version = 8.00  
SerialNum = 0123456789  
UUID = 0123456789ABCDEF0123456789ABCDEF  
SKUNum = 0123456789  
Family = AMI

**[BaseBoard]**
Manufacturer = AMI  
Product = CORE  
Version = 8.00  
SerialNum = 0123456789

**[Chassis]**
Manufacturer = AMI  
Version = 8.00  
SerialNum = 0123456789  
TagNum = 0123456789  
ChassisType = 03  
ChassisOEM = 0123456789

**[OEMString]**
String = AMI  
String = 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”] Read/Write system manufacturer.
- /SP [“String”] Read/Write system product.
- /SV [“String”] Read/Write system version.
- /SS [“String”] Read/Write system serial number.
- /SU [16 Bytes] Read/Write system UUID.
- /SKU [“String”] Read/Write SKU number.
- /SF [“String”] Read/Write family name.

Part 2. Base Board (Type 2)
- /BM [“String”] Read/Write baseboard manufacturer.
- /BP [“String”] Read/Write baseboard product.
- /BV [“String”] Read/Write baseboard version.
- /BS [“String”] Read/Write baseboard serial number.

Part 3. Chassis (Type 3)
- /CM [“String”] Read/Write chassis manufacturer.
- /CT [8-Bits value] Read/Write chassis type.
- /CV [“String”] Read/Write chassis version.
- /CS [“String”] Read/Write chassis serial number.
- /CA [“String”] Read/Write chassis tag.
- /CO [32-Bits value] Read/Write chassis OEM-defined value.

Part 4. OEM String (Type 11)
- /OS [<Number> <“String”>] Read/Write #th OEM string.

Part 5. Others
- /DMS [Output File Name] Create configuration file. Default file name is “CONFIG.DMS”.

### Parameters List

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

- Any parameter enclosed 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.
- 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.

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 SMBIOS8

- 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
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.
• UCORESYS.SYS  Driver for Microsoft® Windows® NT/2000/XP/PE.
• UCOREW64.SYS  Driver for Microsoft® Windows® XP64.
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 Usage of AMIDEDOS and Example of AMIDEDOS 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 Appendix B  AFUDOS v3.xx Commands)

Requirements

Supported Operating System

This utility is supported by the following operating systems:

• MS-DOS environment

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 Appendix B  AFUDOS v3.xx Commands to know details.
**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**

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
- /U
  - Get and display ROM ID from BIOS ROM file
- /M<MAC Address>
  - Update BootBlock MAC address if exists

**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
- /C
  - Destroy CMOS after update BIOS done
- /E
  - Program Embedded Controller block if present
- /K
  - Program all non-critical blocks
- /Kn
  - Program n’th non-critical block only (n=0 - 7)
- /Q
  - 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 enclosed by `< >` is a mandatory field.
- Any parameter enclosed by `[ ]` is an optional field.
- `<Commands>` cannot co-exist with any `[Options]`.
- 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.

### 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> /O`

- **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 Appendix B. AFUDOS v3.xx Commands)

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:
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
Buttons

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.

Click this button to exit this program.

Function Frame

Information Tab

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

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>OS</td>
</tr>
<tr>
<td>Description</td>
<td>This field displays current O/S version.</td>
</tr>
</tbody>
</table>
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 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.

- **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 Block 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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load Current Failsafe</td>
<td>Enable if you do like to load CMOS failsafe settings from current system after BIOS updated.</td>
</tr>
<tr>
<td>Load ROM File’s Optimal</td>
<td>Enable if you do like to load CMOS optimal settings from current system after BIOS updated.</td>
</tr>
<tr>
<td>Load ROM File’s Failsafe</td>
<td>Enable if you do like to load CMOS failsafe settings from current system after BIOS updated.</td>
</tr>
<tr>
<td>Destroy CMOS Checksum</td>
<td>Enable if you do like to destroy CMOS checksum after BIOS updated. This is default setting in CMOS Options block.</td>
</tr>
</tbody>
</table>

### Non Critical Block

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>Enable if you want to update all Non Critical Blocks.</td>
</tr>
<tr>
<td>1 – 8</td>
<td>Enable one of Non Critical Blocks if it needs to update.</td>
</tr>
</tbody>
</table>

### Miscellaneous

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do Not Check ROM ID</td>
<td>Enable if you do not want to check ROM ID before updating BIOS.</td>
</tr>
<tr>
<td>Restart after Programming</td>
<td>Enable if you want to restart system after BIOS updated.</td>
</tr>
<tr>
<td>Preserve SMBIOS Type</td>
<td>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 convenience, you can strike ‘A’ key as first character to select all SMBIOS structures at once.</td>
</tr>
<tr>
<td>Update MAC</td>
<td>This field is used to change BootBlock MAC address. It MUST be hexadecimal-digit string.</td>
</tr>
</tbody>
</table>

### Progress Tab

This tab displays the updating status.
<table>
<thead>
<tr>
<th>Field</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROM Map</td>
<td></td>
<td>This area displays current updating status.</td>
</tr>
<tr>
<td>Legend</td>
<td></td>
<td>This area illustrates the meaning of color in ROM MAP area.</td>
</tr>
<tr>
<td>Stage</td>
<td></td>
<td>This field displays the stage of updating BIOS.</td>
</tr>
<tr>
<td>Address</td>
<td></td>
<td>This field display the address where block is under working.</td>
</tr>
</tbody>
</table>

**Functions**

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

![AFUWIN v4.10](image)

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 ![Save](image) 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 ![Exit](image) button to exit this program.

**Flashing system BIOS with given file**

1. Press ![Open](image) 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 **Exit** 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
• Display, save and restore current CMOS contents

Requirements

Supported Operating System

AMISCE Utility is supported only in following operating system:

• MS-DOS environment

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>
**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>`  
  Create TEXT script file without overwrite.
- `/OX <Script File Name>`  
  Create TEXT script file with overwrite.
- `/OC <Script File Name>`  
  Same as /O command but more information as CMOS Index register, Mask bits…etc.
- `/I <Script File Name>`  
  Parse TEXT script file and update CMOS.
- `/CR [CMOS Image File Name]`  
  Display/Save CMOS contents.
- `/CW <CMOS Image File Name>`  
  Restore CMOS contents.

**Parameters List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Script File Name</td>
<td>This parameter is used to specify path/filename of the TEXT script file with extension.</td>
</tr>
<tr>
<td>CMOS Image File Name</td>
<td>This parameter is used to specify path/filename of the CMOS Image file with extension.</td>
</tr>
</tbody>
</table>

**Rules**

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

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

**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 3½
  - **MFG Default** = [04]1.44/1.25 MB 3½
  
  // Move "*" to the desired Option
  - **Options** = [00]Not Installed
  - [01]360 KB 5¼
  - [02]1.2 MB 5¼
  - [03]720 KB 3½
  - [*]1.44/1.25 MB 3½
  - [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]Auto
  - **MFG Default** = [01]Keyboard
  
  // Move "*" to the desired Option
  - **Options** = <USB Function = Disabled, Enabled>
  - [00]Disabled
  - [01]Keyboard
  - [*]2[02]Auto
  - [03]Keyb+Mouse

- **Child with Multiple Option Set**

  - **Setup Question** = PCI0 Agent To Aperture Access
  - **Token** = 0085 // Do NOT change this line
  - **BIOS Default** = [00]N/A
  - **MFG Default** = [00]N/A
// Move "*" to the desired Option
Options
= <Aperture Access Enable = Disabled>
  = *[00]N/A
<Aperture Access Enable = Enabled>
  = *[00]Enabled
  = [01]Disabled

■ Value
  Setup Question = L1/L2 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**
  AMISC /O <Script File Name>

- **Create TEXT script file and overwrite if the file existed**
  AMISC /OX <Script File Name>

- **Create new TEXT script file to get CMOS index reg. and mask bits information**
  AMISC /OC <Script File Name>

- **Display CMOS contents**
  AMISC /CR

- **Save CMOS contents to file**
  AMISC /CR <CMOS Image File Name>

- **Restore CMOS contents**
  AMISC /CW <CMOS Image File Name>

- **Update CMOS contents by TEXT script file**
  AMISC /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 Usage of AMISCE and Example of AMISCE 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.

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

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
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 Name | This parameter is used to specify path/filename of the CMOS contents file with extension. |

<table>
<thead>
<tr>
<th>Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Any parameter enclosed by &lt; &gt; is a mandatory field.</td>
</tr>
<tr>
<td>• Any parameter enclosed by [ ] is an optional field.</td>
</tr>
</tbody>
</table>

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

Click this button to search for BIOS ROM file from any disk drive.

Click this button to extract Logo Image from ROM to any disk drive by given PATH/FileName.

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

Click this button to replace an existing BIOS Logo module inside the BIOS ROM file.

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

OEM Logo ID=0x0E
Choose this option can change OEM logo module.

Do not convert GRFX
This option is visible only for when OEM Logo is enabled.
If this option to be enabled, the 16-Color BMP image will NOT convert to GRFX format.

Small Logo ID=0x1A
Choose this option can change small logo module.

Functions

To use OEMLOGO, you can double-click the executable file icon to open the operating window:
And refer to the following steps to change new logo:

**Changing OEM Logo**

1. Click the **Load ROM** button to load the BIOS ROM file which contained OEM Logo Module(0Eh) from disk drive.

   ![Image of OEM Logo software interface](image1)

   - Load AMIEICS 8.xx ROM file to modify
   - Load ROM
   - Select the logo module you want to replace:
     - OEM Logo ID = 0x0E
     - Small Logo ID = 0x1A
   - Select 640x480, 16-color BMP file; 640x480, 256-color BMP file; 640x480, 800x600, 1024x768 JPEG file; or 640x480, 256-color PCX file
   - Press **Save Image** to save the image
   - Press **Close** to close the window

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

   ![Image of OEM Logo software interface](image2)

   - Load AMIEICS 8.xx ROM file to modify
   - D:\Project1AAAA003.rom
   - Select the logo module you want to replace:
     - OEM Logo ID = 0x0E
     - Small Logo ID = 0x1A
   - Select 640x480, 16-color BMP file; 640x480, 256-color BMP file; 640x480, 800x600, 1024x768 JPEG file; or 640x480, 256-color PCX file
   - D:\Project6OEMLOGO.jpg
   - Press **Save ROM As** to save the ROM
   - Press **Close** to close the window

3. Click the **Replace Image** button to change logo module. If the function works fine, *New logo is created* message will be displayed behind the button.
4. Click **Save ROM As** button to save new BIOS ROM file to disk drive.

5. Click **Close** button to exit program.

**Changing Small Logo**

1. Click **Load ROM** button to load the BIOS ROM file which contained Small Logo Module(1Ah) from disk drive and choose **Small Logo ID=0x1A** option.

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

4. Click the **Save ROM As** button to save new BIOS ROM file to disk drive.

5. Click the **Close** 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.

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

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

![Main Window Screenshot](image)

Main Buttons

- **Load ROM**
  Click this button to search for BIOS ROM file from any disk drive.

- **Save ROM**
  Click this button to save the changes you have made to BIOS ROM file that is currently opened.

- **Save ROM as**
  Click this button to save the changes you have made to the BIOS ROM file. You can also specify the location and to change the
Function Frame

Insert Module

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

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Module File</strong></td>
<td>This field is used to specify path/filename of new module file with extension.</td>
</tr>
<tr>
<td><strong>Module ID</strong></td>
<td>2-digits hexadecimal Module ID. See <a href="#">Appendix A, Module ID Codes</a> for detail.</td>
</tr>
<tr>
<td><strong>Offset/VID</strong></td>
<td>This field 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.</td>
</tr>
<tr>
<td><strong>Segment/DID</strong></td>
<td>This field 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.</td>
</tr>
<tr>
<td><strong>Link Vendor ID</strong></td>
<td>This field is used to enter the PCI vendor ID for the PCI device that uses the option ROM. <strong>Note:</strong> This filed must be filled only if <strong>Link Present</strong> check box is selected. You must enter the vendor ID of the PCI device that shares the same option ROM with an existing device.</td>
</tr>
<tr>
<td><strong>Link Device ID</strong></td>
<td>This field is used to enter the PCI device ID for the PCI device that uses the option ROM. <strong>Note:</strong> This filed must be filled only if <strong>Link Present</strong> check box is selected. You must enter the device ID of the PCI device that shares the same option ROM with an existing device.</td>
</tr>
</tbody>
</table>

Note: This file 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.
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.

<table>
<thead>
<tr>
<th><strong>Buttons</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Icon</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>Browse</td>
<td>This button is used to search for a new module file from any storage location.</td>
</tr>
<tr>
<td>Insert</td>
<td>This button is used to launch the insert module operation.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Options</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>Link Present</td>
<td>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.</td>
</tr>
<tr>
<td>Compress Module</td>
<td>This option is used to insert the new module in compact form. <strong>Note:</strong> Some modules MUST be uncompressed, for example: BootBlock- Runtime interface, CPU MicroCode Or ROMID.</td>
</tr>
<tr>
<td>Insert Uncompressed</td>
<td>This option is used to insert the module in its original form.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module File</td>
<td>This field is used to specify path/filename of new module file with extension.</td>
</tr>
<tr>
<td>Module ID</td>
<td>2-digits hexadecimal Module ID. See <a href="#">Appendix A  Module ID Codes</a> for detail.</td>
</tr>
<tr>
<td>Offset</td>
<td>This field is used to enter a new module runtime Offset. This field is optional 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.</td>
</tr>
<tr>
<td>Vendor ID</td>
<td>This field is used to enter a new module runtime Offset. This field is optional 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.</td>
</tr>
<tr>
<td>Segment</td>
<td></td>
</tr>
<tr>
<td>Device ID</td>
<td></td>
</tr>
</tbody>
</table>

Buttons

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="#" alt="Browse" /></td>
<td>This button is used to search for a new module file from any storage location.</td>
</tr>
<tr>
<td><img src="#" alt="Replace" /></td>
<td>This button is used to launch the replace module operation.</td>
</tr>
</tbody>
</table>

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
```

<table>
<thead>
<tr>
<th>Module ID</th>
<th>Vendor ID</th>
<th>Device ID</th>
</tr>
</thead>
</table>

You need to specify Vendor ID and Device ID only if deleting PCI Adapter ROM with module ID 0x20. Module with ID 0x80 is generated automatically.

Note: A delete module is no longer available in the BIOS ROM file and cannot be
recovered by using MMTOOL.

**Field**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module ID</td>
<td>2-digits hexadecimal Module ID. See <a href="#">Appendix A, Module ID Codes</a> for detail.</td>
</tr>
<tr>
<td>Offset</td>
<td>This field is used to enter a new module runtime Offset. This field is optional except when deleting an Adapter ROM or Multilanguage modules. You must enter the Vendor ID for the Adapter ROM.</td>
</tr>
<tr>
<td>Vendor ID</td>
<td>This field is used to enter a new module runtime Segment. This field is optional except when deleting an Adapter ROM or Multilanguage modules. You must enter the Device ID for the Adapter ROM.</td>
</tr>
</tbody>
</table>

**Buttons**

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="delete_icon" alt="Delete" /></td>
<td>This button is used to launch the delete module operation.</td>
</tr>
</tbody>
</table>

**Extract Module**

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

**Field**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module File</td>
<td>This field is used to specify path/filename of new module file with extension.</td>
</tr>
<tr>
<td>Module ID</td>
<td>2-digits hexadecimal Module ID. See <a href="#">Appendix A, Module ID Codes</a> for detail.</td>
</tr>
<tr>
<td>Offset</td>
<td>This field is used to enter a new module runtime Offset. This field is optional except when extracting an Adapter ROM or Multilanguage modules. You must enter the Vendor ID for the Adapter ROM.</td>
</tr>
<tr>
<td>Vendor ID</td>
<td>This field is used to enter a new module runtime Segment. This field is optional except when extracting an Adapter ROM or Multilanguage modules. You must enter the Device ID for the Adapter ROM.</td>
</tr>
</tbody>
</table>

Note: The BIOS Module is saved to selected file.
### Buttons

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="button.png" alt="Browse" /></td>
<td>This button is used to search for a new module file from any storage location.</td>
</tr>
<tr>
<td><img src="button.png" alt="Extract" /></td>
<td>This button is used to launch the extract module operation.</td>
</tr>
</tbody>
</table>

### Options

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>As is in ROM File</td>
<td>This option is used to extract module in the same way the module is present in BIOS ROM file.</td>
</tr>
<tr>
<td>In Uncompressed Form</td>
<td>This option is used to extract the module in its original form.</td>
</tr>
</tbody>
</table>

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

### ROM Information

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

### ROM Hole

<table>
<thead>
<tr>
<th>Name</th>
<th>Location</th>
<th>Size</th>
<th>Image</th>
</tr>
</thead>
</table>

Module File

Insert Image | Save Image

### Field

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>This field displays ROM Hole number in BIOS ROM file.</td>
</tr>
<tr>
<td>Location</td>
<td>This field displays the start address of ROM Hole in BIOS ROM file.</td>
</tr>
<tr>
<td>Size</td>
<td>This field displays the ROM Hole size.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Image</td>
<td>If a ROM Hole contains data, this field displays “Yes”, or it will be “No”.</td>
</tr>
</tbody>
</table>

**Buttons**

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Browse</td>
<td>This button is used to search for a new image file from any storage location.</td>
</tr>
<tr>
<td>Insert</td>
<td>This button is used to insert a new image into the marked ROM Hole.</td>
</tr>
<tr>
<td>Save</td>
<td>This button is used to save marked ROM Hole content into file.</td>
</tr>
</tbody>
</table>

**NCB Information**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>This field displays the NCB Region Name for identification.</td>
</tr>
<tr>
<td>Type</td>
<td>This field displays the region type. Usually, it is either Extended Boot Block region or Generic region.</td>
</tr>
<tr>
<td>Modules</td>
<td>This field displays how many modules inside this region.</td>
</tr>
<tr>
<td>Start Address</td>
<td>This field displays region’s start address in BIOS ROM file.</td>
</tr>
<tr>
<td>Size</td>
<td>This field displays the region size in unit of byte.</td>
</tr>
<tr>
<td>Free Space</td>
<td>This field displays remaining size of the region.</td>
</tr>
</tbody>
</table>

**CPU Patch**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patch File</td>
<td>This field is used to specify path/filename of new patch file with extension.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="#" alt="Browze" /></td>
<td>This button is used to search for a new patch file from any storage location.</td>
</tr>
<tr>
<td><img src="#" alt="Apply" /></td>
<td>This button is used to insert a new image into the marked ROM Hole.</td>
</tr>
</tbody>
</table>

**Module Info Frame**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ID</strong></td>
<td>2-digits hexadecimal Module ID. See <a href="#">Appendix A Module ID Codes</a> for detail.</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td>BIOS module name. See <a href="#">Appendix A Module ID Codes</a> for detail.</td>
</tr>
<tr>
<td><strong>RomLoc</strong></td>
<td>Module data location in BIOS ROM image.</td>
</tr>
<tr>
<td><strong>Source Size</strong></td>
<td>Original module data source size in unit of bytes.</td>
</tr>
<tr>
<td><strong>Size in Rom</strong></td>
<td>Actual module data size in unit of bytes in BIOS ROM image.</td>
</tr>
<tr>
<td><strong>%%</strong></td>
<td>This field displays the module’s compressed ratio. Usually, 0.00 means the module is uncompressed and -- means it is a linked module.</td>
</tr>
<tr>
<td><strong>RunLoc</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>NCB</strong></td>
<td>This unique name identifies the Non-Critical Block. If present, the module will be inserted to the region.</td>
</tr>
</tbody>
</table>

**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:
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

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

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

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 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.

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.

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

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.

<table>
<thead>
<tr>
<th>Insert</th>
<th>Replace</th>
<th>Delete</th>
<th>Extract</th>
<th>ROM Info</th>
<th>RomHole</th>
<th>NCB</th>
<th>CPU PATCH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module file:</td>
<td>Project\NewVGA.bin</td>
<td>Module ID:</td>
<td></td>
<td>VendorID:</td>
<td>DeviceID:</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Insert</th>
<th>Replace</th>
<th>Delete</th>
<th>Extract</th>
<th>ROM Info</th>
<th>RomHole</th>
<th>NCB</th>
<th>CPU PATCH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module file:</td>
<td>Project\NewVGA.bin</td>
<td>Module ID:</td>
<td>20</td>
<td>VendorID:</td>
<td>DeviceID:</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Insert</th>
<th>Replace</th>
<th>Delete</th>
<th>Extract</th>
<th>ROM Info</th>
<th>RomHole</th>
<th>NCB</th>
<th>CPU PATCH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module file:</td>
<td>Project\NewVGA.bin</td>
<td>Module ID:</td>
<td>20</td>
<td>VendorID:</td>
<td>DeviceID:</td>
<td>1111</td>
<td>2222</td>
</tr>
</tbody>
</table>

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.
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.

3. Click the **Delete** 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 the **Browse** button to select output module file location

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

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

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

![Image File dialog]

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”.

![Image Field updated]

**Saving ROM Hole image to file**

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.

![ROMHole selection]

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

5. Click **Delete Image** button to delete target ROM Hole image. If the operation is successful, the **Image** field will be display “NO”.

### 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.
2. Click  button to select input patch file location or type the path and the file name in the Patch File field.

3. Click  button to insert the patch data.

**Extracting a patch data**

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

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

2. Click  button to select output patch file location or type the path and the file name in the Patch File field.
3. Type the new patch number into No. field Or select patch data from the CPU MicroCode Patch info frame.

4. Click button to extract target patch data.

Deleting a patch data

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

1. Switch to CPU Patch 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.
3. Click **Apply** button to delete target patch data.

**Saving changes & Exiting**

After all necessary operations finished, press **Save ROM** or **Save ROM as...** 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.
- View and Edit AMIBIOS String.
- View and Modify AMIBIOS DMI Tables.
- SETUP screen layout.

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)
• BIOS ROM file should be building via “8.00.08_AMITOOLS_17” label or above.

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

Menu Bar

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
File drop-down menu options

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

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

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

View drop-down menu options

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

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Toolbar</td>
<td></td>
</tr>
<tr>
<td>Status Bar</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Toolbar</td>
<td>Display or hide the Toolbar. The Toolbar is displayed under the Menu bar.</td>
</tr>
<tr>
<td>Status Bar</td>
<td>Display or hide the Status Bar. The Status Bar is displayed at the bottom of the AMIBCP window.</td>
</tr>
</tbody>
</table>

**Window drop-down menu options**

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

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Window</td>
<td>Open current AMIBIOS ROM to a new window.</td>
</tr>
<tr>
<td>Cascade</td>
<td>Arrange the AMIBIOS ROM windows so that they overlap one another.</td>
</tr>
<tr>
<td>Tile</td>
<td>Display AMIBIOS ROM windows at the same time.</td>
</tr>
<tr>
<td>Arrange Icons</td>
<td>Automatically arrange the icons.</td>
</tr>
</tbody>
</table>

**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 icons:

![Toolbar Image](image)

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

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>📞</td>
<td>Open an AMIBIOS ROM file.</td>
</tr>
<tr>
<td>📊</td>
<td>Save any changes you have made to the AMIBIOS ROM file.</td>
</tr>
<tr>
<td>📚</td>
<td>Display AMIBCP copyrights information.</td>
</tr>
</tbody>
</table>

**Body Frame**

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.

![Body Frame Image](image)

**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:
<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAP</td>
<td>The <strong>Caps Lock</strong> key is latched down.</td>
</tr>
<tr>
<td>NUM</td>
<td>The <strong>Num Lock</strong> key is latched down.</td>
</tr>
<tr>
<td>SCRL</td>
<td>The <strong>Scroll Lock</strong> key is latched down.</td>
</tr>
</tbody>
</table>

**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.

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

- Setup Configuration Tab
- Register Editing Tab
- PCI IRQ Routing Tab
- BIOS Features Tab
- BIOS Strings Tab
- DMI Tables Tab
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:

Fields

The Setup Configuration fields are explained in the following table:

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

The *Setup Configuration* button is explained in the following table:

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Undo</td>
<td>This button allows you to restore the original setup settings.</td>
</tr>
</tbody>
</table>

**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.

Example of Setup Structures are shown in the following table:

<table>
<thead>
<tr>
<th>Field/Check Box</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setup Controls</td>
<td>For setup questions, you can modify the show, access/use, failsafe and optimal fields. Setup questions strings can be edited.</td>
</tr>
<tr>
<td></td>
<td>For the date, time and password controls, you can modify the show and access/use fields.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>Setup question strings can be edited or replaced in the BIOS Strings tab</td>
</tr>
<tr>
<td>Control Group Items</td>
<td>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).</td>
</tr>
<tr>
<td></td>
<td>For the static-text control group item, you can modify the show field and usage (normal/title).</td>
</tr>
<tr>
<td></td>
<td>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).</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>If the refresh option is selected, the text refreshes once per second.</td>
</tr>
</tbody>
</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.

   Note: When modifying Failsafe/Optional 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:

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Icon1]</td>
<td>This icon indicates the chosen Menu Item will be subordinate to the focus item.</td>
</tr>
<tr>
<td>![Icon2]</td>
<td>This icon indicates the chosen Menu Item will be having same rank with the focus item.</td>
</tr>
<tr>
<td>![Icon3]</td>
<td>This icon indicates the chosen Menu Item may not move to the place where you like.</td>
</tr>
</tbody>
</table>

- **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 mouse 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:

![Register Edit Tab Example](image)

**Fields**

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

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register</td>
<td>This field allows you to change the register address of a particular device or chipset.</td>
</tr>
<tr>
<td>Data</td>
<td>This field allows you to change the values to be programmed into the device or chipset.</td>
</tr>
<tr>
<td>Undo</td>
<td>This button allows you to restore the original register values.</td>
</tr>
</tbody>
</table>

**Buttons**

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

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Undo Icon" /></td>
<td>This button allows you to restore the original setup settings.</td>
</tr>
<tr>
<td>Icon</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>Note:</td>
<td>This <em>Undo</em> button is used the same way throughout the AMIBCP program.</td>
</tr>
</tbody>
</table>

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 click 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.

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:
Fields

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

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCI Bus</td>
<td>This field displays the PCI bus that the device/slot is on.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Most boards contain a single PCI bus, so this field is usually set to 0.</td>
</tr>
<tr>
<td>Dev.#</td>
<td>This field displays the PCI device/slot number.</td>
</tr>
<tr>
<td></td>
<td>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).</td>
</tr>
<tr>
<td>Int A-B-C-D Reg</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td><strong>For example:</strong> 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.</td>
</tr>
<tr>
<td></td>
<td>If a slot or device has nothing connected to its Int A pin, then this field must be set to 0.</td>
</tr>
<tr>
<td></td>
<td>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</td>
</tr>
</tbody>
</table>
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hardwired to IRQ 14.</td>
<td></td>
</tr>
<tr>
<td>Int A-B-C-D Bitmap</td>
<td>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.</td>
</tr>
<tr>
<td>Phys.Slot</td>
<td>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.</td>
</tr>
</tbody>
</table>

### Buttons

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

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="undo_icon" alt="Undo" /></td>
<td>This button allows you to restore the original setup settings. Note: This Undo button is used the same way throughout the AMIBCP program.</td>
</tr>
</tbody>
</table>

### 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 ![Save](save_icon) 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:
The BIOS String fields are explained in the following table:

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

The BIOS String button is explained in the following table:

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Find Next</strong></td>
<td>This button allows you to find next string in BIOS Strings list.</td>
</tr>
<tr>
<td><strong>Undo</strong></td>
<td>This button allows you to restore the original setup settings.</td>
</tr>
</tbody>
</table>

**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.
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 click 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:

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

An example of the *BIOS Features* tab is shown below:
### Fields

The *BIOS Features* fields are explained in the following table:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIOS Date</td>
<td>This field displays the date when the AMIBIOS ROM file was built. The value in this field cannot be changed.</td>
</tr>
<tr>
<td>BIOS Name</td>
<td>This field displays the name associated with the AMIBIOS ROM file. The value in this field cannot be changed.</td>
</tr>
<tr>
<td>Processor</td>
<td>This field displays the number that is used to define the processor type. The value in this field cannot be changed.</td>
</tr>
<tr>
<td>Major Version</td>
<td>This field displays the main AMIBIOS revision number that is used with the AMIBIOS release. The value in this field cannot be changed.</td>
</tr>
<tr>
<td>ID String 1</td>
<td>This field displays the ID string that is associated with the AMIBIOS ROM file. The value in this field cannot be changed.</td>
</tr>
<tr>
<td>BIOS Size</td>
<td>This field displays the actual size of the AMIBIOS ROM file. The value in this field cannot be changed.</td>
</tr>
<tr>
<td>BIOS Tag</td>
<td>This field displays the eight-character tag that is associated with the AMIBIOS ROM file. The value in this field cannot be changed.</td>
</tr>
<tr>
<td>Minor Version</td>
<td>This field displays the minor AMIBIOS revision number that is used with the AMIBIOS release. This field is editable.</td>
</tr>
<tr>
<td>CPU Microcode Update Patches</td>
<td>This field displays processor patches that are contained in the AMIBIOS ROM file. The values in this field cannot be changed.</td>
</tr>
</tbody>
</table>
| 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:

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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 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 the icon on **Toolbar** to save the changes you have made.

*Note:* You can click 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:

![DMI Tables Tab Example](image)

**Fields**

The **DMI Tables** fields are explained in the following table:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formatted area</td>
<td>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.</td>
</tr>
</tbody>
</table>
### Buttons

The *DMI Tables* button is explained in the following table:

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;&lt;</td>
<td>This button allows you to go to the previous DMI table in the table list.</td>
</tr>
<tr>
<td>Next</td>
<td>This button allows you to go to the next DMI table in the table list.</td>
</tr>
<tr>
<td>Undo</td>
<td>This button allows you to restore the original setup settings.</td>
</tr>
</tbody>
</table>

**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 click on the *Undo* button to restore the original setup settings before saving any new changes.
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  AMIBIOS Utility CORE APIs DLL.
• UCOREVXD.VXD  Driver for Microsoft® Windows® 98/ME.
• UCORESYS.SYS  Driver for Microsoft® Windows® NT/2000/XP/PE.
• UCOREW64.SYS  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 window](image)

**Menu Bar**

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

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Save</td>
<td>Save current type information to specific path/filename.</td>
</tr>
<tr>
<td>Save All</td>
<td>Save all type information to specific path/filename.</td>
</tr>
<tr>
<td>Exit</td>
<td>Quit program.</td>
</tr>
</tbody>
</table>

View drop-down menu options
### View Menu Item List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Toolbar</td>
<td>Display or hide the Toolbar. The Toolbar is displayed under the Menu bar.</td>
</tr>
<tr>
<td>Status Bar</td>
<td>Display or hide the Status Bar. The Status Bar is displayed at the bottom of the DMI Editor window.</td>
</tr>
<tr>
<td>Refresh</td>
<td>Reload all SMBIOS information from actual BIOS ROM.</td>
</tr>
</tbody>
</table>

### Undo drop-down menu options

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>Reload all SMBIOS information from buffer.</td>
</tr>
<tr>
<td>Current Type</td>
<td>Reload current type information from buffer.</td>
</tr>
</tbody>
</table>

### Update drop-down menu options

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>Write all SMBIOS information to actual BIOS ROM.</td>
</tr>
<tr>
<td>Current Type</td>
<td>Write current type information to actual BIOS ROM.</td>
</tr>
</tbody>
</table>
About

The *About* is used to display AMIBCP copyrights information.

Toolbar

The *Toolbar* is located under the *Menu bar*.

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.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Icon 1]</td>
<td>Save current type information to specific path/filename.</td>
</tr>
<tr>
<td>![Icon 2]</td>
<td>Save all type information to specific path/filename.</td>
</tr>
<tr>
<td>![Icon 3]</td>
<td>Reload all SMBIOS information from actual BIOS ROM.</td>
</tr>
<tr>
<td>![Icon 4]</td>
<td>Reload all SMBIOS information from buffer.</td>
</tr>
<tr>
<td>![Icon 5]</td>
<td>Reload current type information from buffer.</td>
</tr>
<tr>
<td>![Icon 6]</td>
<td>Write all SMBIOS information to actual BIOS ROM.</td>
</tr>
<tr>
<td>![Icon 7]</td>
<td>Write current type information to actual BIOS ROM.</td>
</tr>
</tbody>
</table>
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.

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.

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:

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAP</td>
<td>The Caps Lock key is latched down.</td>
</tr>
<tr>
<td>Item</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>NUM</td>
<td>The Num Lock key is latched down.</td>
</tr>
<tr>
<td>SCRL</td>
<td>The Scroll Lock key is latched down.</td>
</tr>
</tbody>
</table>

**Functions**

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

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

**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 on **Toolbar**.
3. Input path/file name on dialog box and press **Save** to get information file Or press **Cancel** to ignore the function.

### Saving all SMBIOS information to file

1. Open *File* drop-down menu and select *Save All* item Or single click icon on *Toolbar*.
2. Input path/file name on dialog box and press _Save_ to get information file or press _Cancel_ to ignore the function. The dialog box is same as above.

**Updating SMBIOS type**

*This function is valid only on AMIBIOS system.*

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

2. Select an item that will be modified.

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

5. Press Yes to confirm the update instruction or No to ignore.

6. Now is updating data when the instruction is confirmed.
7. Update done. You will get the notice dialog box. Just press **OK** to finish the update operation.

**Updating all SMBIOS type**

*This function is valid only on AMIBIOS system.*

1. Repeat step 1-3 of **Update SMBIOS Type** to modify items.
2. Open **Update** drop-down menu and select **All** item or single click icon on **Toolbar**.
3. See step 5-7 of Update SMBIOS Type 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 icon on *Toolbar*.

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

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

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/n</td>
<td>Do not check ROM ID</td>
</tr>
</tbody>
</table>
| /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> | Save current system BIOS ROM into disk |
| /u<ROM File Name> | Display ROM file’s ROM ID string |