AMIBIOS8 ROM Utilities:

User Guide

Document Revision 1.29 – July 24, 2008

NDA REQUIRED

NDA Required
(NDA)

American Megatrends, Inc.

6145-F Northbelt Parkway

Norcross, GA 30071, USA

Limited Warranty

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

Limitations of Liability

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

Trademarks

American Megatrends acknowledges the following trademarks:

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

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

Disclaimer

This manual describes the operation of the AMIBIOS8 ROM Utilities. Although efforts have been made to insure the accuracy of the information contained here, American Megatrends expressly disclaims liability for any error in this information, and for damages, whether direct, indirect, special, exemplary, consequential or otherwise, that may result from such error, including but not limited to the loss of profits resulting from the use or misuse of the manual or information contained therein (even if American Megatrends has been advised of the possibility of such damages). Any questions or comments regarding this document or its contents should be addressed to American Megatrends at the address shown on the cover.

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

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

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

Revision History

Date	Ver.	Description
08/23/2004	1.00	- Initial Document. - This version contains AMIMMDOS.EXE, AMIOLDOS.EXE and ROMSETUP.EXE user guide.
10/04/2004	1.01	- Add new user guides for AMIDEDOS.EXE, AMIDEWIN.EXE, DMIEDIT and OEMLOGO.EXE.
11/03/2004	1.02	- Update for AMIMMDOS.EXE and AMIOLDOS.EXE.
11/16/2004	1.03	- Add new user guides for AFUDOS.EXE, AFUWIN.EXE
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.
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.
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
05/04/2005	1.07	- Add new comments for AMIMMWIN.EXE, AMIOLWIN.EXE.
06/04/2005	1.08	- Add user interface manual for AFUWIN.EXE.
09/03/2005	1.09	- Add new user guide for AFULNX.EXE, AMIPWD.EXE, AMIPWDW.EXE.
02/16/2006	1.10	- Add new user guide for TEXTBCPD|W.EXE.
06/08/2006	1.11	- Update comment for AFULNX.
07/13/2006	1.12	- Add new comments for AFUDOS. - Add new user guide for AFUBSD.
07/21/2006	1.13	- TXTBCPD can be used only on DOS system.
07/26/2006	1.14	- Add new comment for AFUDOS.
11/03/2006	1.15	- Add new commands to AMIDEDOS and AMIDEWIN.
01/04/2007	1.16	- Add Vista support comments. - Add AMIDEDOS comments for new supported function.

		- Add AMIDEWIN comments.
01/19/2007	1.17	- Add AFU Utilities supports ROM Hole Update and Rom Hole protection functions. - Add AFU Utilities supports /D command for backward comatible. - Add AFU ROM ID checking become case non-senstive cehcking rule.
03/29/2007	1.18	- Add AFU supports Clear Event Log function.
04/20/2007	1.19	- Add DMI editor for Linux command mode usage.
07/6/2007	1.20	- Add AFU supports Non Critical Block update & output function. - Update AMI DMI Editor require modules. - Add /GENDRV cmd for AMIDELNX to generate drver configuration and source code files.
09/6/2007	1.21	- Add AFU supports /DeDftCfg to disable all default settings. - Add AFU supports ~command name to disable certain command’s default.
09/19/2007	1.22	- Add AFU /DeDftCfg ~Command Name examples.
10/09/2007	1.23	- Change ~Command Name to /-Command Name due to ~shutdown is a build in command in Linux
2007-11-07	1.24	- Minor formatting for documentation standard
2007-12-07	1.25	- Add DMIEditor Tool support /BT command to update SMBIOS Type 2 Asset Tag. - Notice: Please update Core8 BIOS SMBIOS Module version to 8.00.08_SMB-3.1.02_CORE_RC24 to support function properly.
2008-02-11	1.26	- Added information so utility names match up to names used in marketing
2008-02-26	1.27	- Added information for DMIEditor needs to include DMI16.EXE when BIOS supports PNP function only. - Added AMIUCP User Guide (New Tool).
2008-05-21	1.28	- Correction of DMIEditor document. - Add AFU Utilities Error Code List table.
2008-07-24	1.29	- Add comments for following Utilities about supporting Microsoft Windows PE 2.0 x64 1. AMIDEWINx64.EXE 2. AMISCEWx64.EXE 3. AFUWINx64.EXE

Table of Contents

Part 1: Command Mode

Chapter 1 AMIMMDOS/WIN (MMTool)

Overview

AMIMMDOS/WIN (MMTool) 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
- Microsoft® Windows® Vista 32/64

AMIMMWIN is supported by the following operating systems:

- Microsoft® Windows® 98

- Microsoft® Windows® ME
- Microsoft® Windows® NT 4.0
- Microsoft® Windows® 2000
- Microsoft® Windows® XP/XP64
- Microsoft® Windows® PE
- Microsoft® Windows® Vista 32/64

BIOS Requirements

The loaded BIOS ROM file should have the followings:

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

Getting Started

Installation

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

Usage

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

BIOS ROM File Name

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

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.

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

Parameters List	
Name	**Description**
New MicroCode File Name	This parameter is used to specify path/filename of the new CPU MicroCode Patch file with extension.
L-VID	4-digits hexadecimal linked-vendor ID.
L-DID	4-digits hexadecimal linked-device ID.
VID	4-digits hexadecimal vendor ID.
DID	4-digits hexadecimal device ID.
8-Bits Value	This parameter MUST be 2-digits hexadecimal value.
M#1	This parameter MUST be 4-digits hexadecimal value. For generic module, it is the module runtime Segment. For module ID – 20h, it is the PCI ROM device ID. For module ID – 21h, it is a Flags.
M#2	This parameter MUST be 4-digits hexadecimal value. For generic module, it is the module runtime Offset. For module ID – 20h, it is the PCI ROM vendor ID. For module ID – 21h, it is the Country Code.
NCB Name	= EBB, The module placed in Extended Boot Block. = NCB00, The module placed in first NCB region. = NCB01, The module placed in second NCB region. . . = NCBxx, The module placed in [xx]th NCB region.
Start Address	8-digits hexadecimal starting address of the module in the ROM Image.

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

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

Examples

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

- **Display CPU MicroCode Patch module information**

  AMIMMDOS <BIOS ROM File Name> /CP

- **Save CPU MicroCode Patch module information to file**

  AMIMMDOS <BIOS ROM File Name> /CP <Output File Name>

- **Delete a CPU MicroCode Patch data from ROM Image**

  AMIMMDOS <BIOS ROM File Name> /CPD <CMP ID>

- **Extract a CPU MicroCode Patch data to file**

  AMIMMDOS <BIOS ROM File Name> /CPE <CMP ID> <Output File Name >

- **Insert a CPU MicroCode Patch data to ROM Image**

  AMIMMDOS <BIOS ROM File Name> /CPI <New MicroCode File Name>

- **Deleting an existing module**

  AMIMMDOS <BIOS ROM File Name> /D <Module ID> [/M <M#1> <M#2>]

- **Extracting a module**

  AMIMMDOS <BIOS ROM File Name> /E <Module ID> <Output File Name> [/M <M#1> <M#2>] [/U]

- **Inserting a new module**

  AMIMMDOS <BIOS ROM File Name> /I <Module ID> <Module File Name> [/A] [/C] [/N <NCB Name>] [/M <M#1> <M#2>] [/S <Start Address>] [/U]

- **Inserting a linked-ID module**

  AMIMMDOS <BIOS ROM File Name> /LM <L-VID> <L-DID> <VID> <DID>

- **Display BIOS ROM Information**

  AMIMMDOS <BIOS ROM File Name> /INFO

- **Save BIOS ROM Information to file**

  AMIMMDOS <BIOS ROM File Name> /INFO <Output File Name>

- **Replacing an existing module**

  AMIMMDOS <BIOS ROM File Name> /R <Module ID> <Module File Name> [/A] [/C] [/N <NCB Name>] [/M <M#1> <M#2>] [/S <Start Address>] [/U]

- **Insert a file to ROM Hole**

  AMIMMDOS <BIOS ROM File Name> /RHF <Hole Number> <Hole File Name>

- **Read SLP string from BIOS ROM File**

  AMIMMDOS <BIOS ROM File Name> /SLP

- **Write SLP string to BIOS ROM File**

  AMIMMDOS <BIOS ROM File Name> /SLP <”SLP string”>

- **Read Sign-On Message from BIOS ROM File**

  AMIMMDOS <BIOS ROM File Name> /SM

- **Write Sign-On Message to BIOS ROM File**

  AMIMMDOS <BIOS ROM File Name> /SM <”Sign-On Message”>

Chapter 2 AMIOLDOS/WIN (ChangeLogo)

Overview

AMIOLDOS/WIN (ChangeLogo) 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
- Microsoft® Vista 32/64

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
- Microsoft® Windows® Vista 32/64

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)
- 256-Color GIF formate, even width, 640*480 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 [/A] [/S]

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:

/A	Change Animated GIF logo
/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 encolsed by < > is a mandatory field.**
- **Any parameter enclosed by [ ] is an optional field.**
- Change large OEM Logo and check Logo format are default operation.
- Small Logo image will be converted to GRFX format automatically.
- [**/D**] MUST be used alone except [**/S**].
- [**/D**] MUST be used alone except [**/A**].
- [**/N**] and [**/FN**] option cannot co-exist with [**/S**].
- [**/XY**] is available only for GRFX logo format.
- [**/N**] and [**/FN**] option does not affect with [**/A**]

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

Examples

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

- **Replacement of large OEM Logo**

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

- **Replacement of small OEM Logo**

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

- **Replacement of Animated GIF Logo**

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

- **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

- **Deleting Animated GIF Logo**

  AMIOLDOS <BIOS ROM File Name> /D /A

- **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 Animated GIF Logo**

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

- **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.
- *Token:* ***Build_For_ROMSETUP = ON***.
- Token Build_For_ROMSETUP token to 1.

Getting Started

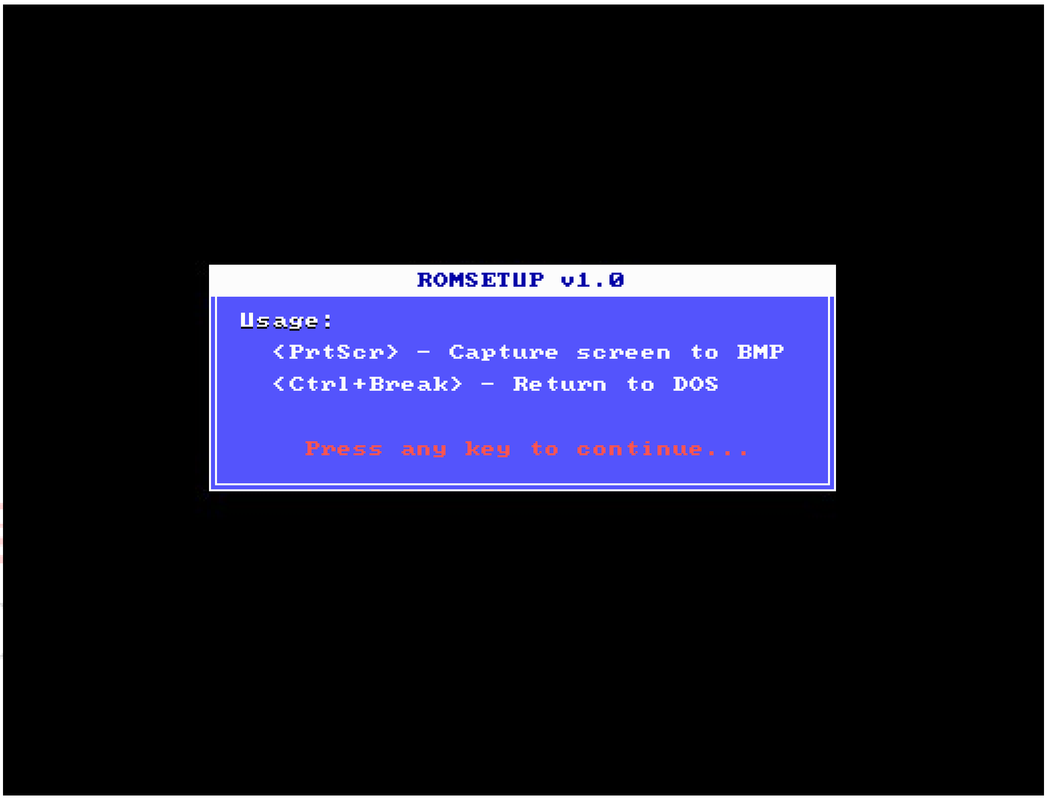
Installation

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:

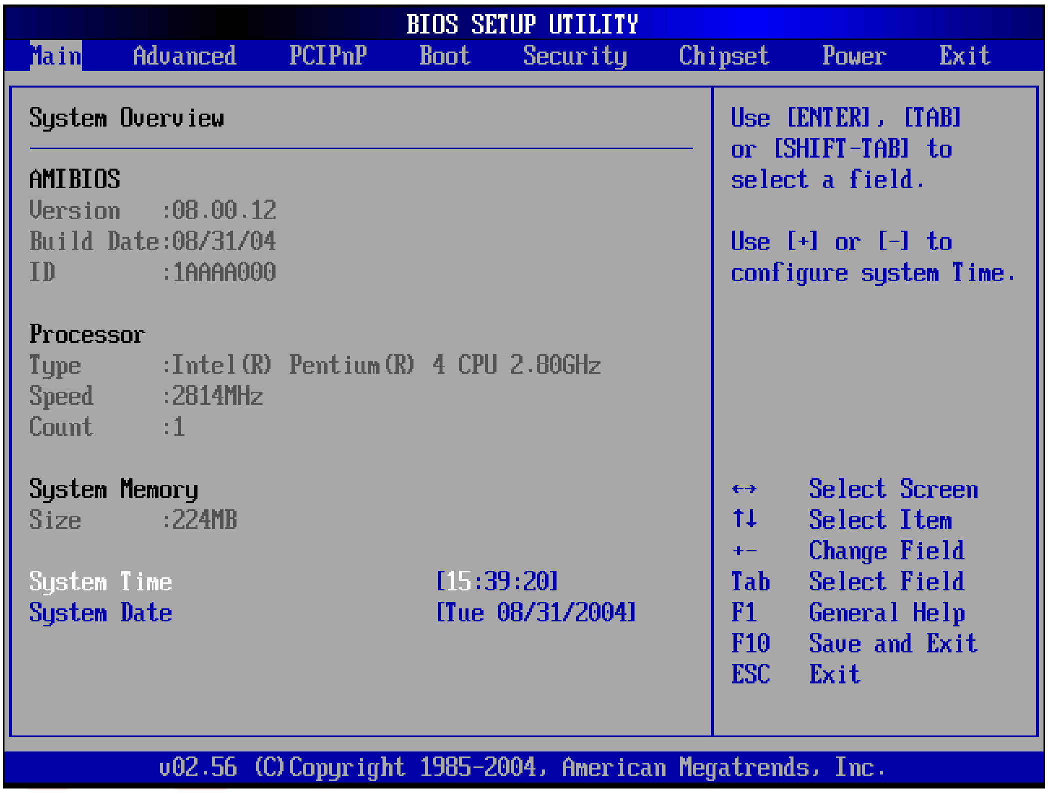
1. Boot to DOS and make sure that EMM386 does not install.
2. Switch to the directory where ROMSETUP is existing.
3. Type ROMSETUP behind command prompt symbol(\>) and press ENTER to run ROMSETUP, afterward, the usage screen will be displayed:



```
                ROMSETUP v1.0
 Usage:
   <PrtScr> - Capture screen to BMP
   <Ctrl+Break> - Return to DOS

     Press any key to continue...
```

Print Screen	Capture screen and save it as BMP file format to disk.
Ctrl+Break	Break program and back to DOS.

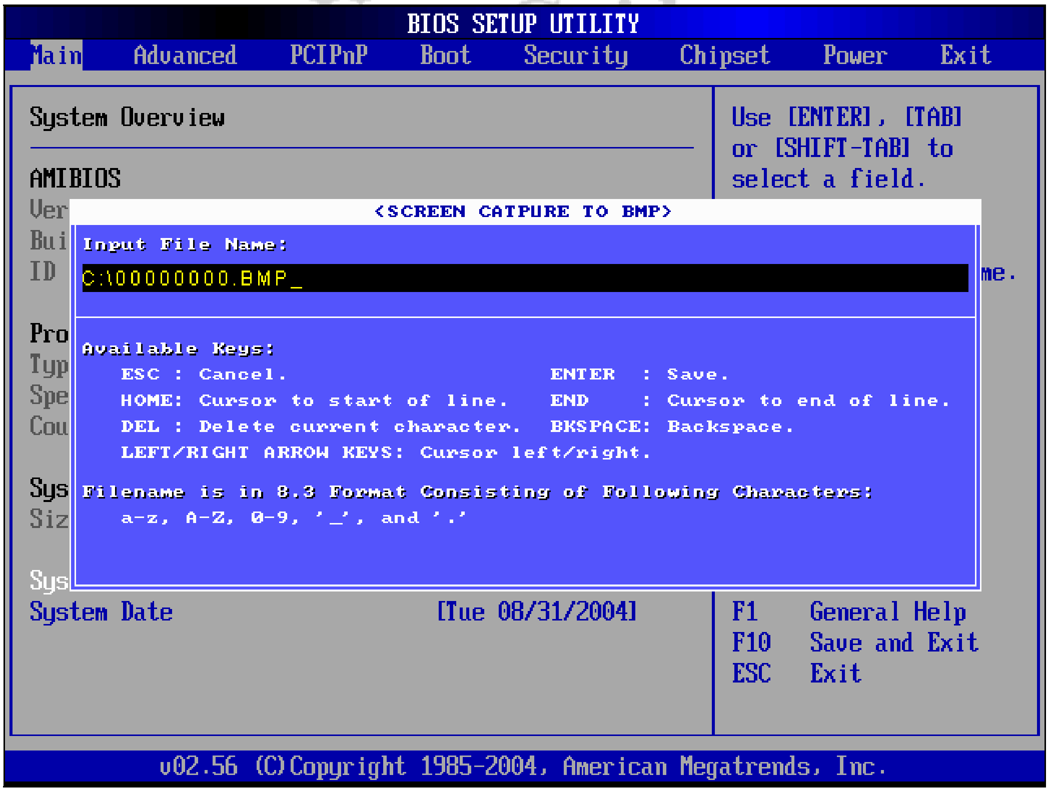
4. Press any key to continue. SETUP screen will be displayed:



```
                              BIOS SETUP UTILITY
 Main   Advanced   PCIPnP   Boot   Security   Chipset   Power   Exit

 System Overview                                    Use [ENTER], [TAB]
                                                    or [SHIFT-TAB] to
 AMIBIOS                                            select a field.
 Version    :08.00.12
 Build Date:08/31/04                                Use [+] or [-] to
 ID         :1AAAA000                               configure system Time.

 Processor
 Type       :Intel(R) Pentium(R) 4 CPU 2.80GHz
 Speed      :2814MHz
 Count      :1

 System Memory                                      ↔    Select Screen
 Size       :224MB                                  ↑↓   Select Item
                                                    +-   Change Field
 System Time                   [15:39:20]           Tab  Select Field
 System Date                   [Tue 08/31/2004]     F1   General Help
                                                    F10  Save and Exit
                                                    ESC  Exit

         v02.56 (C)Copyright 1985-2004, American Megatrends, Inc.
```

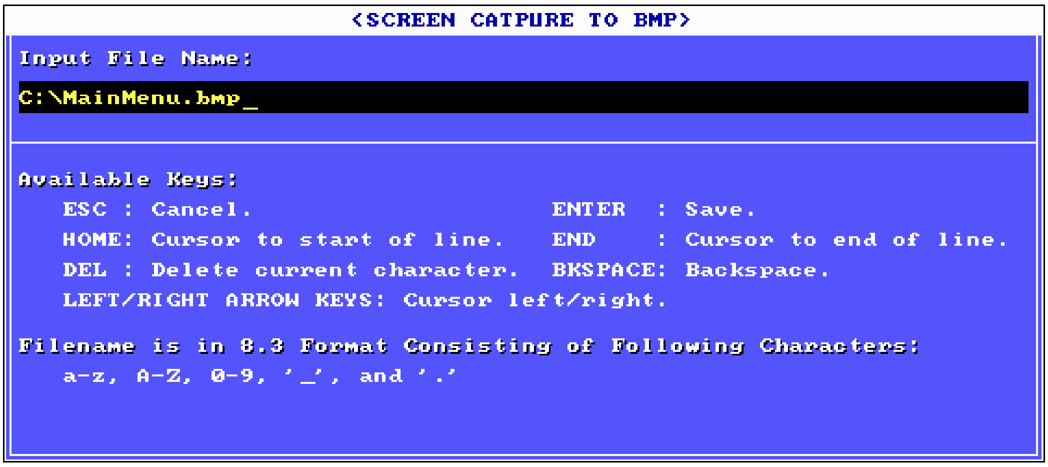
Note: The SETUP screen may be different from above map.
It is depending on OEM's requirement.

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



```
                              BIOS SETUP UTILITY
 Main   Advanced   PCIPnP   Boot   Security   Chipset   Power   Exit

 System Overview                                    Use [ENTER], [TAB]
                                                    or [SHIFT-TAB] to
 AMIBIOS                                            select a field.
 Ver            <SCREEN CAPTURE TO BMP>
 Bui  Input File Name:
 ID   C:\00000000.BMP_                                                  me.

 Pro  Available Keys:
 Typ     ESC : Cancel.                    ENTER   : Save.
 Spe     HOME: Cursor to start of line.   END     : Cursor to end of line.
 Cou     DEL : Delete current character.  BKSPACE: Backspace.
         LEFT/RIGHT ARROW KEYS: Cursor left/right.
 Sys  Filename is in 8.3 Format Consisting of Following Characters:
 Siz     a-z, A-Z, 0-9, '_', and '.'

 Sys
 System Date                   [Tue 08/31/2004]     F1   General Help
                                                    F10  Save and Exit
                                                    ESC  Exit

         v02.56 (C)Copyright 1985-2004, American Megatrends, Inc.
```

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



```
                        <SCREEN CATPURE TO BMP>
Input File Name:
C:\MainMenu.bmp_

Available Keys:
   ESC : Cancel.                     ENTER  : Save.
   HOME: Cursor to start of line.    END    : Cursor to end of line.
   DEL : Delete current character.   BKSPACE: Backspace.
   LEFT/RIGHT ARROW KEYS: Cursor left/right.

Filename is in 8.3 Format Consisting of Following Characters:
   a-z, A-Z, 0-9, '_', and '.'
```

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 (DMIEdit)

Overview

AMIDEDOS (DMIEdit) 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:

- BIOS Information (Type 0)
- System (Type 1)
- Base Board (Type 2)
- Chassis (Type 3)
- OEM String (Type 11)
- System Configuration Options (Type 12)

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-1.00.10” label or later.
- ***SMBIOS*** *eModule* with “8.00.08_SMB-3.1.02_CORE_RC24” 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

AMDEDOS <Command 1>

Or

AMDEDOS [Option 1] [Option 2]…..

Configuration File Name

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

```
[BIOS]
Version = 080012
Date = 12/28/2006

[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
AssetTag = To be filled by O.E.M.

[Chassis]
Manufacturer = AMI
Version = 8.00
SerialNum = 0123456789
```

```
TagNum = 0123456789
ChassisType = 03
ChassisOEM = 0123456789

[OEMString]
String = AMI
String = WWW.AMI.COM
String = BIOS vendor

[Configuration]
String = To Be Filled By O.E.M.
```

Commands

/ALL [Output File Name]	Output information to screen Or file.
/DMS [Output File Name]	Create configuration file. Default file name is “CONFIG.DMS”.
/DUMP # [#] [#]	Read Type # data.
/DUMPALL [FileName]	Output all SMBIOS data to screen Or file.

Options

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 0. System (Type 0)

/IV [“String”]	Read/Write BIOS Version.
/ID [“String”]	Read/Write BIOS release date.

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.
/SU AUTO	Generate system UUID automatically and update Type 1
/SK [“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.
- **/BT [“String”]** Read/Write Asset Tag

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. OEM String (Type 12)

- **/SCO [<Number> <”String”>]** Read/Write #th OEM string.

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

Rules
▪ **Any parameter encolsed by < > is a mandatory field.**
▪ **Any parameter enclosed by [ ] is an optional field.**
▪ For command part 1-4, if parameter present, the WRITE function is going to update else READ function will be enabled.
▪ 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.

Rules
▪ 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 AMIBIOS8
  ```

- **Get chassis serial number**

  ```
  AMIDEDOS /CS
  ```

- **Get system UUID, baseboard version and chassis type information at once**

  ```
  AMIDEDOS /SU /BV /CT
  ```

- **Get OEM String**

  ```
  AMIDEDOS /OS
  ```

- **Update system manufacturer and get system UUID at once**

  ```
  AMIDEDOS /SM AMI /SU
  ```

- **Read Type 1, 2 and 3**

  ```
  AMIDEDOS /DUMP 1 2 3
  ```

- **Read all SMBIOS data and display the information on screen**

  ```
  AMIDEDOS /DUMPALL
  ```

- **Read all SMBIOS data and save the information to file**

  ```
  AMIDEDOS /DUMPALL SMBIOS.TXT
  ```

Chapter 5 AMIDEWIN (DMIEdit)

Overview

AMIDEWIN (DMIEdit) is a SMBIOS data manipulation 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:

- BIOS Information (Type 0)
- System (Type 1)
- Base Board (Type 2)
- Chassis (Type 3)
- OEM String (Type 11)
- System Configuration Options (Type 12)

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
- Microsoft® Windows® Vista 32/64
- Microsoft® Windows® PE 2.0 x64 (AMIDEWINx64.EXE)

BIOS Requirements

System BIOS should have the followings:

- AMIBIOS CORE version 8.xx.xx.
- ***SMIFlash*** *eModule* with “8.00.00_SMIFlash-1.00.10” 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.
• ***UCORE.DLL***	Driver for PnP function.

Getting Started

Installation

Copies ***AMIDEWIN.EXE***, ***AMIDEWINx64.EXE*** **(for Microsoft Windows PE 2.0 x64)** , ***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 (AMI Firmware Update)

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>

Or

AFUDOS /MAI

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
▪ **/Ln**	Refer to Options: /Ln
▪ **/M<MAC Address>**	Refer to Options: /M
▪ **/MAI**	Display current system and ROM file's MA information.
▪ **/HOLE**	Update specific ROM Hole according to given name.
▪ **/HOLEOUT**	Save specific Rom Hole Data according to given name.
▪ **/D**	Verification test of given ROM File without flashing BIOS.
▪ **/EC**	Flash EC firmware BIOS (Refer to OFBD Spec) Path: $BIOS/Corebin/800/ROMUtils/On Flash Block Description Specification.PDF. Sample Code Module Path: $BIOS/Examples/On Flash Block Description
▪ **/NCB**	Flash NCB Area (Refer to OFBD Spec)
▪ **/NCBOUT**	Output NCB Data according to given name.
▪ **/C**	Destroy CMOS checksum.

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
/ECUF	Update EC BIOS when newer version is detected.
/ShutDown	Shutdown system after programming.
/clnevnlog	Clean Event Log.
/DeDftCfg	Delete all default settings from BIOS.
/-Command Name	Delete certain command’s default setting. [OEM Uses Only.]

Rules
▪ **Any parameter encolsed 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.

Rules
▪ If [**/E**] present alone, there is only the Embedded Controller block to be updated.
▪ If [**/E**] and **[/ECUF]** co-exist, **[/ECUF]** will be no function.
▪ If [**/Kn**] present alone, there is only non-critical block to be updated.
▪ When [**/Ln**] is co-exist with [**/C**], [**/C**] will be no function.
▪ [**/M**] can be used as a command for backward compatible.

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

Examples

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

- **Save current BIOS ROM to file**

  ```
  AFUDOS <BIOS ROM File Name> /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 Embedded Controller Block if newer version is detected**

  ```
  AFUDOS <BIOS ROM File Name> /ECUF
  ```

- **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

- **Update dedicate ROM Hole Area**

  AFUDOS <ROM Hole File Name> /Hole:Name

- **Update dedicate NCB Area**

  AFUDOS <NCB File Name> /NCB:Name

- **Output dedicate ROM Hole File**

  AFUDOS <Output ROM Hole File Name> /HOLEOUTt:Name

- **Output dedicate NCB File**

  AFUDOS <Output NCB File Name> /NCBOUT:Name

- **Cancel Embedded AFU default commands**

  - Below sample cancels B & P commands if BIOS has embedded B & P commands in OFBD.

  AFUDOS <BIOS ROM File Name> /-B /-P

  Notice: if /p & /b are set as default command only and /-B /-P commands are issued then P command will still be issued because if none of command is issued then /p will still issue as AFU default.

- **Cancel ALL Embedded AFU default commands**

  AFUDOS <BIOS ROM File Name> /DeDftCfg

Error Code List

Error Number	Description
00h	No error.
01h	Unknown command.
02h	Can't open ROM ID file.
03h	ROM ID file is not a ROM file.
04h	Invalid MAC address.
05h	Invalid retry count.
06h	System doesn't support MAC programming.
07h	This program can not run under this operating system.

08h	Flash part is not supported.
09h	Problem extracting module from ROM file.
0Ah	Can not analyze ROM file. ROM file may be corrupted.
0Bh	NCB error
0Ch	Invalid option
0Dh	BIOS does not support AFU.
0Eh	ROM file size incorrect
0Fh	File ROM ID incorrect
10h	Bootblock error
11h	Loading driver
12h	Unloading driver
13h	Invalid NCB
14h	Closing memory manager
15h	Mapping BIOS data buffer error
16h	Problem allocating memory
17h	Problem freeing memory
18h	Problem allocating BIOS buffer
19h	Problem freeing BIOS buffer

Chapter 7 AFULNX/AFUBSD

Overview

AFULNX/AFUBSD (AMI Firmware Update) is an updating system BIOS utility with command line interface. It has same parameters and behavior as AFUDOS.

For the convenience, AFULNX has a big change to related drivers. All necessary drivers are generated and loaded automatically since version 4.10. User can just launch the program and wait for updating job finish.

By the way, do not forget that target board MUST be ***AMIBIOS*** system while using this utility.

Features

This utility offers the following features:

- No need to build driver by yourself for different distributions of Linux(v4.10 or above)
- 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

AFULNX Utility is supported in following operating system:

- Linux CORE v2.4/2.6

AFUBSD Utility is supported in following operating system:

- FreeBSD operating system

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 for AFULNX only

You can forget this section if you are using AFULNX v4.10 or above.

Following driver for different version of Linux system are required by this utility:

• ***UCORELNX.O32***	Driver for 32-Bit Linux.
• ***UCORELNX.O64***	Driver for 64-Bit Linux.

Getting Started for AFULNX v4.06 and below

Preparing suitable driver file

1. Log in Linux as root.
2. The compiler suite(GCC) must be installed. If these packages are not installed, the driver CANNOT be built.
3. Kernel sources must be installed, *CONFIGURED*, and then compiled.
   Following are steps to do this:
   3.1 Find Running Kernel's Configuration File:
   - To configure the sources, simply change to the kernel source directory (typically /usr/src/linux). If it doesn't exist, you need to install kernel source. Typically, the reference configuration for the kernel can be found in the /boot directory with filename '.config', 'kernel.config', or 'vmlinux-2.4.18-3.config'. Type 'uname -a' and use the configuration filename that best matches the output from 'uname -a'.
   - On some distributions Red Hat for instance, there is a config directory under /usr/src/linux.
   - Copy this configuration file into the root of the linux kernel source tree(usually it is /usr/src/linux). This file must be renamed to ".config"(dot config).

   3.2 Make the Linux Kernel:
   - Under linux kernel source root(/usr/src/linux), type the command 'make' if your Linux kernel version is 2.6 or above; otherwise use 'make oldconfig dep' instead.
   - This will generate files that are required to build the driver.
   - The process of compiling the Linux kernel will take a while to accomplish.

   3.3 Copy Your AMI Flash Driver:
   - The AMI flash driver is distributed in a compressed TAR archive. After saving this file to your Linux system, it must be extracted so that the driver

may be built.

- First, create a directory for AFU with the command 'mkdir afu'. Change your working path into it by 'cd afu'.
- To extract the archive, you'll need to run the shell command(as root):

```
tar xvzf afulnx2.tgz
```

- The name of the archive may be different, but the overall syntax is the same.

3.4 Determining Which Driver Makefile to Use:

- Due to a change made into Linux kernel 2.6. The Makefile for 2.6 is different from older kernel.
- Type 'uname -r' to see the version.
- If your kernel version is 2.6 or greater, copy the file 'Makefile.v26' to 'Makefile'.
- If your kernel version is below 2.6, copy the file 'Makefile.v24' to 'Makefile'.
- The command to do the copy is 'cp Makefile.v2x Makefile'.

3.5 Make Your AMI Flash Driver(UCORELNX.O32/UCORELNX.O64):

- For most distribution, the command to build the driver is:

```
make
```

- If your linux's kernel source tree is under /usr/src/linux-2.4 instead of the default path '/usr/src/linux', add a KERNEL flag:

```
make KERNEL=/usr/src/linux-2.4
```

- For Red Hat 8.0 distribution with kernel located under /usr/src/linux-2.4, use this command:

```
make REDHAT9=1 KERNEL=/usr/src/linux-2.4
```

- If KERNEL is omitted, the default is /usr/src/linux. This should work for MOST distributions.
- To clean up object file, use the command:

```
make clean
          Or
make REDHAT9=1 KERNEL=/usr/src/linux-2.4 clean
```

3.6 Check Your Build:

- Check the version of running Linux kernel with 'uname -r'.
- Check the version of UCORELNX.O32/UCORELNX.O64 with 'modinfo ucorelnx.032' and ‘modinfo ucorelnx.o64’.
- If they mismatch, you will need to select the correct configuration file(.config), rebuild your kernel, and then rebuild your driver as described in (3.1), (3.2), and (3.5).

Installation

- Copies ***AFULNX2_32***, ***AFULNX2_64, UCORELNX.O32***, ***UCORELNX.O64*** to any storage location accessible by the host system
- To determine which version of Linux system, type 'uname -m'. under shell screen. If it says 'x86_64', then ***AFULNX2_64*** should be used; otherwise, use the ***AFULNX2_32***.
- Run the suitable file and remember to keep the relative driver in same directory.

Troubleshooting

Q1: I get following error message when loading driver:
"insmod: error inserting 'UCORELNX.O32'|'UCORELNX.O64': -1 Invalid module format".

A1: Most likely this is cause by wrong configuration file and your kernel refuses to accept your driver because version strings(more precisely, version magic) do not match.

To check the version of running Linux kernel, type "uname -r".
To check the version of UCORELNX.O32/UCORELNX.O64, type "modinfo UCORELNX.O32" or “modinfo UCORELNX.O64”

If they mismatch, you will need to select the correct configuration file(.config), rebuild your kernel, and then rebuild your driver as described in “preparing suitable driver file” section.

Q2: When I run ./afulnx2, it says "Unable to load driver".

A2: Some Linux distributions do not display driver debug messages on screen by default. Type "dmesg" to see those debug messages. This is very likely the same problem as Q1.

Q3: When I run ./afulnx2, it simply freezes.

A3: This is caused by a Linux feature called "NMI Watchdog" which is used to debug Linux kernel. This feature must be disabled to run AFULNX2.
Please do "cat /proc/interrupts" twice and check if NMI is counting.
If it is, then boot Linux with a kernel parameter "nmi_watchdog=N" where N is either 0, 1 or 2. Find out which configuration can halt NMI from counting by "cat /proc/interrupts" This is the configuration we should use to run AFULNX2.

Usage & Example for command line mode

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

Getting Started for AFULNX v4.10 or above

Installation

- Copies **AFULNX2.TGZ** to any storage location accessible by the host system.
- Extracts the contents to same directory. You will get two folders, one is **AFULNX2_24** and another is **AFULNX2_26**.
- Type 'uname -r' to identify kernel version. If it says 2.4.xx…, you should enter to **AFULNX2_24** folder, otherwise, enter to **AFULNX2_26** folder.
- To determine which version of Linux system, type 'uname -m'. under shell screen. If it says 'x86_64', then ***AFULNX_64*** should be used; otherwise, use the ***AFULNX_32***.
- Run the suitable file. AFULNX will generate necessary drivers and load it automatically. If AFULNX cannot work well, please refer to "Generating driver file manually" or "Troubleshooting" section to get help.

Generating driver file manually

1. Log in Linux as root.
2. The compiler suite(gcc) must be installed. If these packages are not installed, the driver CANNOT be built.
3. For most distributions, AFULNX2 will generate AMI Flash Driver file automatically without notification. Certainly, the driver file may NOT be generated in some specific case and the loading driver failure message will be displayed. If you get this error, first, you can read 'Q1' and 'Q2' in 'TROUBLESHOOTING section' to shut out the kernel issues, and second, you can see Point.4 below to create driver file by yourself and launch AFULNX2 again.
4. Kernel sources must be installed, *CONFIGURED*, and then compiled. Following are steps to do this:

   4.1 Find Running Kernel's Configuration File

   - To configure the sources, simply change the kernel source directory (typically /lib/module/$(uname -r)/build). If it doesn't exist, you need to install kernel source. Typically, the reference configuration for the kernel can be found in the /boot directory with filename '.config', 'kernel.config', or 'vmlinux-2.4.18-3.config'. Type 'uname -a' and use the configuration filename that best matches the output from 'uname –a'.
   - On some distributions Red Hat for instance, there is a config directory under/lib/modules/$(uname -r)/build.

- Copy this configuration file into the root of the linux kernel source tree(usually it is /lib/modules/$(uname -r)/build). This file must be renamed to ".config"(dot config).

4.2 Make Your AMI Flash Driver(amifldrv_mod.drv)

- For most distribution, the command to build the driver is:

  AFULNX_32 /MAKEDRV

  Or

  AFULNX_64 /MAKEDRV

- If your linux's kernel source tree is under /lib/modules/$(uname -r)/build instead of the default path '/lib/modules/$(uname -r)/build', add a KERNEL flag:

  AFULNX_32 /MAKEDRV KERNEL=/lib/modules/$(uname -r)/build

  Or

  AFULNX_64 /MAKEDRV KERNEL=/lib/modules/$(uname -r)/build

- If KERNEL is omitted, the default is /lib/modules/$(uname -r)/build.
- This should work for MOST distributions.

4.3 Check your build

- Check the version of running Linux kernel with 'uname -r'.
- Check the version of amifldrv_mod.drv with 'modinfo amifdrv_mod.drv'.
- If they mismatch, you will need to select the correct configuration file(.config), rebuild your kernel, and then rebuild your driver as described in (4.1), (4.2), and (4.3).
- The amifdrv_mod.drv must be in same directory with afulnx_32(afulnx_64). If they match, continue on to the 'AFULNX2' section to run afulnx2.

Troubleshooting

Q1: I get following error message when loading driver:
"insmod: error inserting 'amifldrv_mod.o': -1 Invalid module format".

A1: Most likely this is cause by wrong configuration file and your kernel refuses to accept your driver because version strings(more precisely, version magic) do not match.

To check the version of running Linux kernel, type "uname -r".
To check the version of amifldrv_mod.drv, type "modinfo amifdrv_mod.drv"

If they mismatch, you will need to select the correct configuration file(.config), rebuild your kernel, and then rebuild your driver as described in “Generating driver file manually” section.

Q2: When I run ./afulnx_32(./afulnx_64), it says "Unable to load driver".

A2: Some Linux distributions do not display driver debug messages on screen by default. Type "dmesg" to see those debug messages. This is very likely the same problem as Q1.

Q3: When I run ./afulnx_32(./afulnx_64), it simply freezes.

A3: This is caused by a Linux feature called "NMI Watchdog" which is used to debug Linux kernel. This feature must be disabled to run AFULNX2.
Please do "cat /proc/interrupts" twice and check if NMI is counting.
If it is, then boot Linux with a kernel parameter "nmi_watchdog=N" where N is either 0, 1 or 2. Find out which configuration can halt NMI from counting by "cat /proc/interrupts" This is the configuration we should use to run AFULNX2.

Error Code List

See AFUDOS Error Code List table for detail.

Getting Started for AFUBSD v2.00 or above

Installation

- Copies **AFUBSD.TGZ** to any storage location accessible by the host system.
- Extracts the contents to same directory. You will get a folder named **AFUBSD**.
- Run **AFUBSD** in command prompt.

Usage & Example for command line mode

For AFULNX v4.10, we have added a new command:

AFULNX /MAKEDRV <Kernel Path>

This command can help user to build driver manually. Please see “Generating driver file manually” section to know detail. In addition to this command, other behaviors are same as AFUDOS.EXE. So you can see Usage of AFUDOS and Example of AFUDOS to learn more information.

Error Code List

See AFUDOS Error Code List table for detail.

Chapter 8 AFUWIN (AMI Firmware Update)

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
- Microsoft® Windows® Vista 32/64
- Microsoft® Windows® PE 2.0 x64 (AFUWINx64.EXE)

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.
- ***UCOREW64.SYS*** Driver for Microsoft® Windows® XP64.

Getting Started

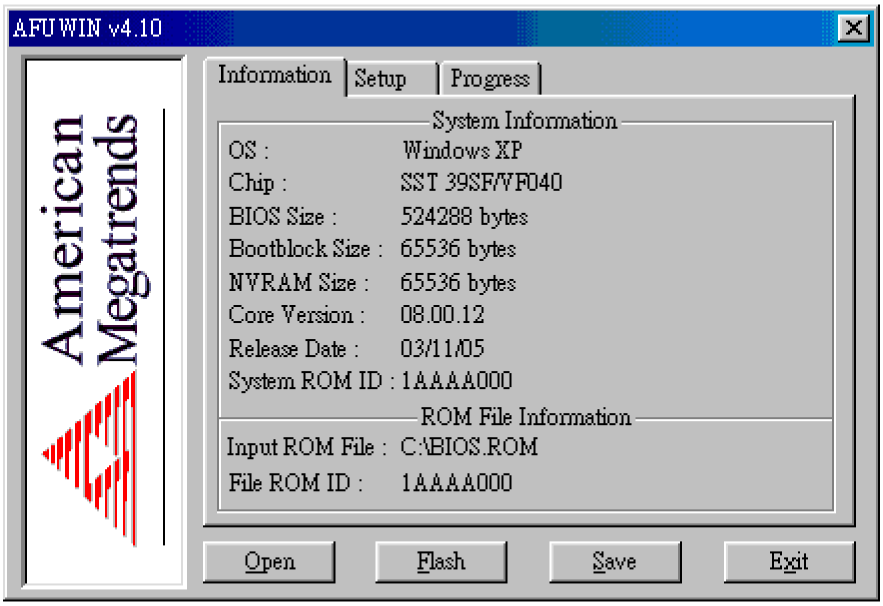
Installation

Copies ***AFUWIN.EXE***, ***AFUWINx64.EXE (for Microsoft Windows PE 2.0 x64)***, ***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



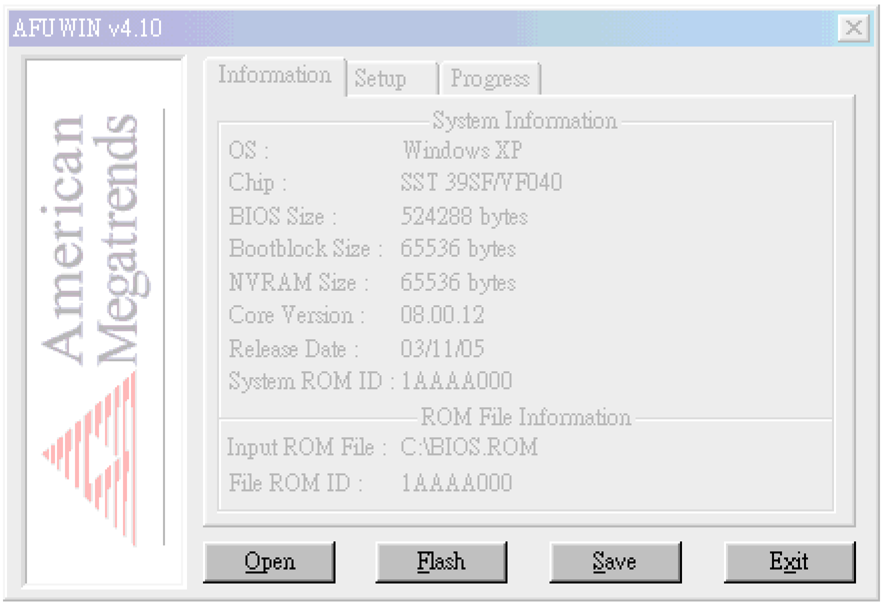
System Information

OS : Windows XP
Chip : SST 39SF/VF040
BIOS Size : 524288 bytes
Bootblock Size : 65536 bytes
NVRAM Size : 65536 bytes
Core Version : 08.00.12
Release Date : 03/11/05
System ROM ID : 1AAAA000

ROM File Information

Input ROM File : C:\BIOS.ROM
File ROM ID : 1AAAA000

Buttons



System Information

OS : Windows XP
Chip : SST 39SF/VF040
BIOS Size : 524288 bytes
Bootblock Size : 65536 bytes
NVRAM Size : 65536 bytes
Core Version : 08.00.12
Release Date : 03/11/05
System ROM ID : 1AAAA000

ROM File Information

Input ROM File : C:\BIOS.ROM
File ROM ID : 1AAAA000

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

Flash   Click this button to starting update BIOS.

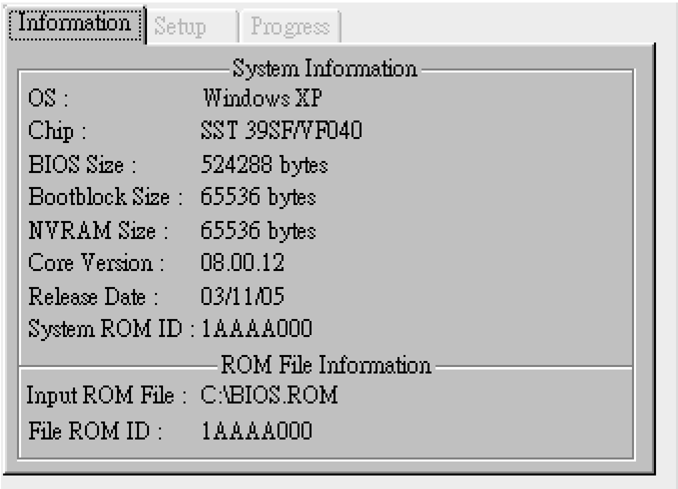
Save   Click this button to save BIOS ROM image to disk drive.

Exit   Click this button to exit this program.

Function Frame

Information Tab

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



System Information

OS : Windows XP
Chip : SST 39SF/VF040
BIOS Size : 524288 bytes
Bootblock Size : 65536 bytes
NVRAM Size : 65536 bytes
Core Version : 08.00.12
Release Date : 03/11/05
System ROM ID : 1AAAA000

ROM File Information

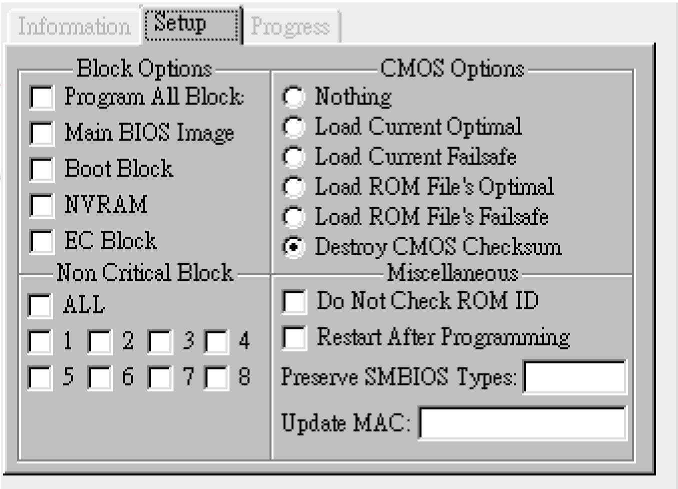
Input ROM File : C:\BIOS.ROM
File ROM ID : 1AAAA000

Field

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

Setup Tab

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



Block Options
□ Program All Block
□ Main BIOS Image
□ Boot Block
□ NVRAM
□ EC Block

Non Critical Block
□ ALL
□ 1 □ 2 □ 3 □ 4
□ 5 □ 6 □ 7 □ 8

CMOS Options
○ Nothing
○ Load Current Optimal
○ Load Current Failsafe
○ Load ROM File's Optimal
○ Load ROM File's Failsafe
● Destroy CMOS Checksum

Miscellaneous
□ Do Not Check ROM ID
□ Restart After Programming
Preserve SMBIOS Types:
Update MAC:

Field

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

CMOS Options

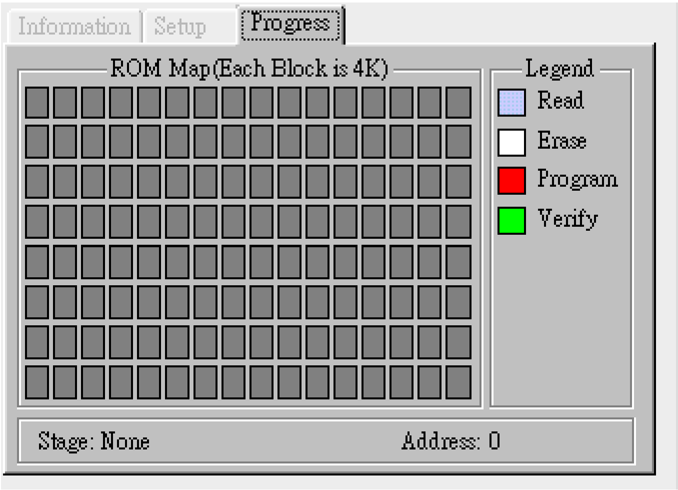
Name	Description
Nothing	Enable if you want to do nothing for CMOS after BIOS updated.
Load Current Optimal	Enable if you do like to load CMOS optimal settings from current system after BIOS updated.
Load Current Failsafe	Enable if you do like to load CMOS failsafe settings from current system after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

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

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

Progress Tab

This tab displays the updating status.



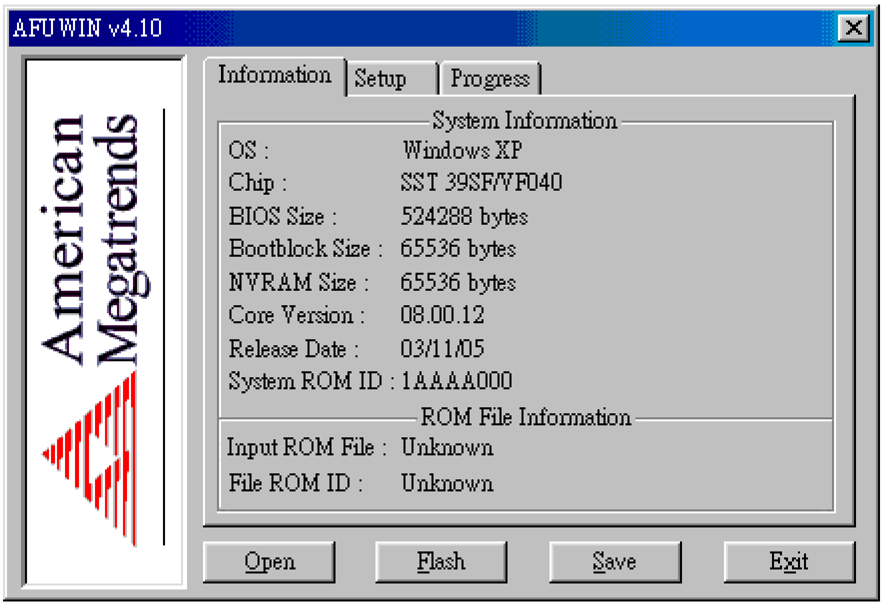
Stage: None   Address: 0

Field

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

Functions

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



System Information

OS : Windows XP
Chip : SST 39SF/VF040
BIOS Size : 524288 bytes
Bootblock Size : 65536 bytes
NVRAM Size : 65536 bytes
Core Version : 08.00.12
Release Date : 03/11/05
System ROM ID : 1AAAA000

ROM File Information

Input ROM File : Unknown
File ROM ID : Unknown

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 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 button to exit this program.

Flashing system BIOS with given file

1. Press Open 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.

Error Code List

See AFUDOS Error Code List table for detail.

Chapter 9 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	
Name	**Description**
Script File Name	This parameter is used to specify path/filename of the TEXT script file with extension.
CMOS Image File Name	This parameter is used to specify path/filename of the CMOS Image file with extension.

Rules
▪ **Any parameter encolsed 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
                    *[04]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
                    *[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**

  AMISCE /O <Script File Name>

- **Create TEXT script file and overwrite if the file existed**

  AMISCE /OX <Script File Name>

- **Create new TEXT script file to get CMOS index reg. and mask bits information**

  AMISCE /OC <Script File Name>

- **Display CMOS contents**

  AMISCE /CR

- **Save CMOS contents to file**

  AMISCE /CR <CMOS Image File Name>

- **Restore CMOS contents**

  AMISCE /CW <CMOS Image File Name>

- **Update CMOS contents by TEXT script file**

  AMISCE /I <Script File Name >

Chapter 10 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
- Microsoft® Windows® Vista 32/64
- Microsoft® Windows® PE 2.0 x64 (AMISCEWx64.EXE)

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.
• ***UCOREW64.SYS***	Driver for Microsoft® Windows® XP64.

Getting Started

Installation

Copies ***AMISCEW.EXE****,* **AMISCEWx64.*EXE* (for Microsoft Windows PE 2.0 x64),** ***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 11 AMICMOS v2.xx

Overview

AMICMOS is a CMOS RAM contents processor with command line interface. It is useful for 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.

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

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

Examples

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

- **Display CMOS contents on screen**

  AMICMOS /CR

- **Save CMOS contents as CMOS Image file**

  AMICMOS /CR <CMOS Image File Name>

- **Restore CMOS contents**

  AMICMOS /CW <CMOS Image File Name>

Copyright 2008 - NDA Required (NDA)

Chapter 12 AMIPWD (Password Update)

Overview

AMIPWD is a change ROM password utility with command line interface. It allows you to redefine ROM password without modifying BIOS Code.

Features

The utility offers you following features:

- Update password more quickly.
- Supervisor and User password can be updated at once.

Requirements

Supported Operating System

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

Usage

```
AMIPWD <TEXT Script File Name>
```

TEXT Script File Name

The input file MUST provide three syntaxes for changing password. Following lists the syntaxes of Password TEXT script file:

▪ **Current supervisor password:**	Input password behind colon(:) symbol.
▪ **New supervisor password:**	Input password behind colon(:) symbol.
▪ **New user password:**	Input password behind colon(:) symbol.

Examples

Examples on how to change ROM password using the command prompt are shown in following:

- **Create PASSWORD.TXT file for changing ROM password**
  1. *Open a TEXT editor and new a file*
  2. *Type following text in the edit window:*

     Current supervisor password:111111

     New supervisor password:222222

     New user password:333333
  3. *Save the new file and name it as PASSWORD.TXT.*
- **Change ROM password with the PASSWORD.TXT file**

  AMIPWD <PASSWORD.TXT>

Chapter 13 AMIPWDW (Password Update)

Overview

AMIPWDW is a change ROM password utility with command line interface. It allows you to redefine ROM password without modifying BIOS Code.

Features

This utility offers the following features:

- Update password more quickly.
- Supervisor and User password can be updated at once.

Requirements

Supported Operating System

AMIPWDW 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
- Microsoft® Windows® Vista 32/64

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.
• ***UCOREW64.SYS***	Driver for Microsoft® Windows® XP64.

Getting Started

Installation

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

Usage & Example

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

Chapter 14 TXTBCPD/W (AMIBCP)

Overview

TXTBCPD|W stands for TEXT BIOS Configuration Program for DOS/Windows. This program is same as AMIBCP guided in Part.2 - Chapter 3 but it works by script file. It is a command line interface program. It also 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 TXTBCPD|W, 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

The utility offers you following features:

- SETUP screen layout
- 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.
- Backup and Restore previous settings.

Requirements

Supported Operating System

TXTBCPD is supported by the following operating systems:

- MS-DOS environment

TXTBCPW 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
- Microsoft® Windows® Vista 32/64

Getting Started

Installation

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

Usage

TXTBCPD <ROM File Name> <Script File Name> <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 [Options]**	Create TEXT script file.
▪ **/I [New ROM File Name]**	Parse TEXT script file and update ROM File.

<table>
<tr><th colspan="3">Parameters List</th></tr>
<tr><th>Name</th><th colspan="2">Description</th></tr>
<tr><td>ROM File Name</td><td colspan="2">This parameter is used to specify path/filename of the ROM file with extension.</td></tr>
<tr><td>Script File Name</td><td colspan="2">This parameter is used to specify path/filename of the TEXT script file with extension.</td></tr>
<tr><td rowspan="10">Options</td><td colspan="2">If option is absent:</td></tr>
<tr><td colspan="2">TEXTBCPD|W will create script file with full information</td></tr>
<tr><td colspan="2">If option is present:</td></tr>
<tr><td>Option</td><td>Description</td></tr>
<tr><td>/BF</td><td>Output BIOS Feature Table only.</td></tr>
<tr><td>/BS</td><td>Output BIOS String Table only.</td></tr>
<tr><td>/P</td><td>Output PCI IRQ Routing Table only.</td></tr>
<tr><td>/R</td><td>Output Register Table only.</td></tr>
<tr><td>/S</td><td>Output BIOS SETUP Setting Table only.</td></tr>
<tr><td>/SL</td><td>Output BIOS SETUP Screen Layout Table only.</td></tr>
<tr><td>New ROM File</td><td colspan="2">This parameter is used to specify path/filename of the new ROM file with</td></tr>
</table>

Parameters List	
Name	**Description**
Name	extension.

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

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

Script Syntax

This section does not plan to show you detailed syntax. It introduces you the group syntax only. You can find out the explanation of whole syntax in "**AMIBCP Script Language Specification.doc**".

Following are six groups syntax supported in the script file:

■ BIOSFeaturesGroup / EndBIOSFeaturesGrooup

This group allows you to view and configure some of the AMIBIOS features.
You can view the following fields:

- BIOS Date
- BIOS Name
- BIOS Size
- BIOS Tag
- Sign On Message
- OEM Data(If available)

■ BIOSStringsGroup / EndBIOSStringsGroup

This group allows you to view and edit AMIBIOS strings.

■ PCIIRQRoutingTableGroup / EndPCIIRQRoutingTableGroup

This group allows you to view and modify the *PCI IRQ Routin*g 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

- **RegisterTableGroup / EndRegisterTableGroup**

  This group allows you to edit the AMIBIOS register tables. Each table contains register, data, or other fields that can be edited.

- **SetupScreenLayoutGroup / EndSetupScreenLayoutGroup**

  This group allows you to change SETUP screen layout.
  The item format is:

#	X	：	XXXX	SPACE	Object Type/Object Name

Field	Description
#	It means the item was hidden. Do not remove this signature.
X	Level number. User can modify this field to adjust item level.
：	Separator.
XXXX	Index ID. Do not change.
SPACE	Separator.
Object	Object Type and Object Name. Do not change.

- **SetupDataGroup / EndSetupDataGroup**

  This group 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:

  - Show Or hide setup screen and setup questions
  - Modify access levels and usage
  - Edit failsafe and optimal values

Examples

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

- **Create TEXT script file with whole information**

  TXTBCPD(Or TXTBCPW) <ROM File Name> <Script File Name> /O

- **Create TEXT script file with BIOS Features only**

  TXTBCPD(Or TXTBCPW) <ROM File Name> <Script File Name> /O /BF

- **Create TEXT script file with BIOS Strings only**

  TXTBCPD(Or TXTBCPW) <ROM File Name> <Script File Name> /O /BS

- **Create TEXT script file with PCI IRQ Routing Tables only**

  TXTBCPD(Or TXTBCPW) <ROM File Name> <Script File Name> /O /P

- **Create TEXT script file with Register Tables only**

  TXTBCPD(Or TXTBCPW) <ROM File Name> <Script File Name> /O /R

- **Create TEXT script file with SETUP Settings only**

  TXTBCPD(Or TXTBCPW) <ROM File Name> <Script File Name> /O /S
- **Create TEXT script file with SETUP Screen Layout only**

  TXTBCPD(Or TXTBCPW) <ROM File Name> <Script File Name> /O /SL
- **Update BIOS settings by given TEXT script file**

  TXTBCPD(Or TXTBCPW) <ROM File Name> <Script File Name> /I
- **Update BIOS settings and output new ROM File by given TEXT script file**

  TXTBCPD(Or TXTBCPW) <ROM File Name> <Script File Name> /I <New ROM File Name>

Chapter 15 AMIDELNX (DMIEdit)

Overview

AMIDELNX is a SMBIOS data manipulation 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)
- System Configuration Options (Type 12)

Requirements

Supported Operating System

AMIDELNX Utility is supported in following operating system:

- Linux CORE v2.4/2.6

BIOS Requirements

System BIOS should have the followings:

- AMIBIOS CORE version 8.xx.xx.
- ***SMIFlash*** *eModule* with “8.00.00_SMIFlash-1.00.10” label or later.
- ***SMBIOS*** *eModule* with “8.00.08_SMB-3.1.02_CORE_RC20” label or later.

Getting Started

Installation

- Copies **AMIDELNX.TGZ** to any storage location accessible by the host system.
- Extracts the contents to same directory. You will get 4 files, **amidelnx_26_32, amidelnx_26_64, amidelnx_24_32, and amidelnx_24_64** executables.
- Type ‘uname -r’ to identify kernel version. If it says 2.4.xx…, you should use to **amidelnx_24_xx** file, otherwise, enter to **amidelnx_26_xx** file.
- To determine which version of Linux system, type 'uname -m'. Under shell screen. If it says 'x86_64', then ***amidelnx_xx_64*** should be used; otherwise, use the ***midelnx_xx_32***.
- Run the suitable file. AMIDELNX will generate necessary drivers and load it automatically. If AMIDELNX cannot work well, please refer to “Generating driver file manually” or “Troubleshooting” section to get help.

Generating driver file manually

1. Log in Linux as root.
2. The compiler suite(gcc) must be installed. If these packages are not installed, the driver CANNOT be built.
3. For most distributions, AMIDELNX will generate AMI Flash Driver file automatically without notification. Certainly, the driver file may NOT be generated in some specific case and the loading driver failure message will be displayed. If you get this error, first, you can read 'Q1' and 'Q2' in 'TROUBLESHOOTING section' to shut out the kernel issues, and second, you can see Point.4 below to create driver file by yourself and launch AMIDELNX again.
4. Kernel sources must be installed, *CONFIGURED*, and then compiled. Following are steps to do this:

   4.1 Find Running Kernel's Configuration File:

   - To configure the sources, simply change to the kernel source directory (typically /lib/modules/$(uname -r)/build). If it doesn't exist, you need to install kernel source. Typically, the reference configuration for the kernel can be found in the /boot directory with filename '.config', 'kernel.config', or 'vmlinux-2.4.18-3.config'. Type 'uname -a' and use the configuration filename that best matches the output from 'uname -a'.

- On some distributions Red Hat for instance, there is a config directory under /lib/modules/$(uname -r)/build.

- Copy this configuration file into the root of the linux kernel source tree(usually it is /lib/modules/$(uname -r)/build). This file must be renamed to ".config"(dot config).

4.2 Make Your AMI Flash Driver(amifldrv_mod.drv):

For most distribution, the command to build the driver is:

```
amidelnx_26_32 /MAKEDRV
        or
amidelnx_26_64 /MAKEDRV
        or
amidelnx_24_32 /MAKEDRV
        or
amidelnx_24_64 /MAKEDRV
```

- If your linux's kernel source tree is under /lib/modules/$(uname -r)/build instead of the default path '/lib/modules/$(uname -r)/build', add a KERNEL flag:

```
amidelnx_26_32 /MAKEDRV KERNEL=/lib/modules/$(uname -r)/build
    or
amidelnx_26_64 /MAKEDRV KERNEL=/lib/modules/$(uname -r)/build
    or
amidelnx_24_32 /MAKEDRV KERNEL=/lib/modules/$(uname -r)/build
    or
amidelnx_24_64 /MAKEDRV KERNEL=/lib/modules/$(uname -r)/build
```

- If KERNEL is omitted, the default is /lib/modules/$(uname -r)/build.This should work for MOST distributions.

4.3 Make Your AMI Flash Driver from dirver source files (amifldrv_mod.o):

Using command /GENDRV, it will generate driver source files to specific directory.

```
        amidelnx_26_32 /GENDRV [Option 1] [Option 2]
  or
        amidelnx_26_64 /GENDRV [Option 1] [Option 2]
  or
        amidelnx_24_32 /GENDRV [Option 1] [Option 2]
  or
        amidelnx_24_64 /GENDRV [Option 1] [Option 2]
```

[Option 1]: Specific kernel source 'KERNEL=XXXX' same as the /MAKEDRV
[Option 2]: Specific output directory 'OUTPUT=XXXX'

Generate files as below:

File Name	Description
amiwrap.c	Driver source code.
amiwrap.h	Driver header.
amifldrv.o_shipped	Object file for driver. kernel version 2.6 (For kernel version 2.4 object file is amifldrv.o)
Makefile	Makefile

For most distribution, the command to build the driver is:

```
make
```

If your linux's kernel source tree is under /lib/modules/$(uname -r)/build instead of the default path '/lib/modules/$(uname -r)/build', add a KERNEL flag:

```
make KERNEL=/lib/modules/$(uname -r)/build
```

If KERNEL is omitted, the default is /lib/modules/$(uname -r)/build. This should work for MOST distributions.

4.4 Check Your Build:

Check the version of running Linux kernel with 'uname -r'.
Check the version of amidmidrv with 'modinfo amifldrv.o'.

If they mismatch, you will need to select the correct configuration file(.config), rebuild your kernel, and then rebuild your driver as described in (4.1), (4.2), (4.3). and (4.4).

The amidmidrv must be in same directory with amidelnx_26_32(amidelnx_26_64, amidelnx_24_32,amidelnx_24_64).If they match, continue on to the 'AMIDELNX' section to run amidelnx.

Troubleshooting

Q1: I get following error message when loading driver:

"insmod: error inserting 'amifldrv_mod.o': -1 Invalid module format".

A1: Most likely this is cause by wrong configuration file and your kernel refuses to accept your driver because version strings(more precisely, version magic) do not match.

To check the version of running Linux kernel, type "uname -r".
To check the version of amifldrv_mod.drv, type "modinfo amifdrv_mod.drv"

If they mismatch, you will need to select the correct configuration file(.config), rebuild your kernel, and then rebuild your driver as described in “Generating driver file manually” section.

Q2: When I run ./afulnx_32(./afulnx_64), it says "Unable to load driver".

A2: Some Linux distributions do not display driver debug messages on screen by default. Type "dmesg" to see those debug messages. This is very likely the same problem as Q1.

Q3: When I run ./afulnx_32(./afulnx_64), it simply freezes.

A3: This is caused by a Linux feature called "NMI Watchdog" which is used to debug Linux kernel. This feature must be disabled to run AFULNX2.
Please do "cat /proc/interrupts" twice and check if NMI is counting.
If it is, then boot Linux with a kernel parameter "nmi_watchdog=N" where N is either 0, 1 or 2. Find out which configuration can halt NMI from counting by "cat /proc/interrupts" This is the configuration we should use to run AFULNX2.

Q4: I get following error message When running midelnx_26_32 (all other version).

A4: Most likely this is cause by old version of SMBIOS module and SMIFlash module.You can update SMBIOS module to "SMB-3.1.02RC20" label or later, and update SMIFlash module to "8.00_SMIFlash_1.00.10" label or later.

Usage & Example

This utility is same as AMIDEDOS.EXE but running under Linux. So you can see Usage of AMIDEDOS and Example of AMIDEDOS to learn more information on command usage.
Notice: AMIDELNX common usage commands are sync with AMIDEDOS. For other details then please read readme.txt which comes along with amidelnx.tgz.

Part 2: Graphical User Interface Mode

Chapter 1 OEMLOGO (ChangeLogo)

Overview

OEMLOGO (ChangeLogo) is a logo modification utility with a 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
- Microsoft® Windows® Vista 32/64

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)

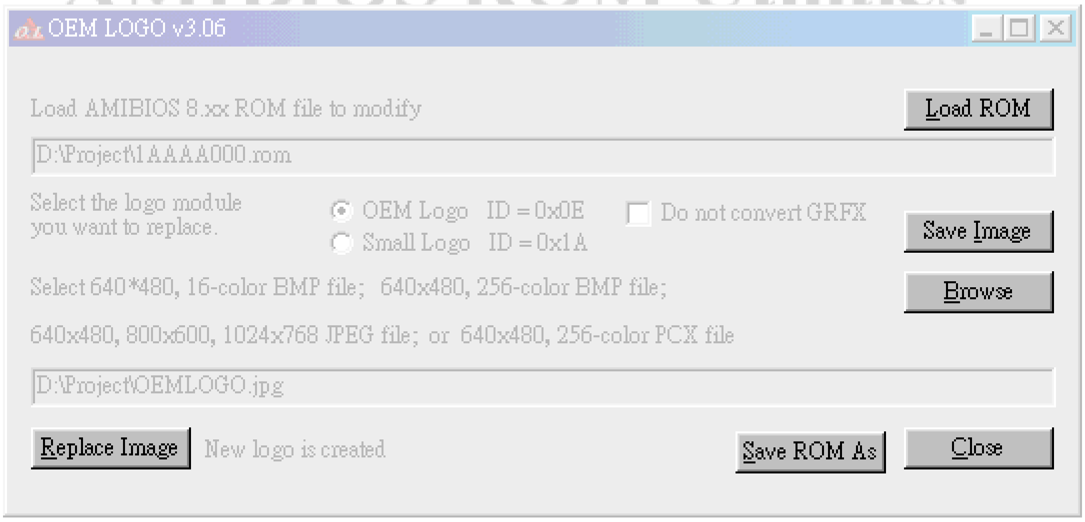
Note: Small OEM Logo does support only 640*80, 16-Color Bitmap format.

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



Load AMIBIOS 8.xx ROM file to modify

D:\Project\1AAAA000.rom

Select the logo module you want to replace.

● OEM Logo ID = 0x0E
○ Small Logo ID = 0x1A

□ Do not convert GRFX

Select 640*480, 16-color BMP file; 640x480, 256-color BMP file;
640x480, 800x600, 1024x768 JPEG file; or 640x480, 256-color PCX file

D:\Project\OEMLOGO.jpg

New logo is created

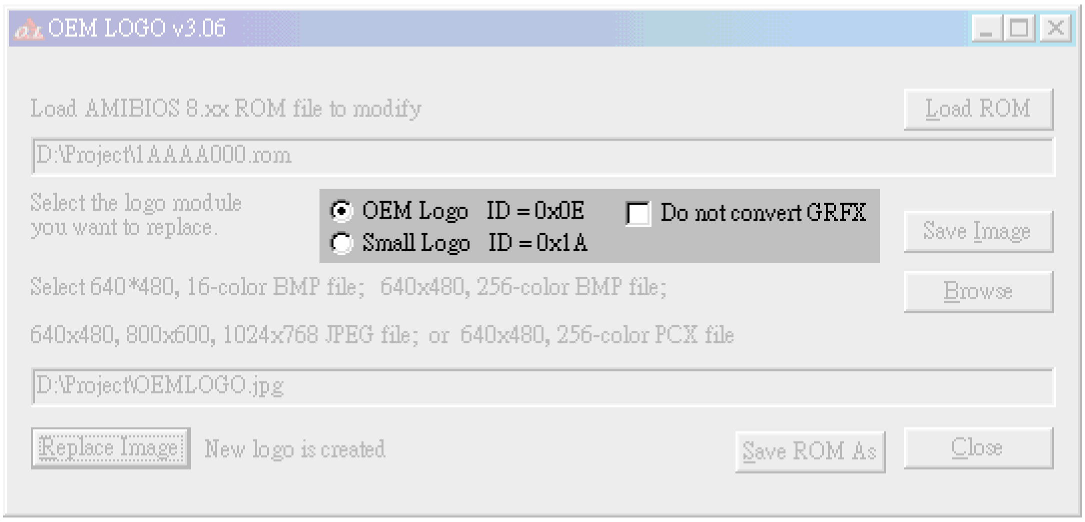
Load ROM	Click this button to search for BIOS ROM file from any disk drive.
Save Image	Click this button to extract Logo Image from ROM to any disk drive by given PATH/FileName.
Browse	Click this button to search for new Logo Image file from any disk drive.

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

Save ROM As — Click this button to save the changes that you have made to the BIOS ROM file. You can also specify the location and to change the existing file name.

Close — Click this button to exit the program.

Options



Load AMIBIOS 8.xx ROM file to modify

D:\Project\1AAAA000.rom

Select the logo module you want to replace.

● OEM Logo ID = 0x0E
○ Small Logo ID = 0x1A
□ Do not convert GRFX

Select 640*480, 16-color BMP file; 640x480, 256-color BMP file;

640x480, 800x600, 1024x768 JPEG file; or 640x480, 256-color PCX file

D:\Project\OEMLOGO.jpg

New logo is created

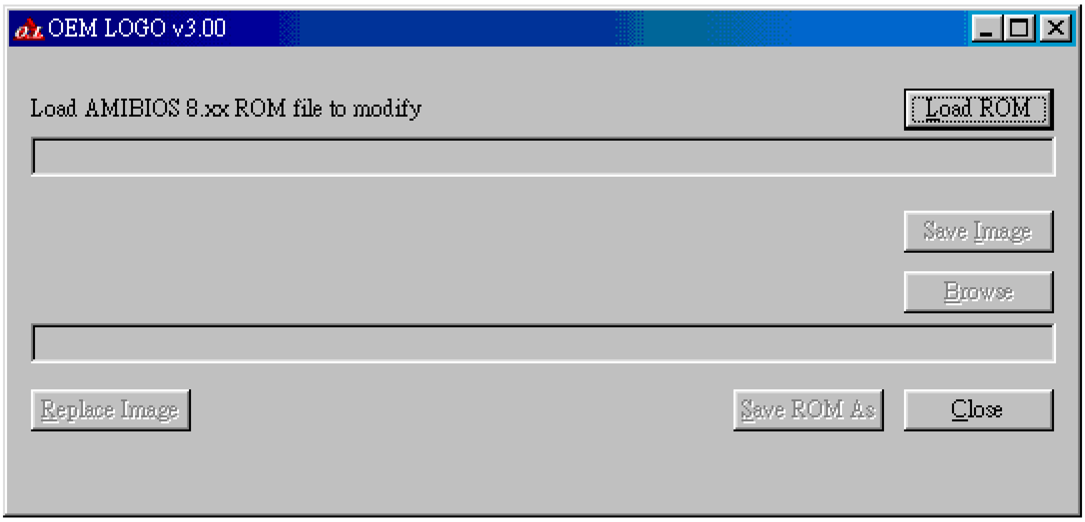
OEM Logo ID=0x0E Choose this option can change OEM logo module.

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:

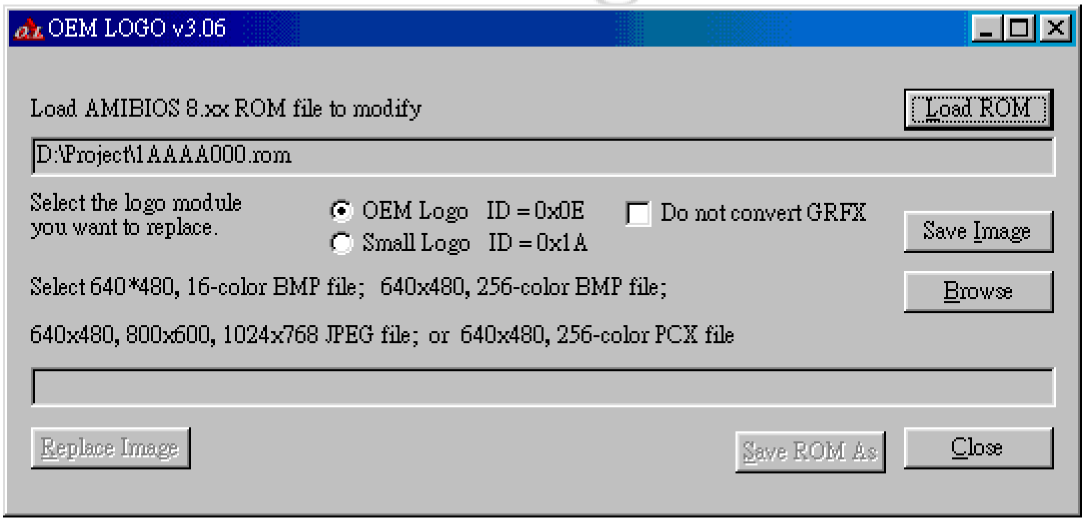


Load AMIBIOS 8.xx ROM file to modify

And refer to the following steps to change new logo:

Changing OEM Logo

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



Load AMIBIOS 8.xx ROM file to modify

D:\Project\1AAAA000.rom

Select the logo module
you want to replace.

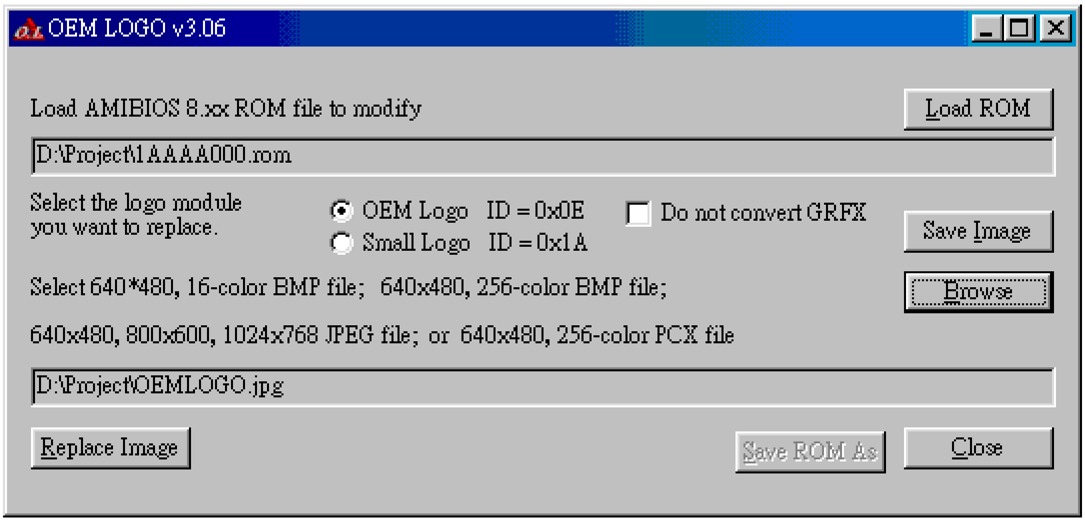
● OEM Logo ID = 0x0E
○ Small Logo ID = 0x1A

□ Do not convert GRFX

Select 640*480, 16-color BMP file; 640x480, 256-color BMP file;

640x480, 800x600, 1024x768 JPEG file; or 640x480, 256-color PCX file

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



OEM LOGO v3.06

Load AMIBIOS 8.xx ROM file to modify

D:\Project\1AAAA000.rom

Select the logo module you want to replace.

● OEM Logo ID = 0x0E
○ Small Logo ID = 0x1A

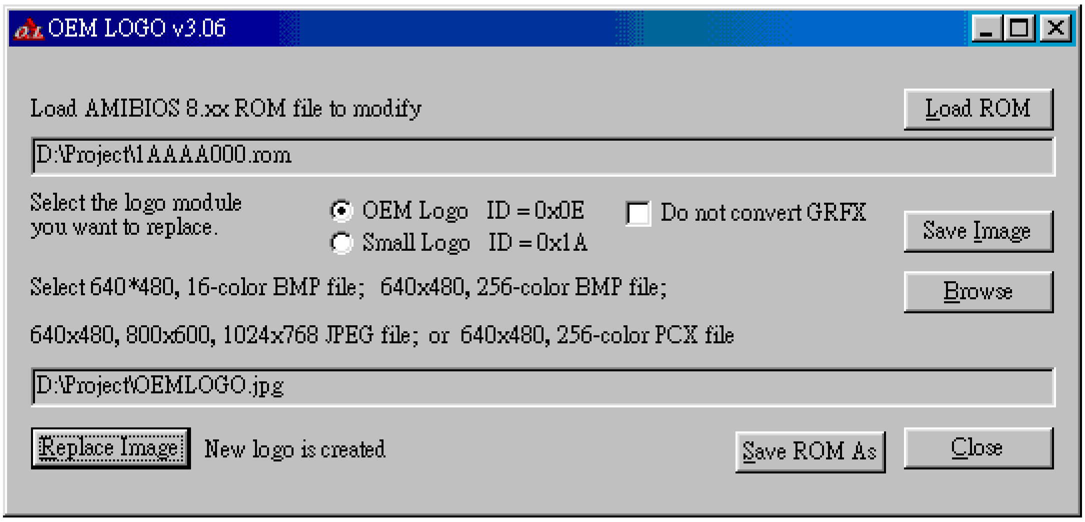
□ Do not convert GRFX

Select 640*480, 16-color BMP file; 640x480, 256-color BMP file;

640x480, 800x600, 1024x768 JPEG file; or 640x480, 256-color PCX file

D:\Project\OEMLOGO.jpg

3. Click Replace Image button to change logo module. If the function works fine, *New logo is created* message will be displayed behind the button.



OEM LOGO v3.06

Load AMIBIOS 8.xx ROM file to modify

D:\Project\1AAAA000.rom

Select the logo module you want to replace.

● OEM Logo ID = 0x0E
○ Small Logo ID = 0x1A

□ Do not convert GRFX

Select 640*480, 16-color BMP file; 640x480, 256-color BMP file;

640x480, 800x600, 1024x768 JPEG file; or 640x480, 256-color PCX file

D:\Project\OEMLOGO.jpg

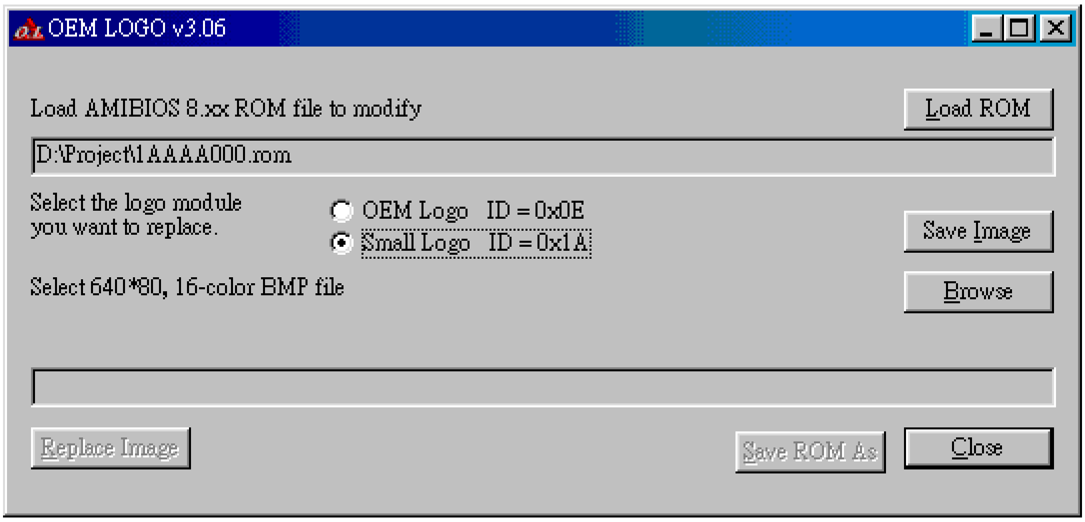
New logo is created

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.



OEM LOGO v3.06

Load AMIBIOS 8.xx ROM file to modify

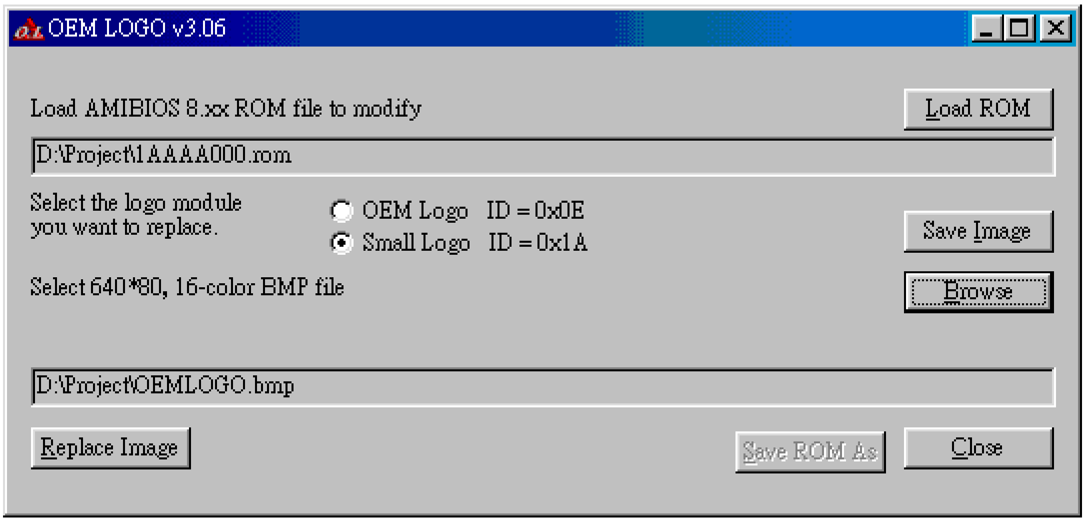
D:\Project\1AAAA000.rom

Select the logo module you want to replace.

○ OEM Logo ID = 0x0E
● Small Logo ID = 0x1A

Select 640*80, 16-color BMP file

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



OEM LOGO v3.06

Load AMIBIOS 8.xx ROM file to modify

D:\Project\1AAAA000.rom

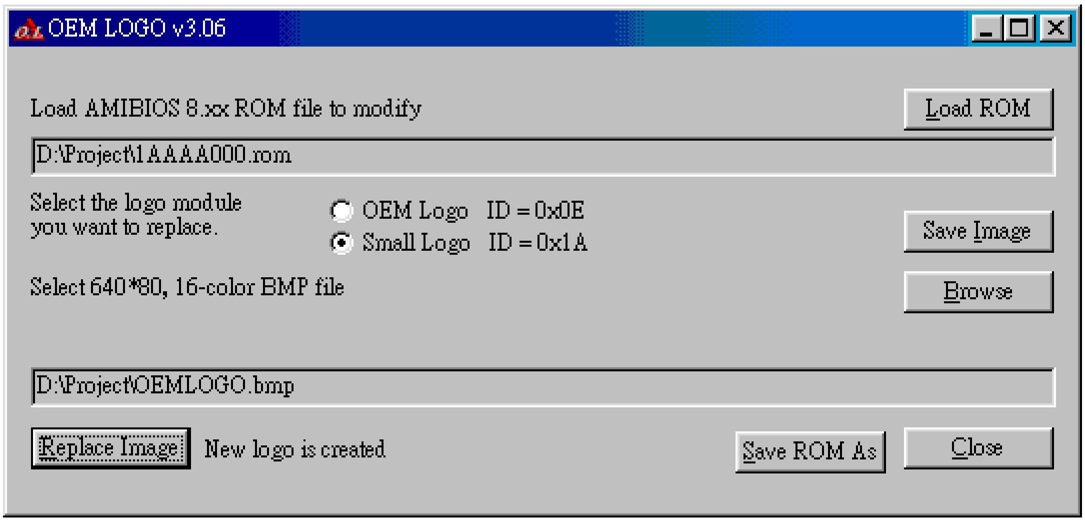
Select the logo module you want to replace.

○ OEM Logo ID = 0x0E
● Small Logo ID = 0x1A

Select 640*80, 16-color BMP file

D:\Project\OEMLOGO.bmp

3. Click Replace Image button to change logo module. If the function works fine, *New logo is created* message will be displayed in right of the button.



OEM LOGO v3.06

Load AMIBIOS 8.xx ROM file to modify

D:\Project\1AAAA000.rom

Select the logo module you want to replace.

○ OEM Logo ID = 0x0E
● Small Logo ID = 0x1A

Select 640*80, 16-color BMP file

D:\Project\OEMLOGO.bmp

New logo is created

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

5. Click Close button to exit program.

Chapter 2 MMTOOL v3.xx

Overview

MMTOOL is a BIOS module manipulation tool with graphical user interface. It allows you to manage AMIBIOS8 modules that are contained in a BIOS ROM file.

Features

This utility offers following features:

- Insert Module (including BIOS Option ROM)
- 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
- Microsoft® Windows® Vista 32/64

BIOS Requirements

The loaded BIOS ROM file should have the followings:

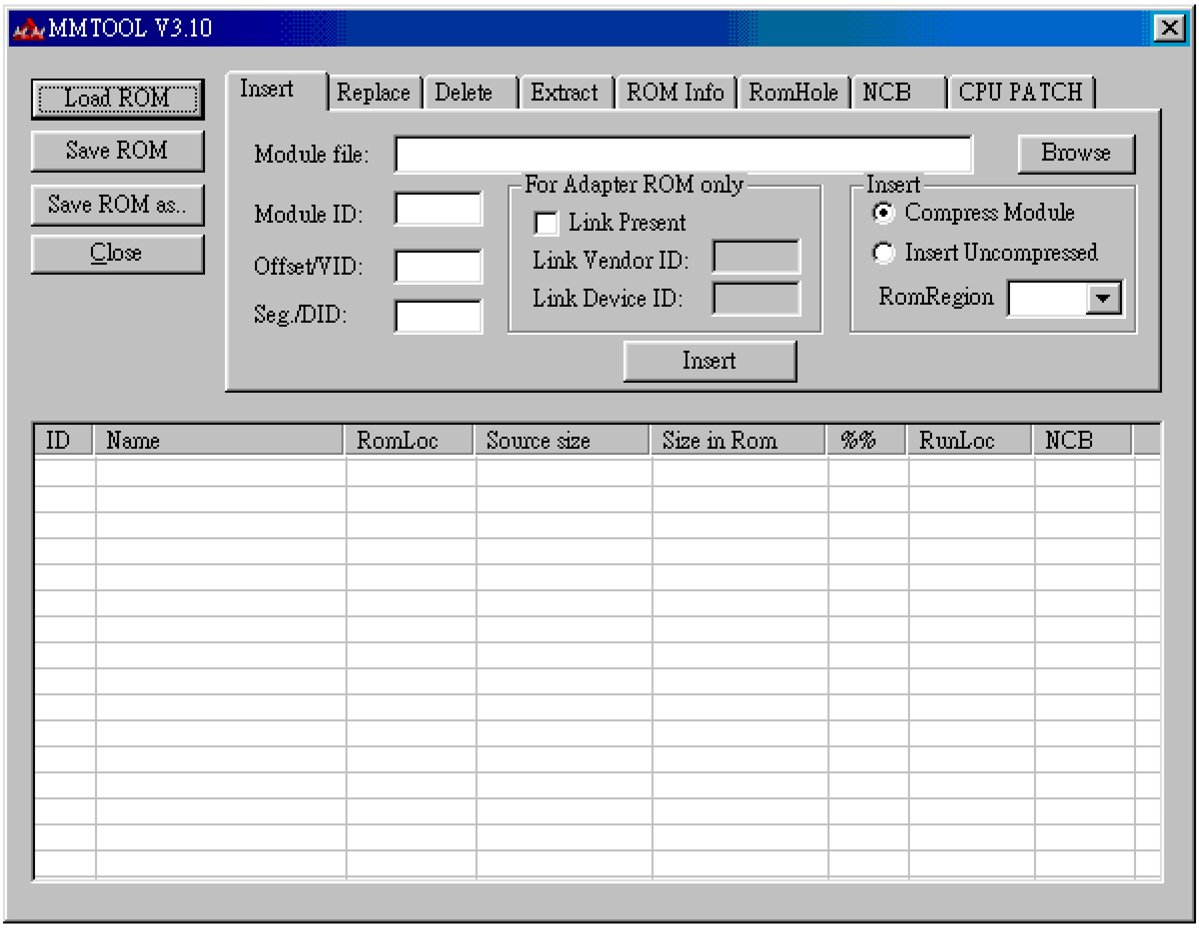
- The file MUST be an AMIBIOS ROM file (Core version 8.xx.xx only)
- BIOS ROM file should be building via “8.00.08_AMITOOLS_17” label or above.

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



ID	Name	RomLoc	Source size	Size in Rom	%%	RunLoc	NCB

Main Buttons

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

Click this button to save the changes you have mode to BIOS ROM file that is currently opened.

Click this button to save the changes you have made to the BIOS

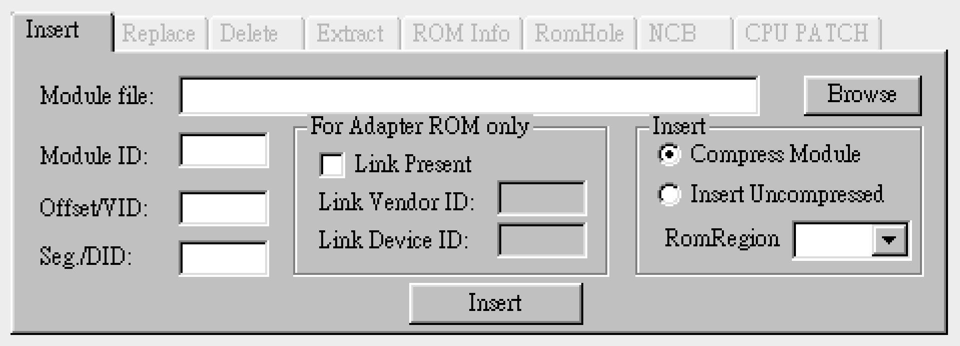
ROM file. You can also specify the location and to change the existing file name.

Close — Click this button to exit the program.

Function Frame

Insert Module

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



Field

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

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

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

Buttons

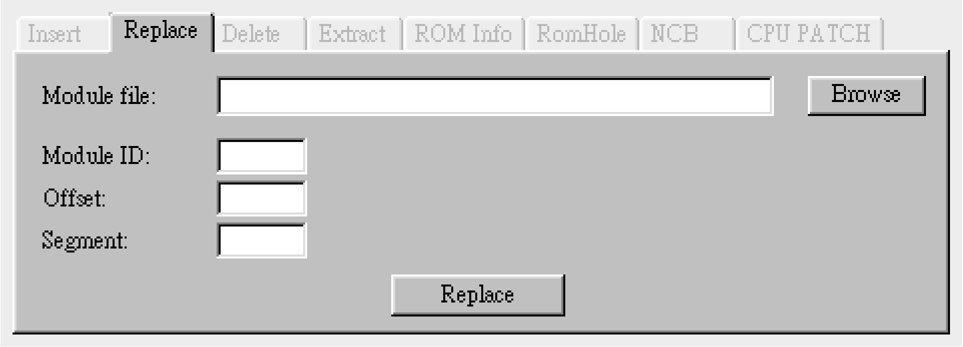
Icon	Description
Browse	This button is used to search for a new module file from any storage location.
Insert	This button is used to launch the insert module operation.

Options

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

Replace Module

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



Field

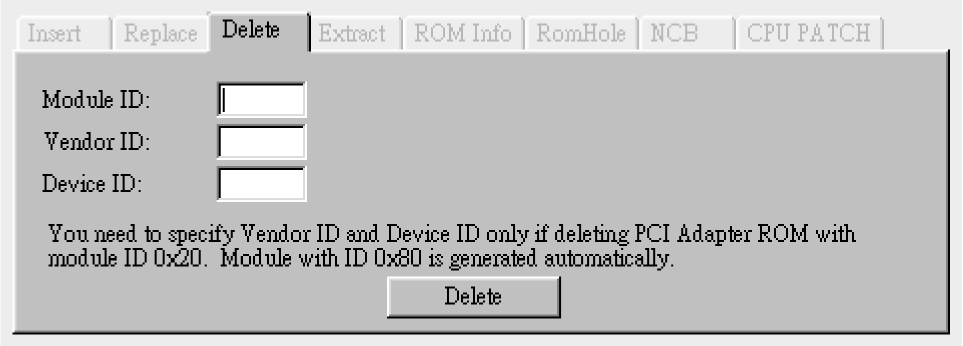
Name	Description
Module File	This field is used to specify path/filename of new module file with extension.
Module ID	2-digits hexadecimal Module ID. See Appendix A Module ID Codes for detail.
Offset Vendor ID	This filed is used to enter a new module runtime Offset. This field is optional except when 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.
Segment Device ID	This filed is used to enter a new module runtime Segment. 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.

Buttons

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

Delete Module

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



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

Name	Description
Module ID	2-digits hexadecimal Module ID. See Appendix A Module ID Codes for detail.
Offset Vendor ID	This filed 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.
Segment Device ID	This filed 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.

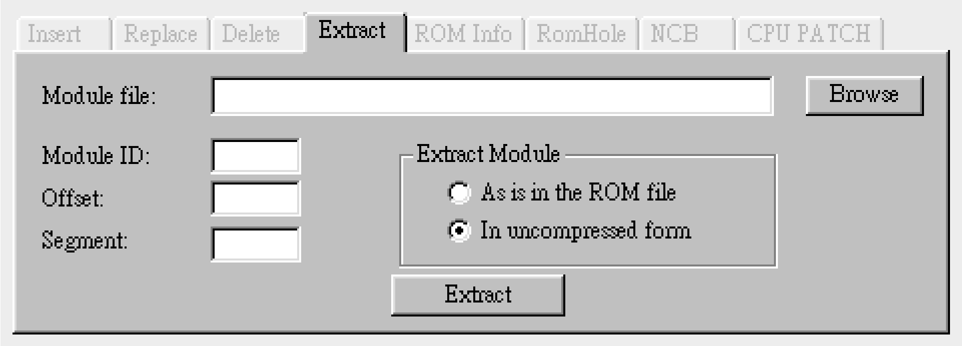
Buttons

Icon	Description
Delete	This button is used to launch the delete module operation.

Note: The original BIOS ROM file is not modified unless you use *Save ROM* button or the *Save ROM As* button to save changes.

Extract Module

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



○ As is in the ROM file
● In uncompressed form

Note: The BIOS Module is saved to selected file.

Field

Name	Description
Module File	This field is used to specify path/filename of new module file with extension.
Module ID	2-digits hexadecimal Module ID. See Appendix A Module ID Codes for detail.
Offset Vendor ID	This filed 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 VendorID for the Adapter ROM.

Name	Description
Segment Device ID	This filed 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 DeviceID for the Adapter ROM.

Buttons

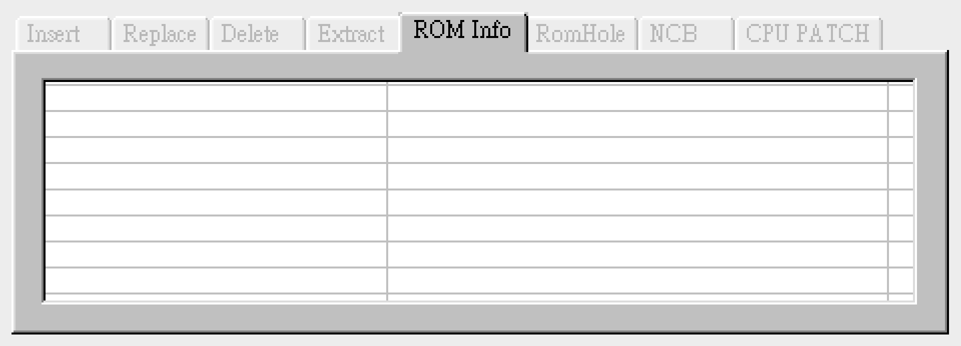
Icon	Description
Browse	This button is used to search for a new module file from any storage location.
Extract	This button is used to launch the extract module operation.

Options

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

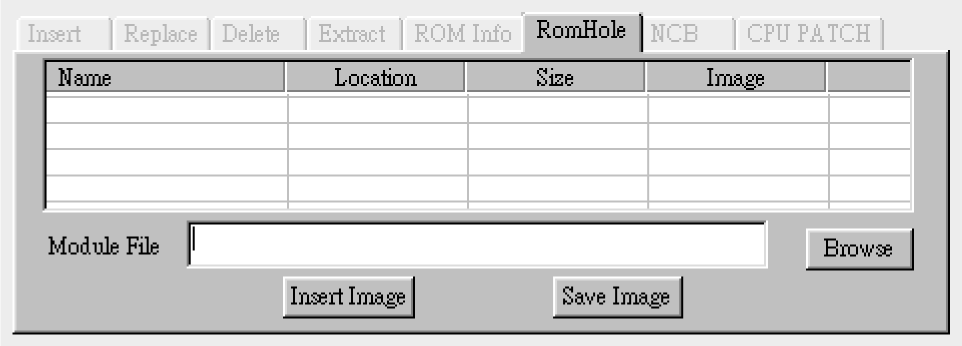
Note: Extracting a BIOS module will not affect the BIOS ROM file.

ROM Information



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

ROM Hole



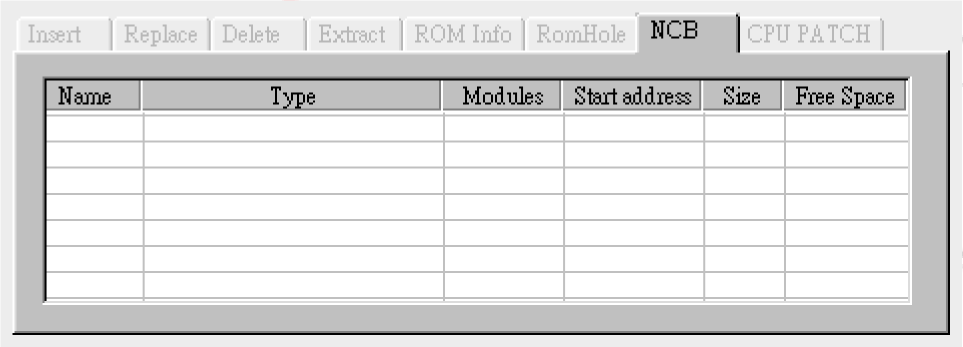
Field

Name	Description
Name	This field displays ROM Hole number in BIOS ROM file.
Location	This field displays the start address of ROM Hole in BIOS ROM file.
Size	This filed display the ROM Hole size.
Image	If a ROM Hole contains data, this field displays “Yes”, or it will be “No”.

Buttons

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

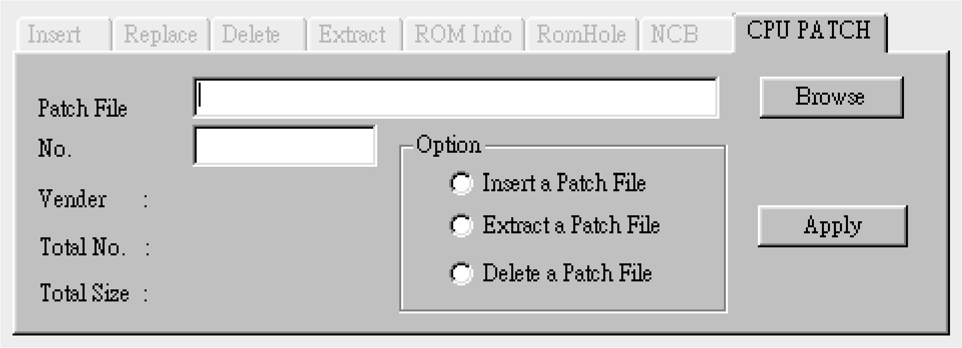
NCB Information



Name	Type	Modules	Start address	Size	Free Space

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

CPU Patch



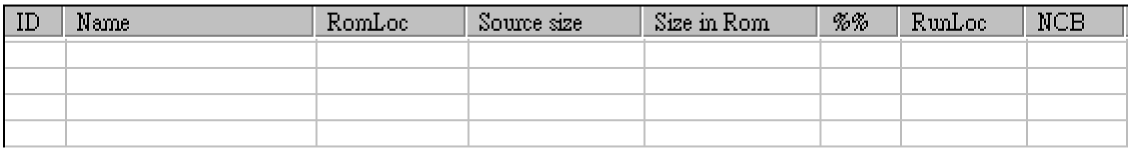
Field

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

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

Module Info Frame

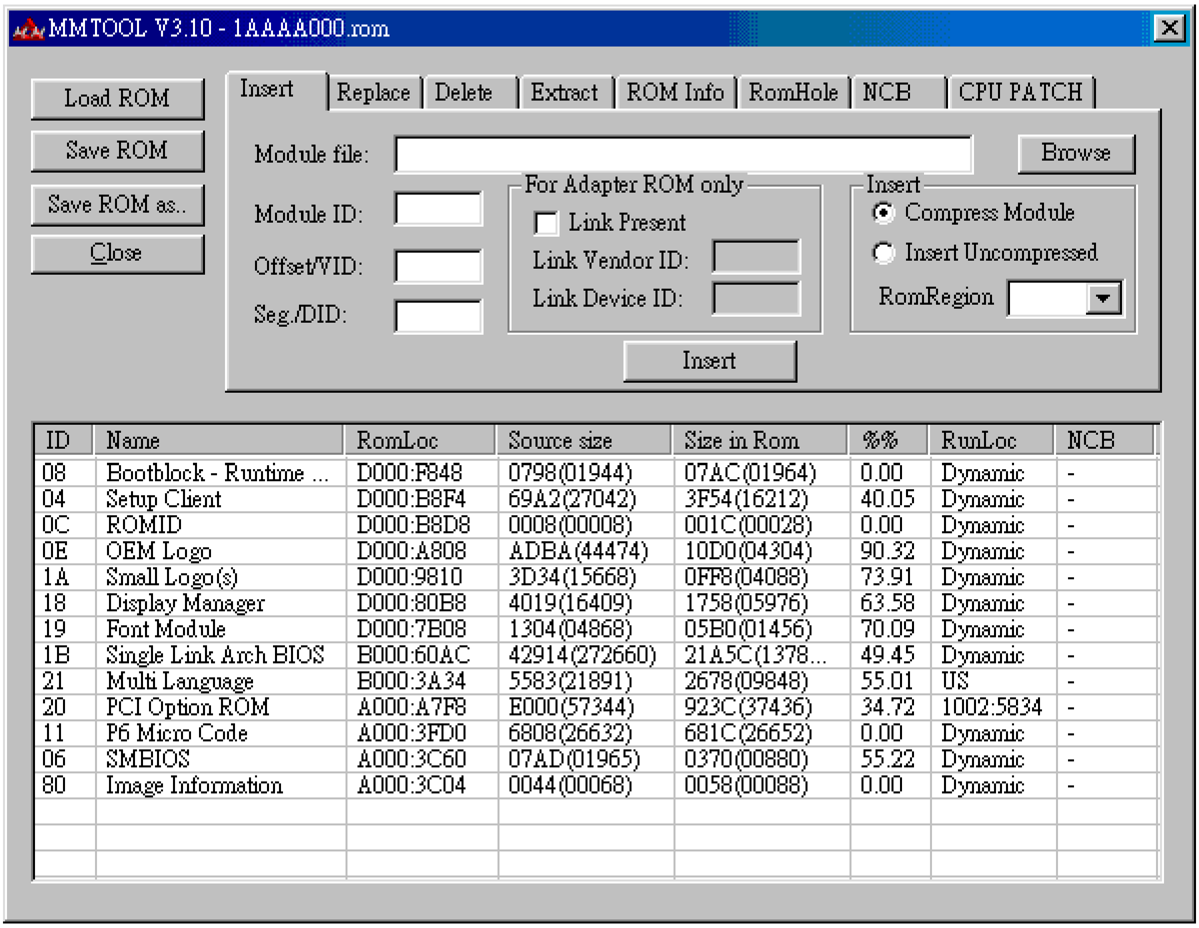


Field	Description
ID	2-digits hexadecimal Module ID. See Appendix A Module ID Codes for detail.
Name	BIOS module name. See Appendix A Module ID Codes for detail.
RomLoc	Module data location in BIOS ROM image.
Source Size	Original module data source size in unit of bytes..
Size in Rom	Actual module data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

Field	Description
NCB	This unique name identifies the Non-Critical Block. If present, the module will be inserted to the region.

Functions

To use MMTOOL, you can double-click the executable file icon to open the operating window and press Load ROM button to load a BIOS ROM file:



ID	Name	RomLoc	Source size	Size in Rom	%%	RunLoc	NCB
08	Bootblock - Runtime ...	D000:F848	0798(01944)	07AC(01964)	0.00	Dynamic	-
04	Setup Client	D000:B8F4	69A2(27042)	3F54(16212)	40.05	Dynamic	-
0C	ROMID	D000:B8D8	0008(00008)	001C(00028)	0.00	Dynamic	-
0E	OEM Logo	D000:A808	ADBA(44474)	10D0(04304)	90.32	Dynamic	-
1A	Small Logo(s)	D000:9810	3D34(15668)	0FF8(04088)	73.91	Dynamic	-
18	Display Manager	D000:80B8	4019(16409)	1758(05976)	63.58	Dynamic	-
19	Font Module	D000:7B08	1304(04868)	05B0(01456)	70.09	Dynamic	-
1B	Single Link Arch BIOS	B000:60AC	42914(272660)	21A5C(1378...	49.45	Dynamic	-
21	Multi Language	B000:3A34	5583(21891)	2678(09848)	55.01	US	-
20	PCI Option ROM	A000:A7F8	E000(57344)	923C(37436)	34.72	1002:5834	-
11	P6 Micro Code	A000:3FD0	6808(26632)	681C(26652)	0.00	Dynamic	-
06	SMBIOS	A000:3C60	07AD(01965)	0370(00880)	55.22	Dynamic	-
80	Image Information	A000:3C04	0044(00068)	0058(00088)	0.00	Dynamic	-

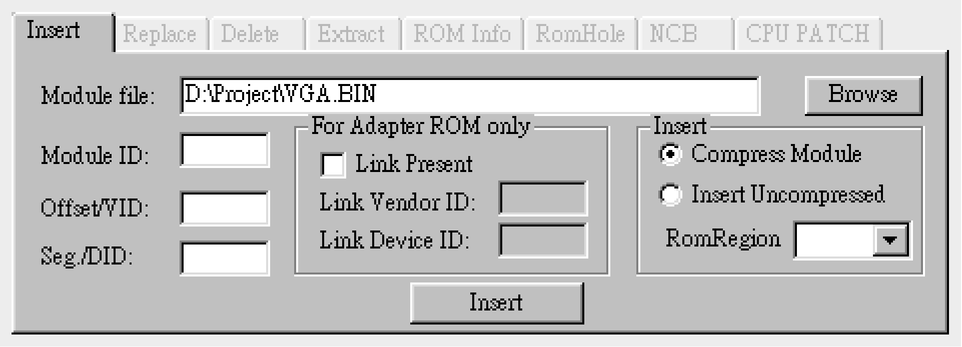
And refer the following steps to manipulate modules:

Inserting Generic Module

You can insert new BIOS module by following steps:

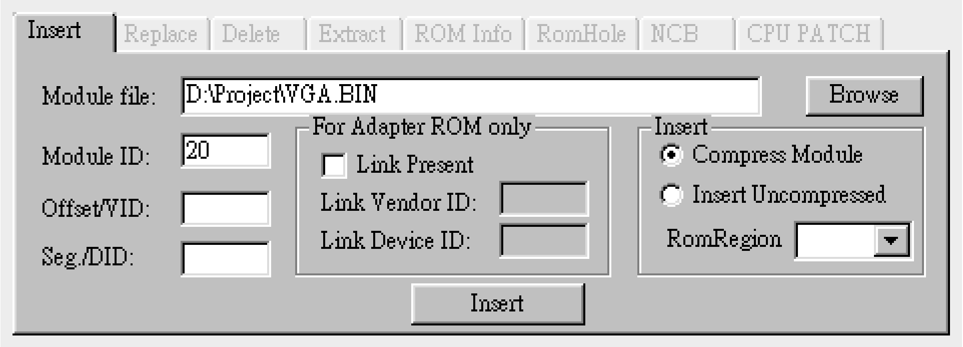
1. Switch to *Insert* tab and click Browse button to specify the new module file location

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



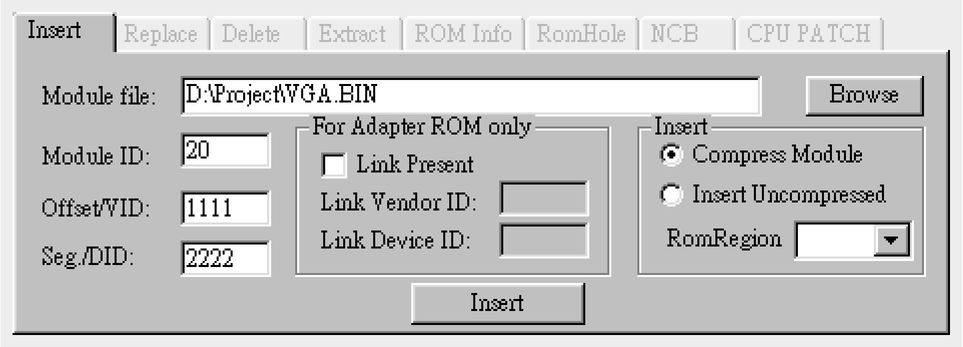
Module file: D:\Project\VGA.BIN
Module ID:
Offset/VID:
Seg./DID:
□ Link Present
● Compress Module
○ Insert Uncompressed

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



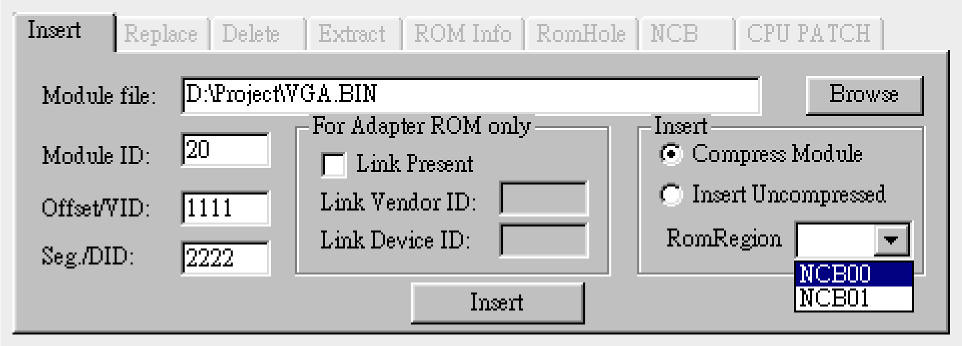
Module file: D:\Project\VGA.BIN
Module ID: 20
Offset/VID:
Seg./DID:
□ Link Present
● Compress Module
○ Insert Uncompressed

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.



Module file: D:\Project\VGA.BIN
Module ID: 20
Offset/VID: 1111
Seg./DID: 2222
□ Link Present
● Compress Module
○ Insert Uncompressed

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.



Module file: D:\Project\VGA.BIN

Module ID: 20

Offset/VID: 1111

Seg./DID: 2222

For Adapter ROM only

□ Link Present

Link Vendor ID:

Link Device ID:

Insert

● Compress Module

○ Insert Uncompressed

RomRegion: NCB00, NCB01

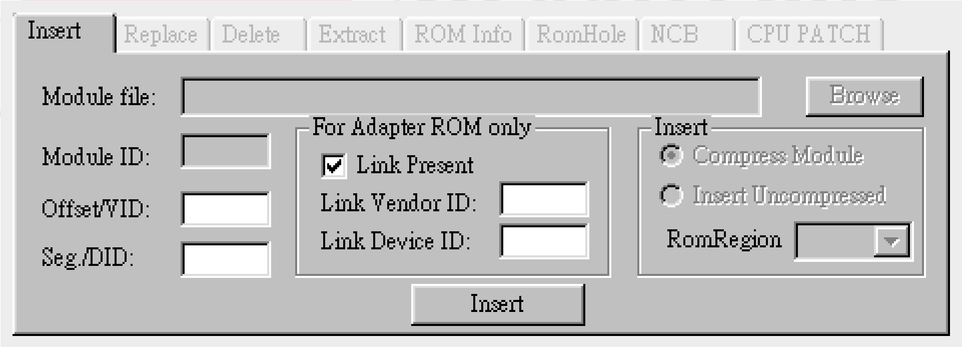
5. Click Insert button to insert the new module into the BIOS ROM image.

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.



Module file:

Module ID:

Offset/VID:

Seg./DID:

For Adapter ROM only

☑ Link Present

Link Vendor ID:

Link Device ID:

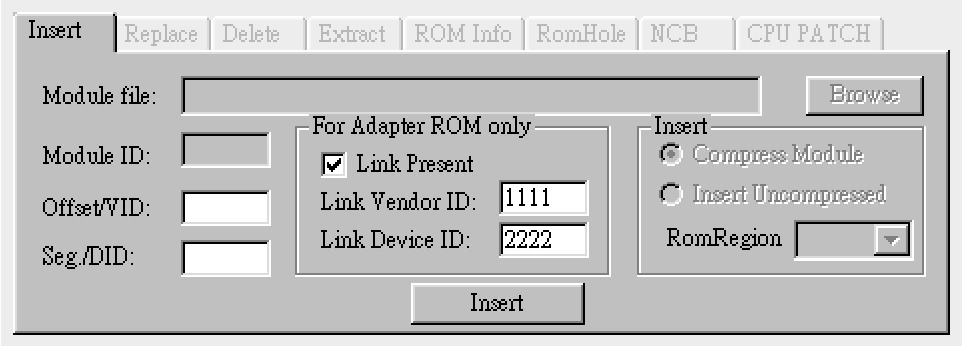
Insert

● Compress Module

○ Insert Uncompressed

RomRegion

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.



Module file:

Module ID:

Offset/VID:

Seg./DID:

For Adapter ROM only

☑ Link Present

Link Vendor ID: 1111

Link Device ID: 2222

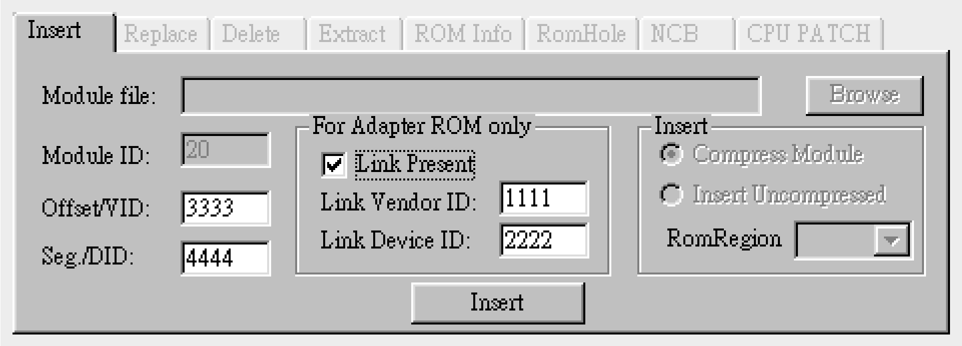
Insert

● Compress Module

○ Insert Uncompressed

RomRegion

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



Module file:

Module ID: 20

Offset/VID: 3333

Seg./DID: 4444

For Adapter ROM only

☑ Link Present

Link Vendor ID: 1111

Link Device ID: 2222

Insert

● Compress Module

○ Insert Uncompressed

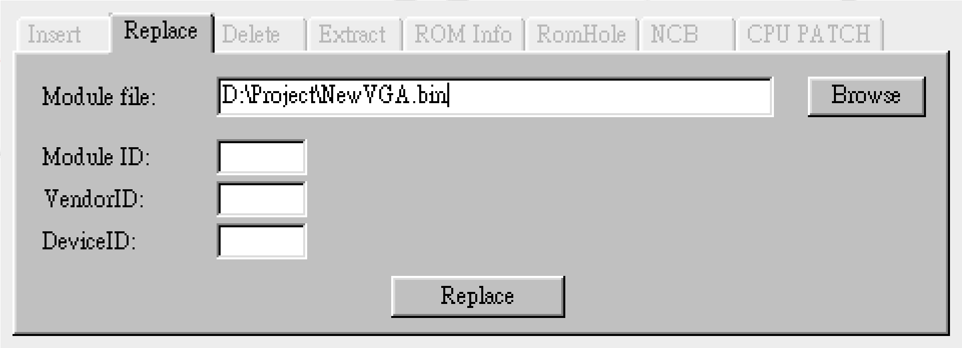
RomRegion

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.



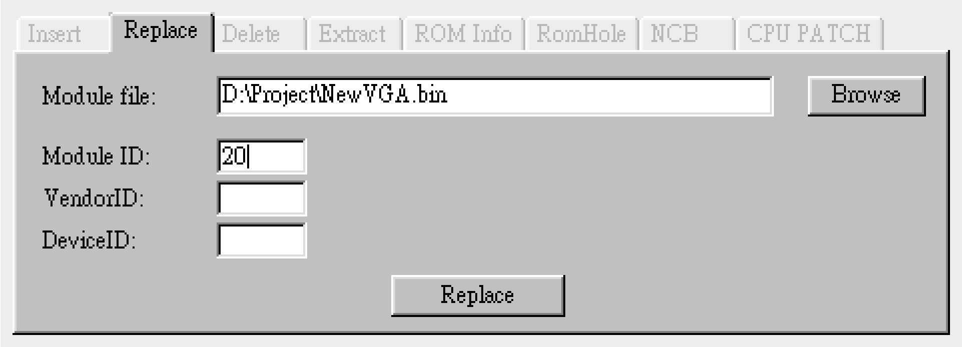
Module file: D:\Project\NewVGA.bin

Module ID:

VendorID:

DeviceID:

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



Module file: D:\Project\NewVGA.bin

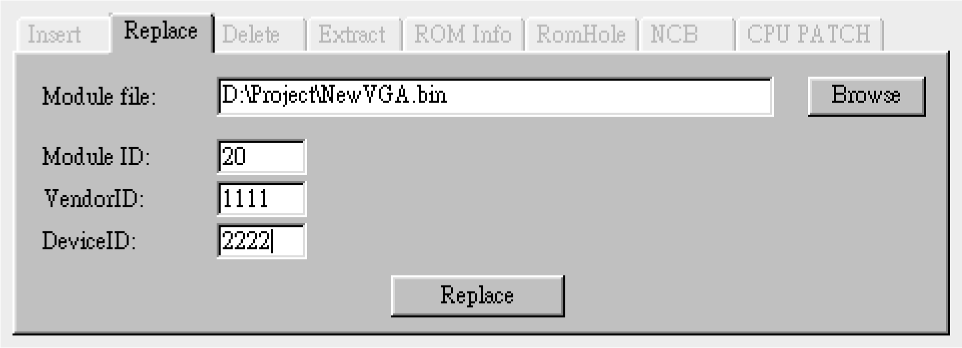
Module ID: 20

VendorID:

DeviceID:

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.



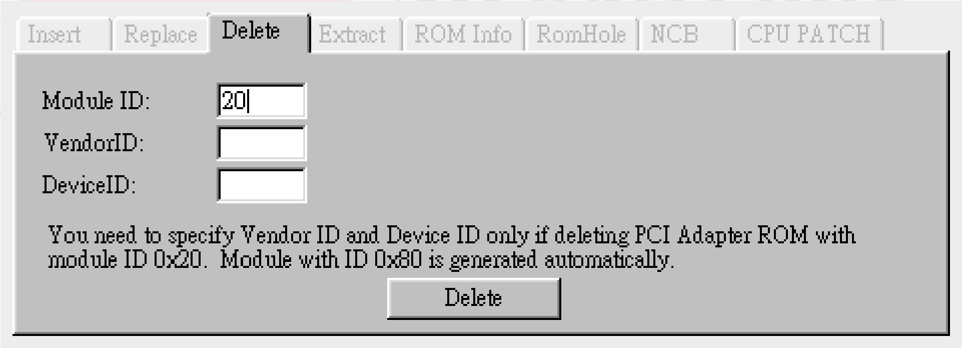
Module file: D:\Project\NewVGA.bin
Module ID: 20
VendorID: 1111
DeviceID: 2222

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:

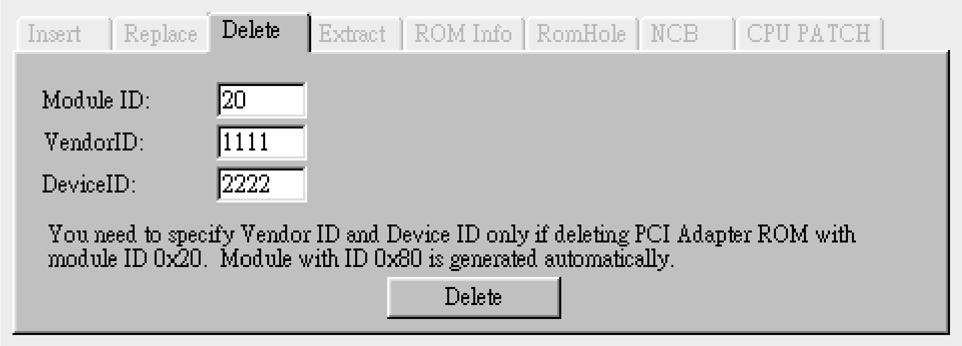
1. Switch to *Delete* tab and type the module ID into the *Module ID* field Or select the module to be deleted from the module info frame.



Module ID: 20
VendorID:
DeviceID:

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.

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.



Module ID: 20
VendorID: 1111
DeviceID: 2222

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.

3. Click Delete button to remove the module from the BIOS ROM image.

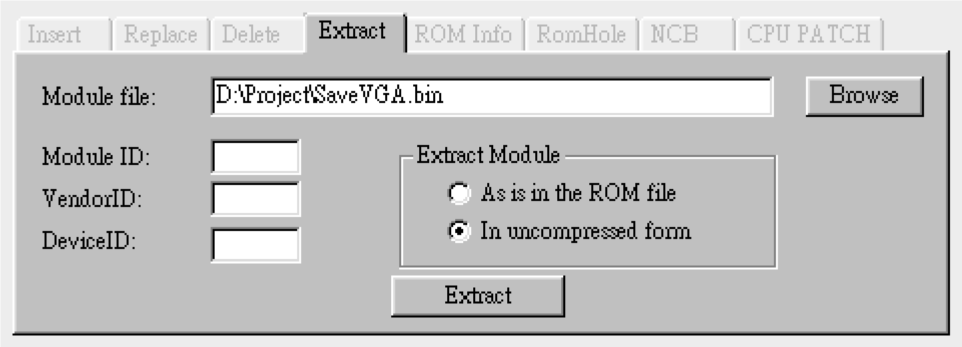
Note: Deleting a BIOS module can cause critical BIOS errors. It can also cause the BIOS to halt the system.

Extracting Module

You can extract BIOS module by following steps:

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

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



Module file: D:\Project\SaveVGA.bin

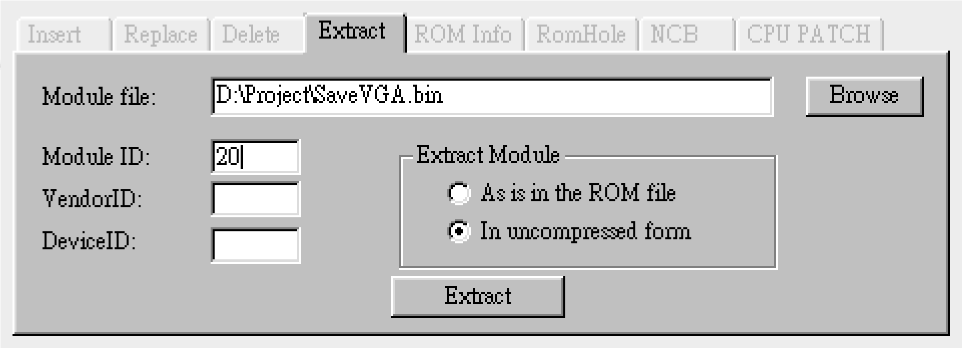
Module ID:

VendorID:

DeviceID:

Extract Module
○ As is in the ROM file
● In uncompressed form

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



Module file: D:\Project\SaveVGA.bin

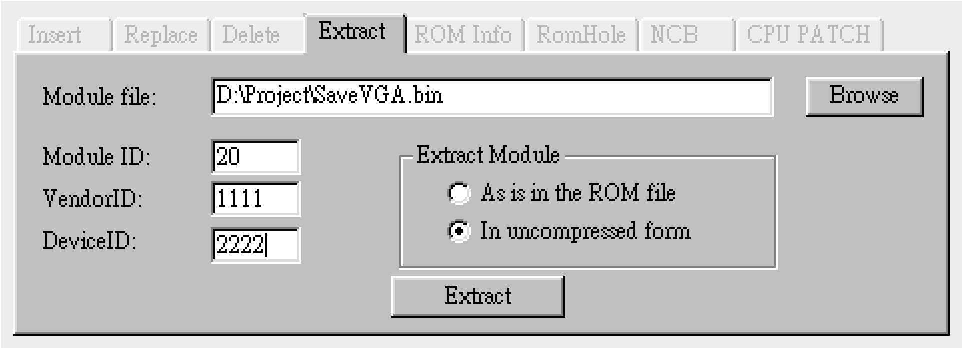
Module ID: 20

VendorID:

DeviceID:

Extract Module
○ As is in the ROM file
● In uncompressed form

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.



Module file: D:\Project\SaveVGA.bin

Module ID: 20

VendorID: 1111

DeviceID: 2222

Extract Module
○ As is in the ROM file
● In uncompressed form

4. Select one of option buttons(*As is in the ROM file* Or *In uncompressed form*) to decide how the module is to be extracted. The default option is *In uncompressed form*.

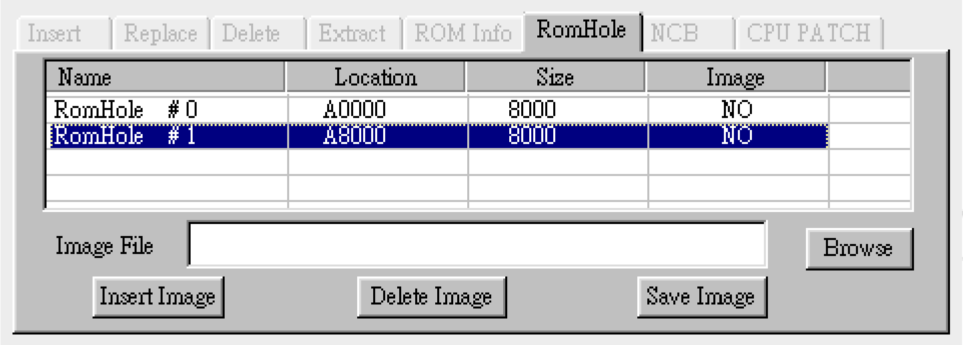
5. Click Extract button to extract the existing module.

Note: It is recommended to extract the module in uncompressed form. BIOS module must not be compressed twice.

Inserting image into a ROM Hole

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

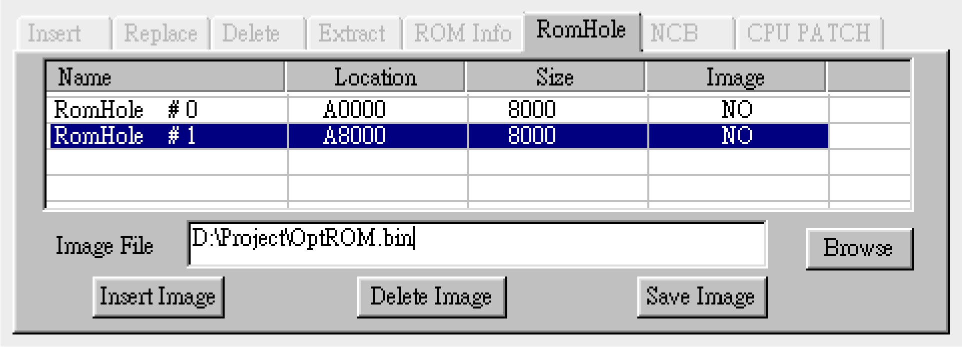
1. Switch to *ROMHole* tab and select a target ROM Hole on the list.



Name	Location	Size	Image
RomHole # 0	A0000	8000	NO
RomHole # 1	A8000	8000	NO

Image File

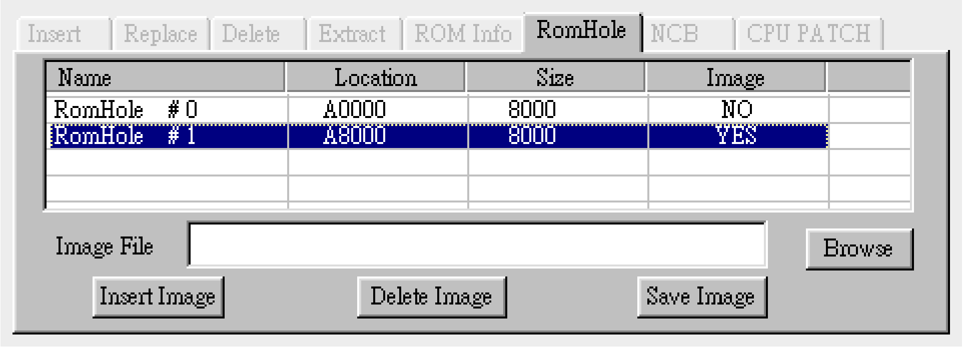
2. Click Browse button to select input image file location Or type the path and the file name in the *Image File* field.



Name	Location	Size	Image
RomHole # 0	A0000	8000	NO
RomHole # 1	A8000	8000	NO

Image File D:\Project\OptROM.bin

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”.



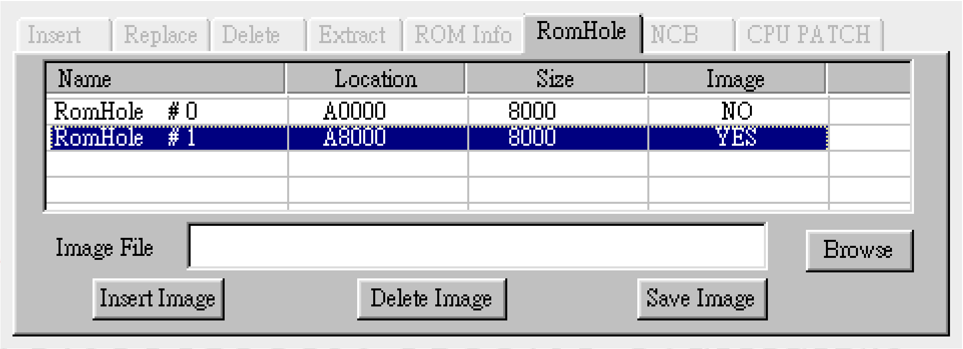
Name	Location	Size	Image
RomHole # 0	A0000	8000	NO
RomHole # 1	A8000	8000	YES

Image File:

Saving ROM Hole image to file

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

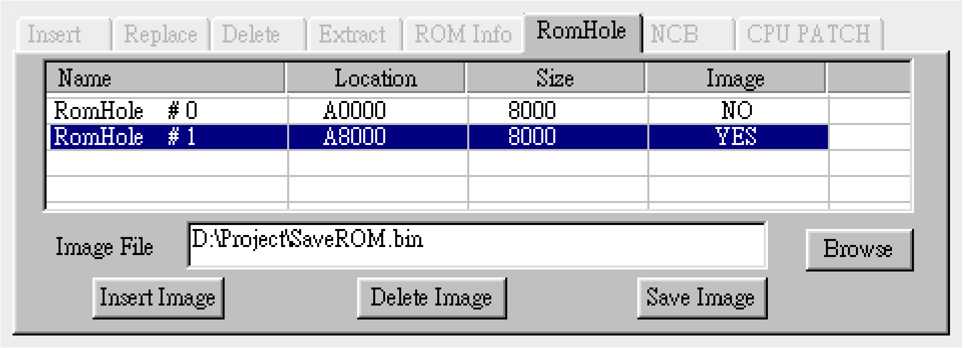
1. Switch to *ROMHole* tab and select a target ROM Hole on the list.



Name	Location	Size	Image
RomHole # 0	A0000	8000	NO
RomHole # 1	A8000	8000	YES

Image File:

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



Name	Location	Size	Image
RomHole # 0	A0000	8000	NO
RomHole # 1	A8000	8000	YES

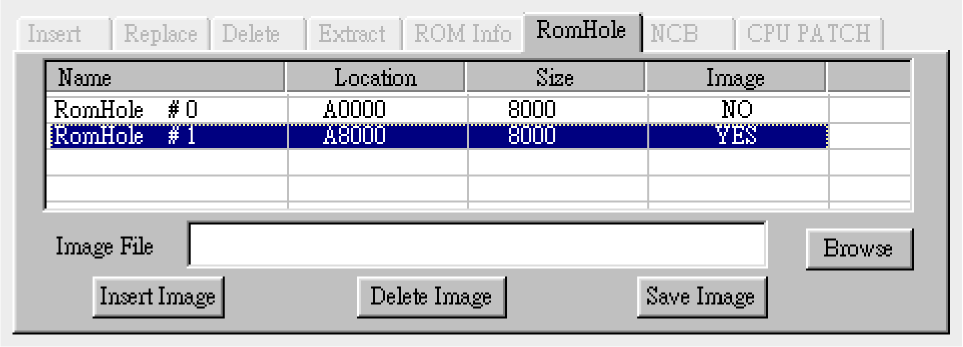
Image File: D:\Project\SaveROM.bin

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

Deleting ROM Hole image

You can delete ROM Hole image by following steps:

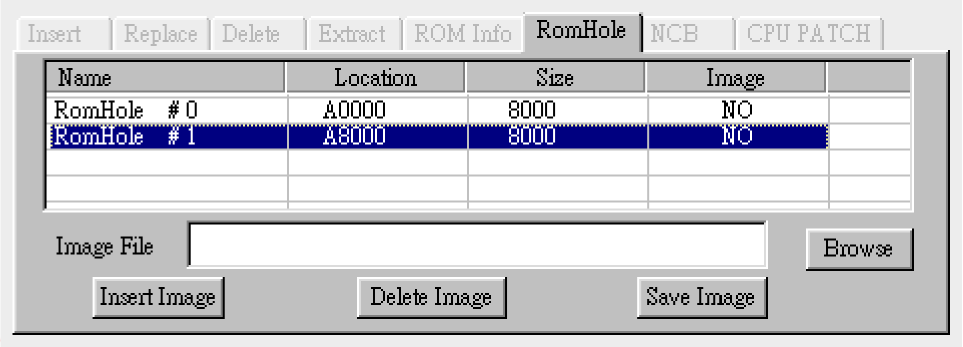
4. Switch to *ROMHole* tab and select a target ROM Hole on the list.



Name	Location	Size	Image
RomHole # 0	A0000	8000	NO
RomHole # 1	A8000	8000	YES

Image File

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



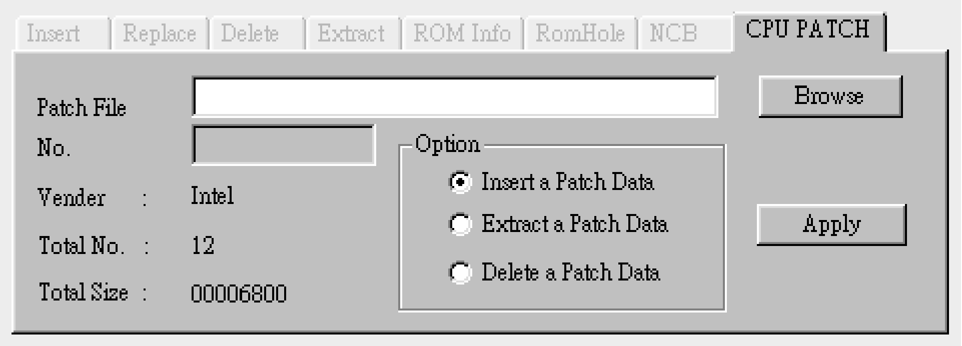
Name	Location	Size	Image
RomHole # 0	A0000	8000	NO
RomHole # 1	A8000	8000	NO

Image File

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.



Patch File

No.

Vender : Intel

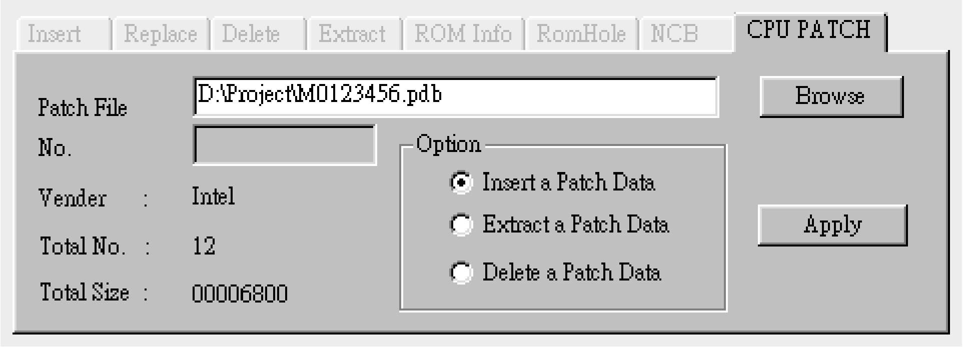
Total No. : 12

Total Size : 00006800

Option

● Insert a Patch Data
○ Extract a Patch Data
○ Delete a Patch Data

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



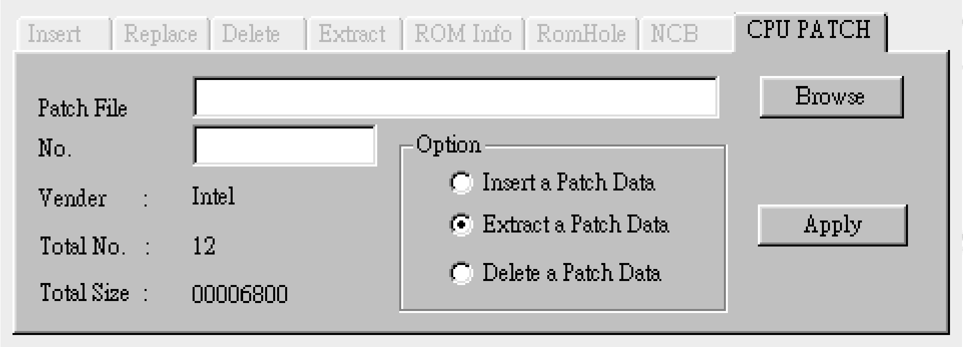
Patch File D:\Project\M0123456.pdb
No.
Vender : Intel
Total No. : 12
Total Size : 00006800
Option
● Insert a Patch Data
○ Extract a Patch Data
○ Delete a Patch Data

3. Click Apply button to insert the patch data.

Extracting a patch data

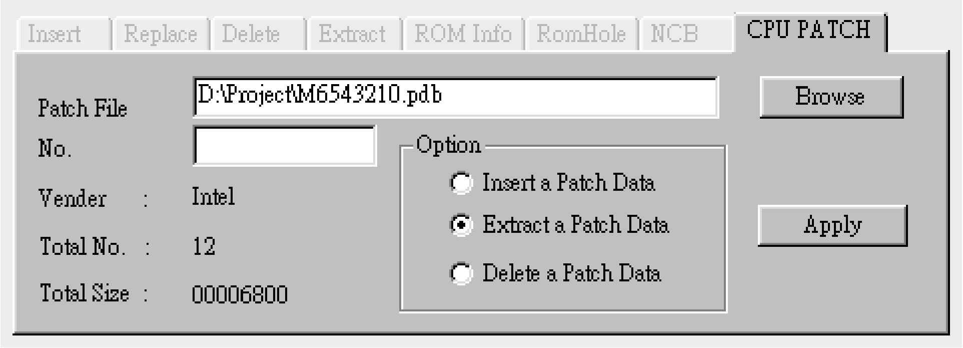
You can extract a patch data from CPU MicroCode Patch module by following steps:

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



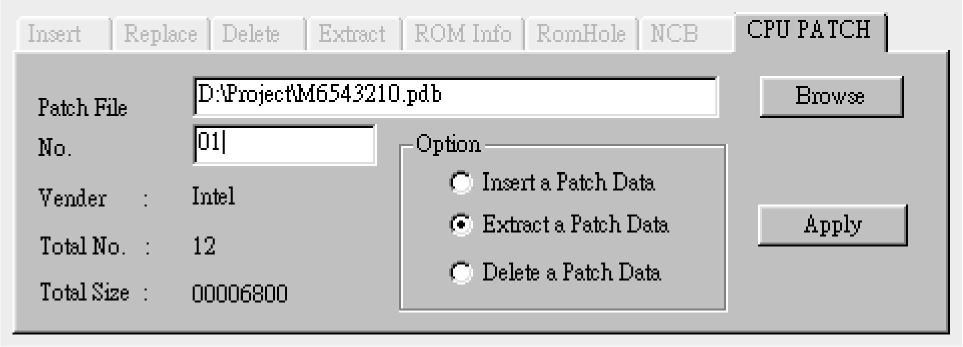
Patch File
No.
Vender : Intel
Total No. : 12
Total Size : 00006800
Option
○ Insert a Patch Data
● Extract a Patch Data
○ Delete a Patch Data

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



Patch File D:\Project\M6543210.pdb
No.
Vender : Intel
Total No. : 12
Total Size : 00006800
Option
○ Insert a Patch Data
● Extract a Patch Data
○ Delete a Patch Data

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



Patch File D:\Project\M6543210.pdb

No. 01|

Vender : Intel

Total No. : 12

Total Size : 00006800

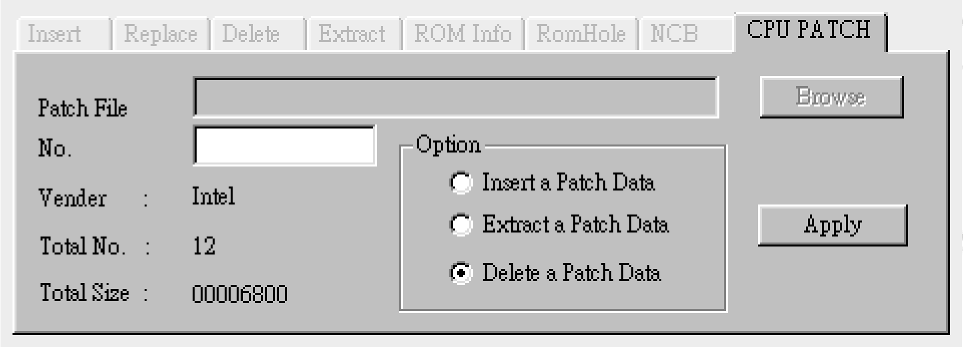
Option
○ Insert a Patch Data
● Extract a Patch Data
○ Delete a Patch Data

4. Click Apply 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 *CPUPatch* tab and enable *Delete a Patch Data* at Option block.



Patch File

No.

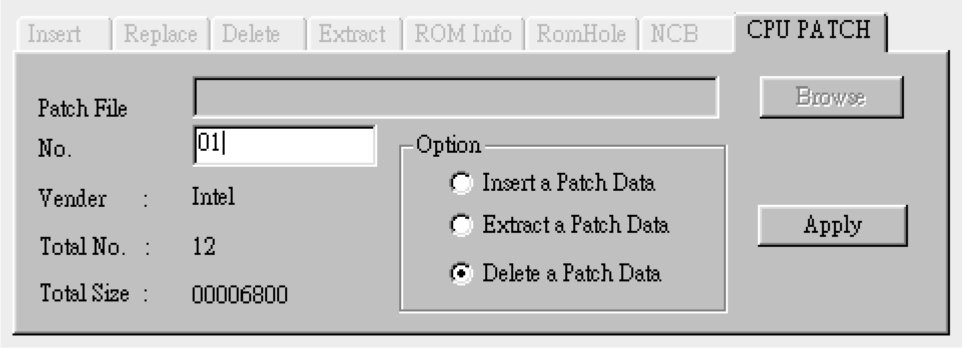
Vender : Intel

Total No. : 12

Total Size : 00006800

Option
○ Insert a Patch Data
○ Extract a Patch Data
● Delete a Patch Data

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



Patch File

No. 01|

Vender : Intel

Total No. : 12

Total Size : 00006800

Option
○ Insert a Patch Data
○ Extract a Patch Data
● Delete a Patch Data

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 (v3.30 and above will not support any more.)
- Microsoft® Windows® ME
- Microsoft® Windows® NT 4.0
- Microsoft® Windows® 2000
- Microsoft® Windows® XP/XP64
- Microsoft® Windows® PE
- Microsoft® Windows® Vista 32/64

BIOS Requirements

The loaded BIOS ROM file should have the followings:

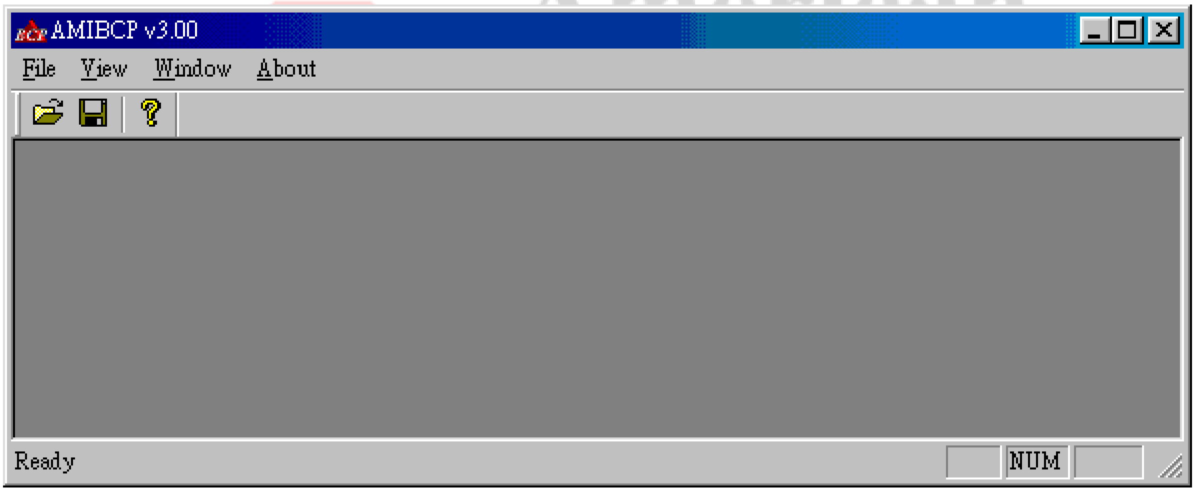
- The file MUST be an AMIBIOS ROM file (Core version 8.xx.xx only)
- BIOS ROM file should be building via "8.00.08_AMITOOLS_17" label or above.

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.*

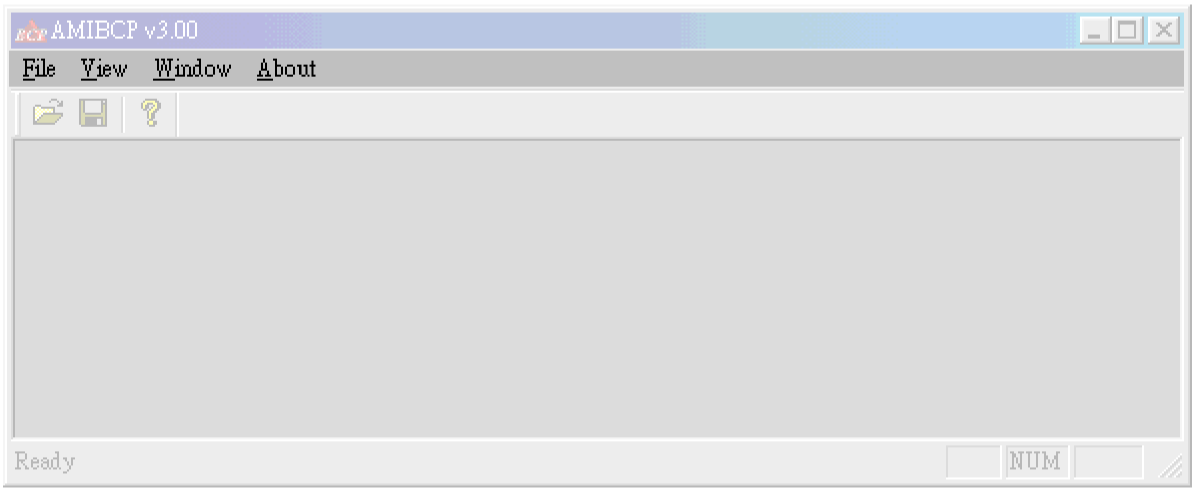
Main Window



Menu Bar

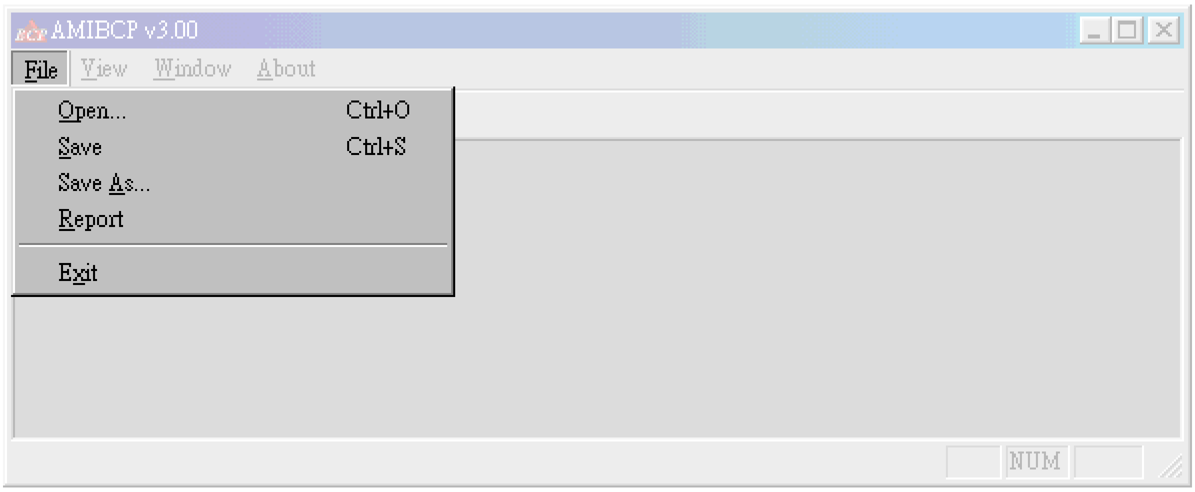
The *Menu bar* is located at the top of the AMIBCP window. The *Menu bar* contains the following:

- 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:

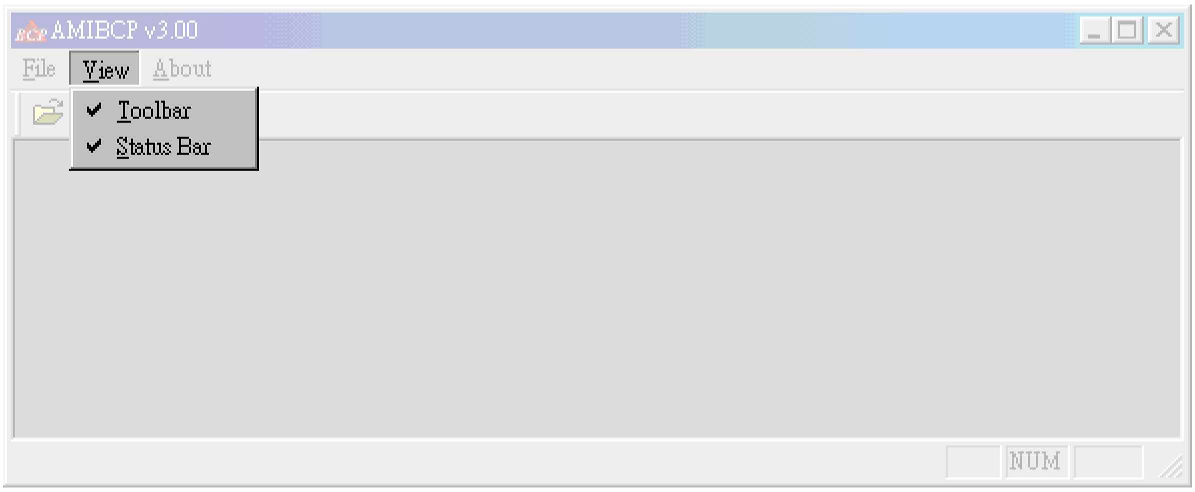


The *File* drop down menu item are explained in the following table:

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

View drop-down menu options

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

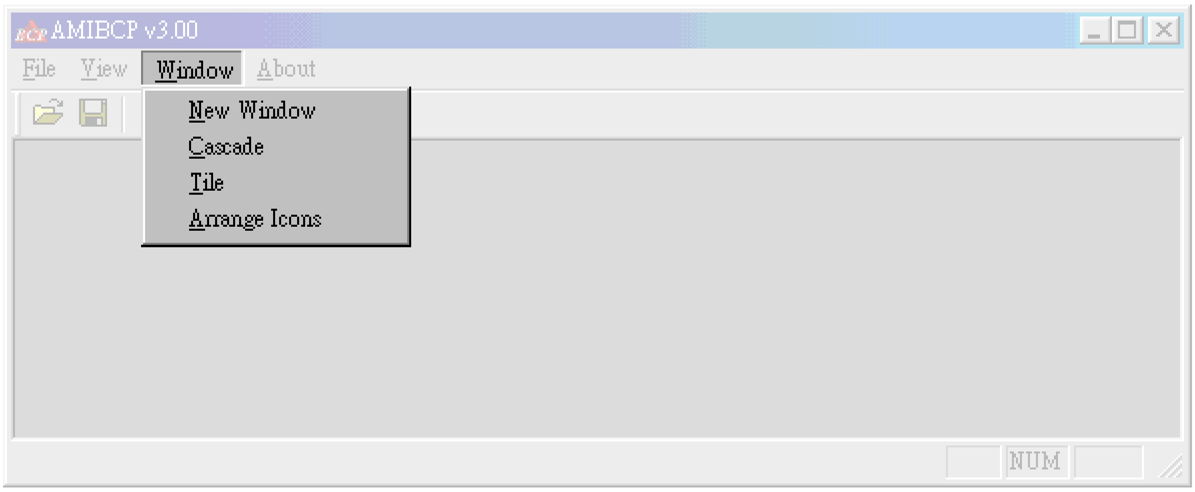


The *View* drop down menu item are explained in the following table:

View Menu Item List	
Name	**Description**
Toolbar	Display or hide the *Toolbar*. The *Toolbar* is displayed under the *Menu bar*.
Status Bar	Display or hide the *Status Bar*. The *Status Bar* is displayed at the bottom of the AMIBCP window.

Window drop-down menu options

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

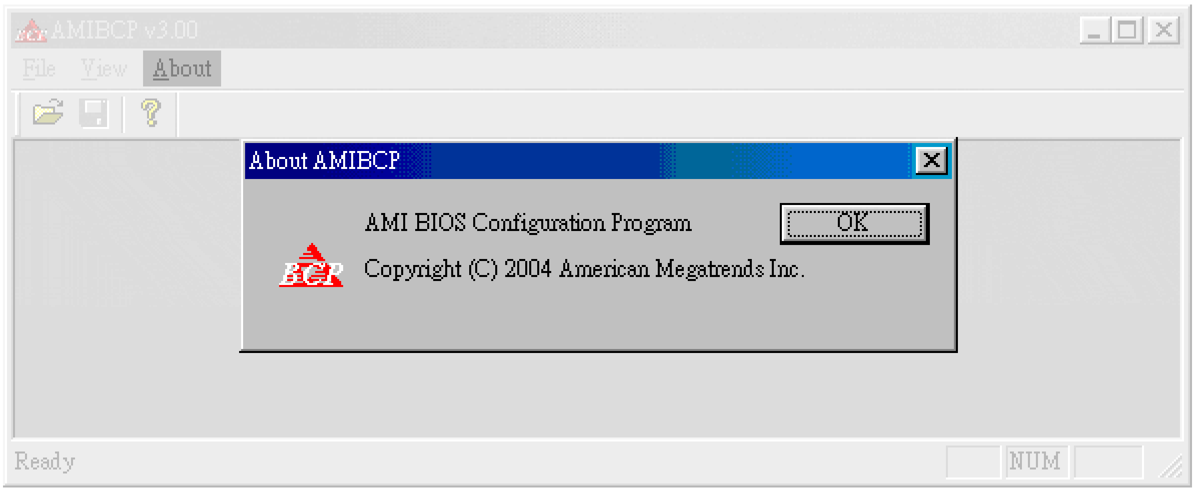


The *Window* drop down menu item are explained in the following table:

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

About

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



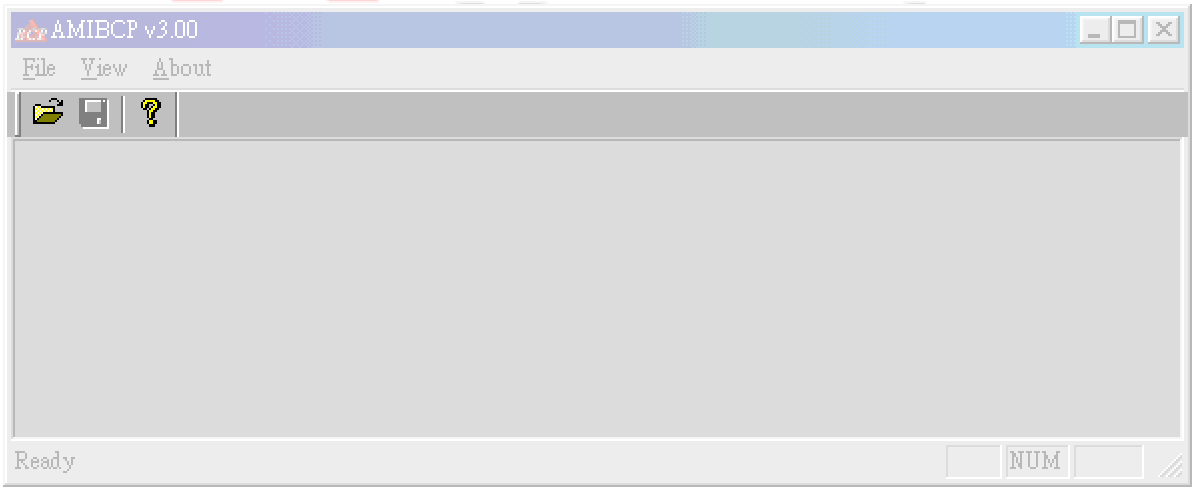
About AMIBCP

AMI BIOS Configuration Program

Copyright (C) 2004 American Megatrends Inc.

Toolbar

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

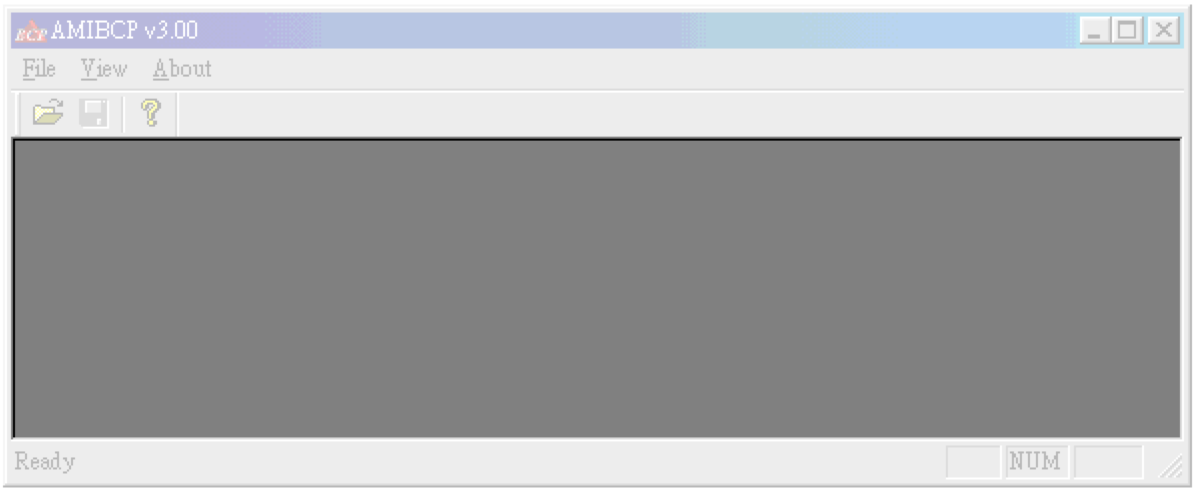


The *Toolbar* icons are explained in the following table:

Toolbar Icon List	
Icon	**Description**
📂	Open an AMIBIOS ROM file.
💾	Save any changes you have made to the AMIBIOS ROM file.
?	Display AMIBCP copyrights information.

Body Frame

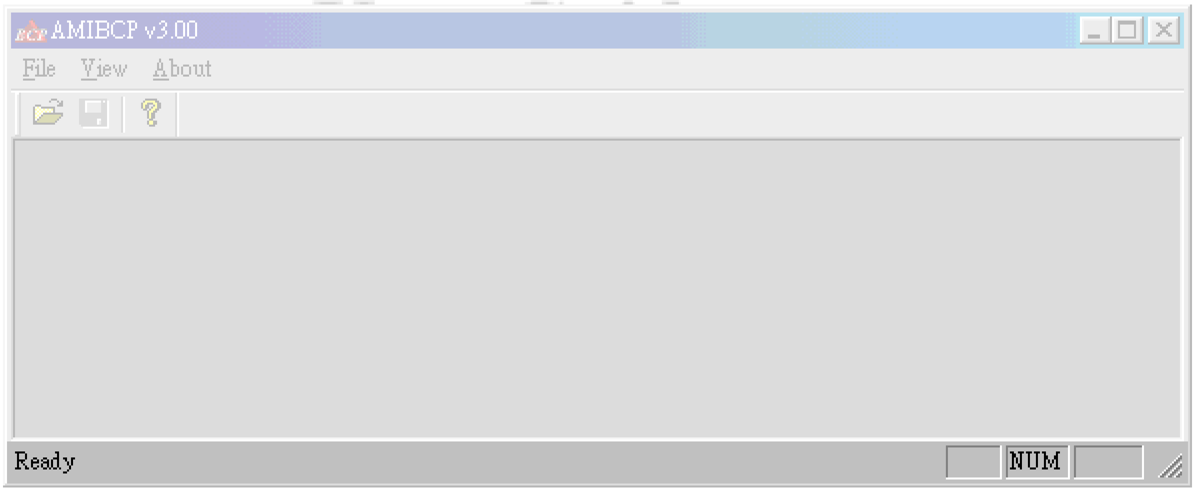
The *Body Frame* is the main frame of AMIBCP. It is located under the *Menu bar* and *Toolbar*. The AMIBCP main functions are displayed in the body frame screen.



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:

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

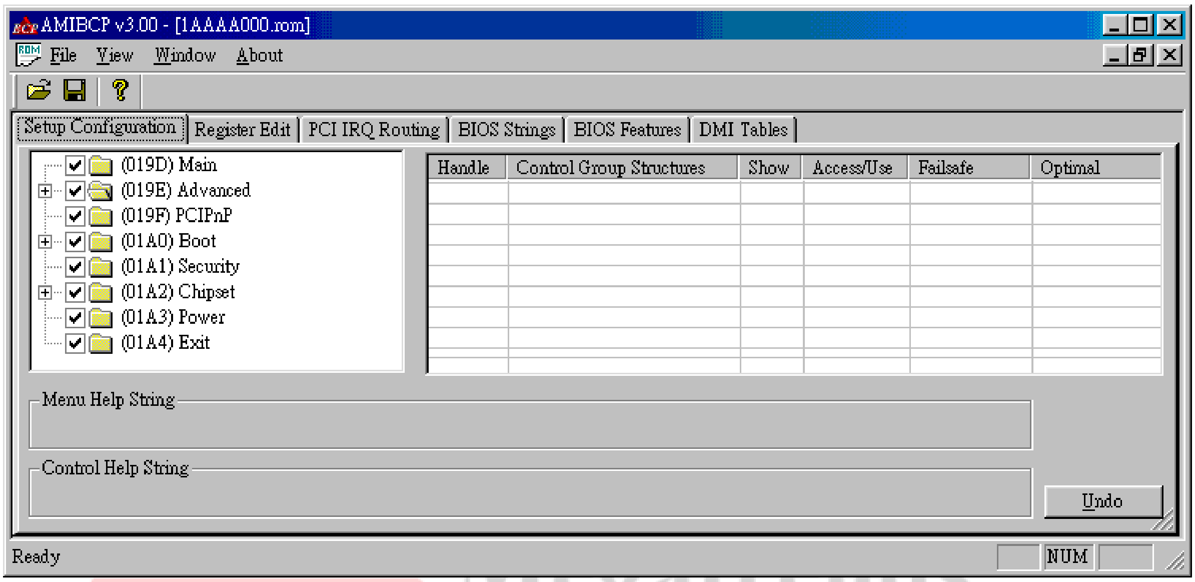


Functions

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

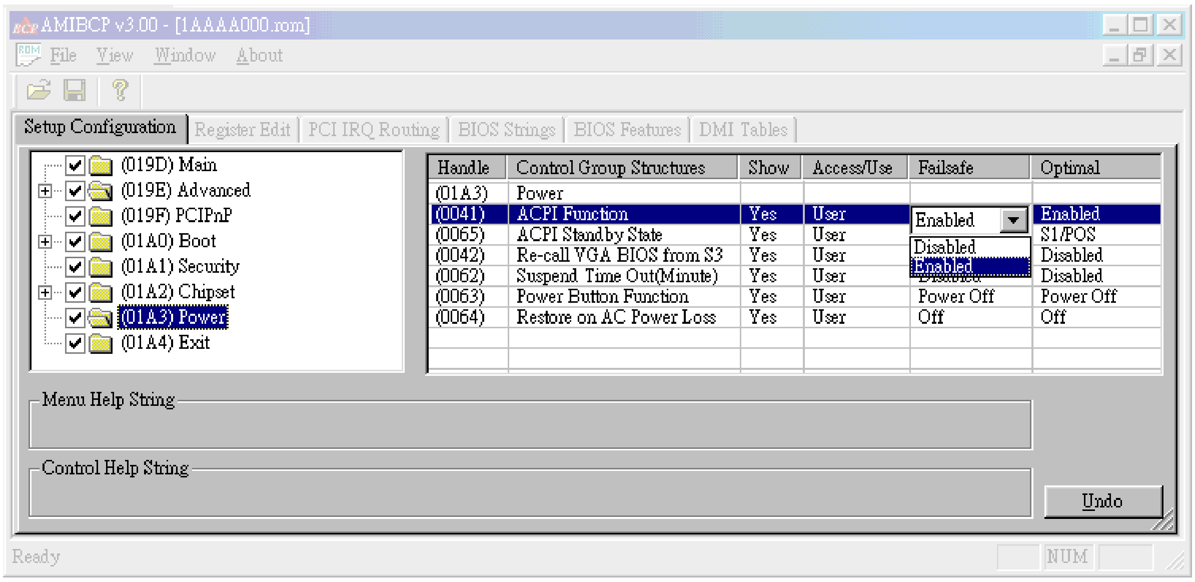


Setup Configuration Tab

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:



☑ (019D) Main
☑ (019E) Advanced
☑ (019F) PCIPnP
☑ (01A0) Boot
☑ (01A1) Security
☑ (01A2) Chipset
☑ (01A3) Power
☑ (01A4) Exit

Handle	Control Group Structures	Show	Access/Use	Failsafe	Optimal
(01A3)	Power				
(0041)	ACPI Function	Yes	User	Enabled Disabled Enabled	Enabled
(0065)	ACPI Standby State	Yes	User		S1/POS
(0042)	Re-call VGA BIOS from S3	Yes	User		Disabled
(0062)	Suspend Time Out(Minute)	Yes	User		Disabled
(0063)	Power Button Function	Yes	User	Power Off	Power Off
(0064)	Restore on AC Power Loss	Yes	User	Off	Off

Fields

The *Setup Configuration* fields are explained in the following table:

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

Buttons

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

Icon	Description
Undo	This button allows you to restore the original setup settings. **Note:** This *Undo* button is used the same way throughout the AMIBCP program.

Menu Help String

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

Control Help String

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

Setup Structures

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

Example of Setup Structures are shown in the following table:

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

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/Optiomal fields, some items does have only list of numbers. These items are implemented using AMIBIOS external functions. Its required executing AMIBIOS code is used to define the list of all possible options for setup items. These fields are filled with numeric identifiers because external functions are not available after booting.

3. To save the changes you have made to the AMIBIOS ROM file, click on the *File* menu bar and select *Save* menu item. You can also click 💾 icon on *Toolbar* to save the changes you have made.

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

Changing SETUP screen layout in Setup Configuration Tab

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

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

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

Icon	Description
👇	This icon indicates the chosen Menu Item will be subordinate to the focus item.
👉	This icon indicates the chosen Menu Item will be having same rank with the focus item.
🛇	This icon indicates the chosen Menu Item may not move to the place where you like.

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

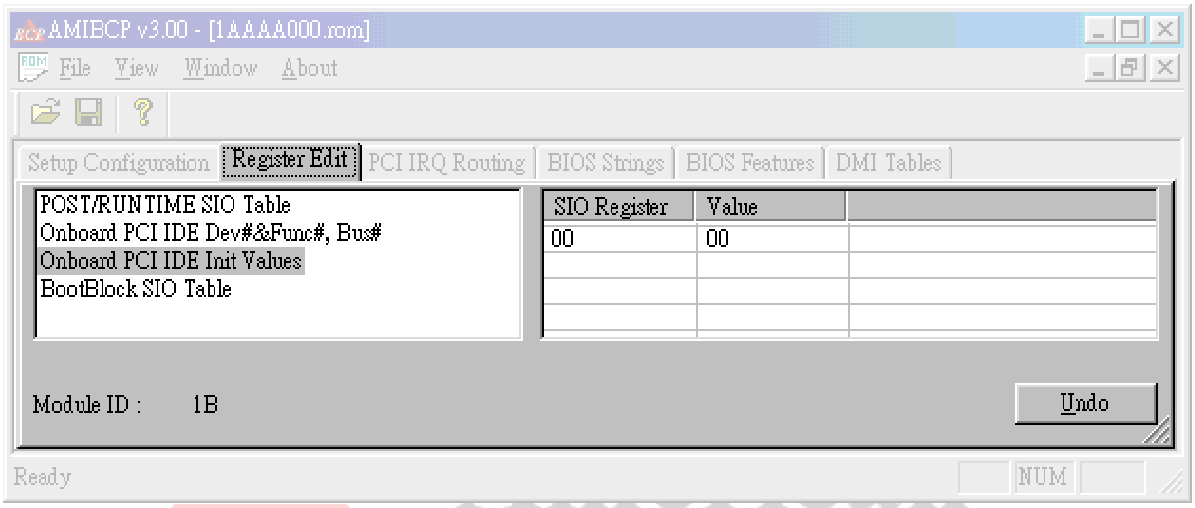
- **Question Item layout between Menus**
  1. Choose a Menu Item on left side of *Setup Configuration tab*.
  2. Choose a Question Item you want to move on right side of *Setup Configuration tab.*
  3. Click right button of mouse to *Cut* the target Question Item.
  4. Choose the Menu Item you like on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

Register Edit Tab

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

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

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



POST/RUNTIME SIO Table
Onboard PCI IDE Dev#&Func#, Bus#
Onboard PCI IDE Init Values
BootBlock SIO Table

SIO Register	Value
00	00

Module ID : 1B

Fields

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

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

Buttons

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

Icon	Description
Undo	This button allows you to restore the original setup settings. **Note:** This *Undo* button is used the same way throughout the AMIBCP program.

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

Using the Register Edit Tab

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

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

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

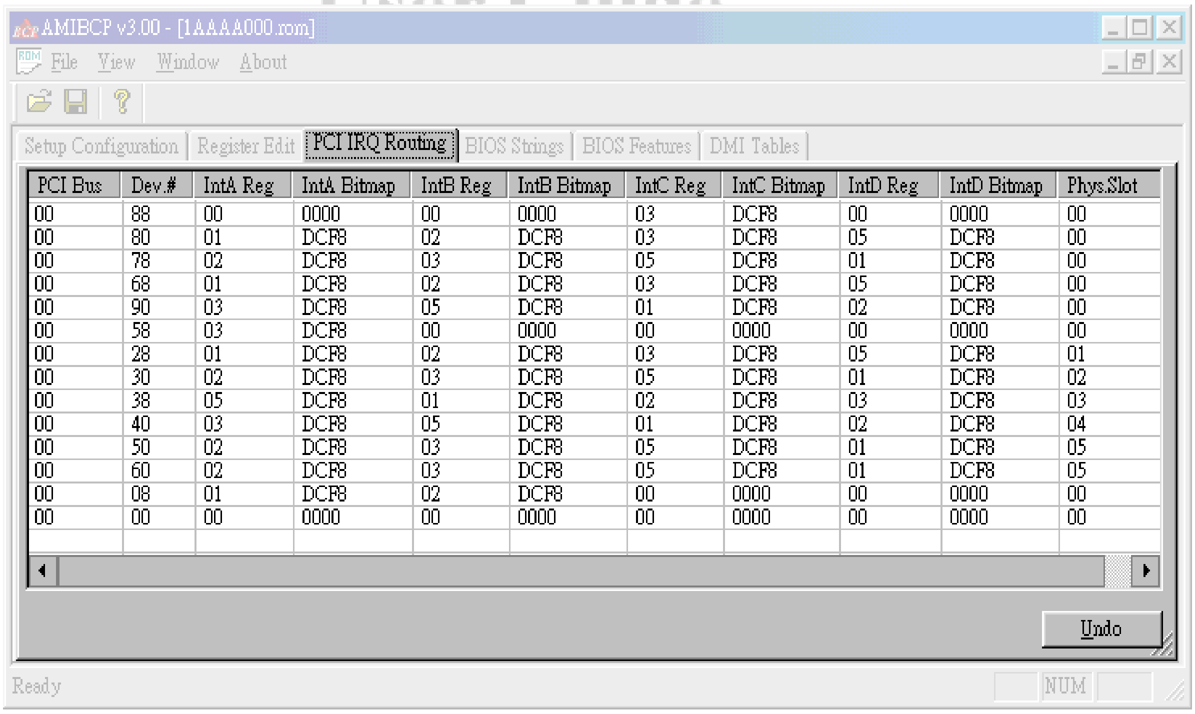
PCI IRQ Routing Tab

The *PCI IRQ Routing* tab allows you to view and modify the *PCI IRQ Routin*g 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:



PCI Bus	Dev.#	IntA Reg	IntA Bitmap	IntB Reg	IntB Bitmap	IntC Reg	IntC Bitmap	IntD Reg	IntD Bitmap	Phys.Slot
00	88	00	0000	00	0000	03	DCF8	00	0000	00
00	80	01	DCF8	02	DCF8	03	DCF8	05	DCF8	00
00	78	02	DCF8	03	DCF8	05	DCF8	01	DCF8	00
00	68	01	DCF8	02	DCF8	03	DCF8	05	DCF8	00
00	90	03	DCF8	05	DCF8	01	DCF8	02	DCF8	00
00	58	03	DCF8	00	0000	00	0000	00	0000	00
00	28	01	DCF8	02	DCF8	03	DCF8	05	DCF8	01
00	30	02	DCF8	03	DCF8	05	DCF8	01	DCF8	02
00	38	05	DCF8	01	DCF8	02	DCF8	03	DCF8	03
00	40	03	DCF8	05	DCF8	01	DCF8	02	DCF8	04
00	50	02	DCF8	03	DCF8	05	DCF8	01	DCF8	05
00	60	02	DCF8	03	DCF8	05	DCF8	01	DCF8	05
00	08	01	DCF8	02	DCF8	00	0000	00	0000	00
00	00	00	0000	00	0000	00	0000	00	0000	00

Fields

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

Field	Description
PCI Bus	This field displays the PCI bus that the device/slot is on. **Note:** Most boards contain a single PCI bus, so this field is usually set to 0.
Dev.#	This field displays the PCI device/slot number. The value of this field is set to a slot or device address on the PCI bus shifted left by three bits (the device number must be in bits 7:3 and bits 2:0 must be 000).
Int A-B-C-D Reg	This field displays the chipset register number that controls the PCI slots (or device) Int A, B, C, and D Pin. The value in this field is basically arbitrary. Slots and devices that share the same chipset interrupt signal must have the same value for this field. **For example:** If Slot 1 Int A pin and Slot 2 Int B pin are both connected to the same chipset interrupt signal, then the chipset register value for Slot 1 Int A must match Slot 2 Int B. If a slot or device has nothing connected to its Int A pin, then this field must be set to 0. If a slot or device has its Int A pin hardwired directly to an IRQ, then this field is set to 0Fxh (where x is 0-F for IRQ 0 - IRQ 15). This is useful if a motherboard has a PCI IDE chip that has its Int A pin hardwired to IRQ 14.
Int A-B-C-D Bitmap	This field displays the IRQ that the chipset is capable of routing to in the slots (or device) Int A, B, C, and D pin. Note: If the value for Chipset Register is set to 0, then all bits in this field are set to 0. **Note:** If the value for chipset register was set to 0Fxh to indicate a hardwired connection to a certain IRQ, then only one bit corresponding to that IRQ must be set in this field.
Phys.Slot	This field displays the slot number of a PCI slot as it appears to the end user. Numbers like 1, 2, 3, and 4 must be used. **Note:** Onboard PCI devices such as PCI IDE chips must have this field set to 0 to indicate that the device is not a removable PCI adapter card.

Buttons

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

Icon	Description
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 PCI IRQ Routing Tab

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

1. To modify any field in the *PCI IRQ Routing* table, simply double-click on the any field you want to modify and type in new value.

   Note: You can add a new PCI entry into the table if all fields in the entry are set to zeros.

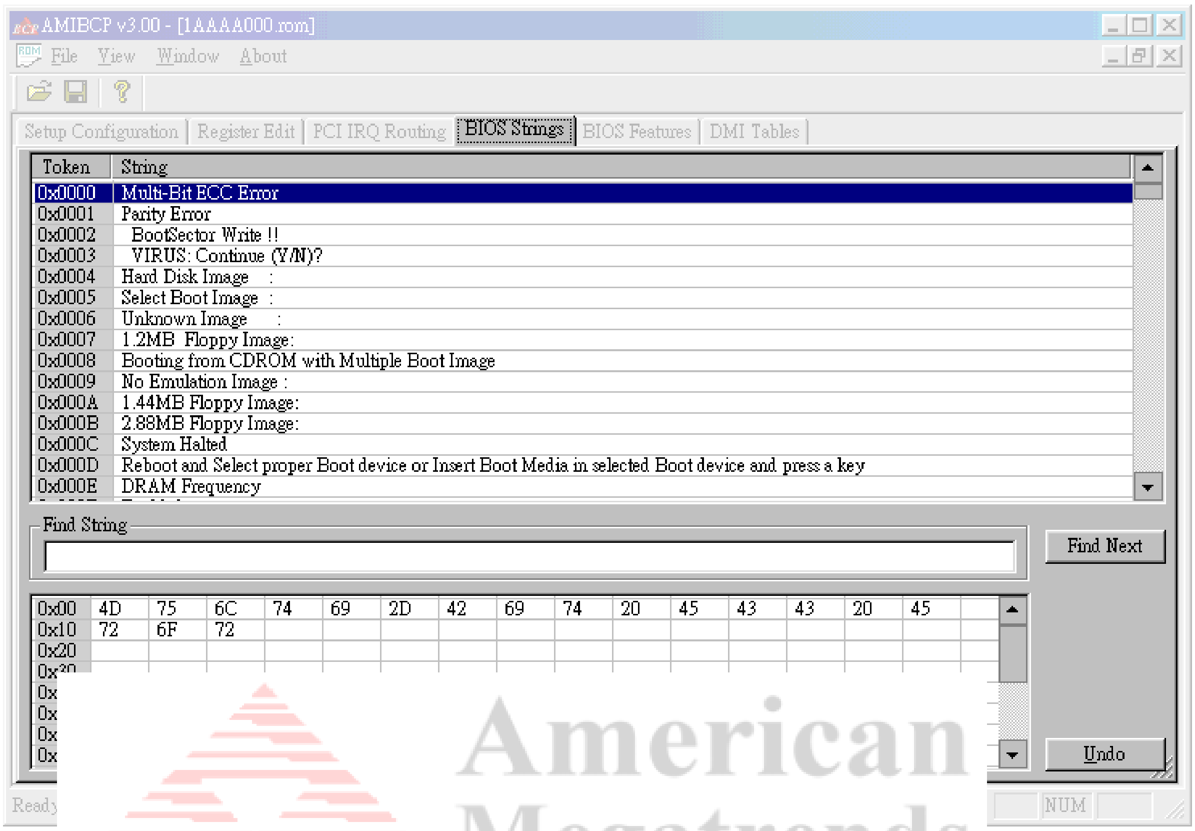
2. To save the changes you have made to the AMIBIOS ROM file, click on the *File* menu bar and select *Save* menu item. You can also click 💾 icon on *Toolbar* to save the changes you have made.

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

BIOS String Tab

The *BIOS Strings* tab allows you to view and edit AMIBIOS strings.

An example of the *BIOS Strings* tab is shown below:



Token	String
0x0000	Multi-Bit ECC Error
0x0001	Parity Error
0x0002	BootSector Write !!
0x0003	VIRUS: Continue (Y/N)?
0x0004	Hard Disk Image :
0x0005	Select Boot Image :
0x0006	Unknown Image :
0x0007	1.2MB Floppy Image:
0x0008	Booting from CDROM with Multiple Boot Image
0x0009	No Emulation Image :
0x000A	1.44MB Floppy Image:
0x000B	2.88MB Floppy Image:
0x000C	System Halted
0x000D	Reboot and Select proper Boot device or Insert Boot Media in selected Boot device and press a key
0x000E	DRAM Frequency

```
0x00  4D  75  6C  74  69  2D  42  69  74  20  45  43  43  20  45
0x10  72  6F  72
0x20
```

Fields

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

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

Buttons

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

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

Using the BIOS String Tab

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

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

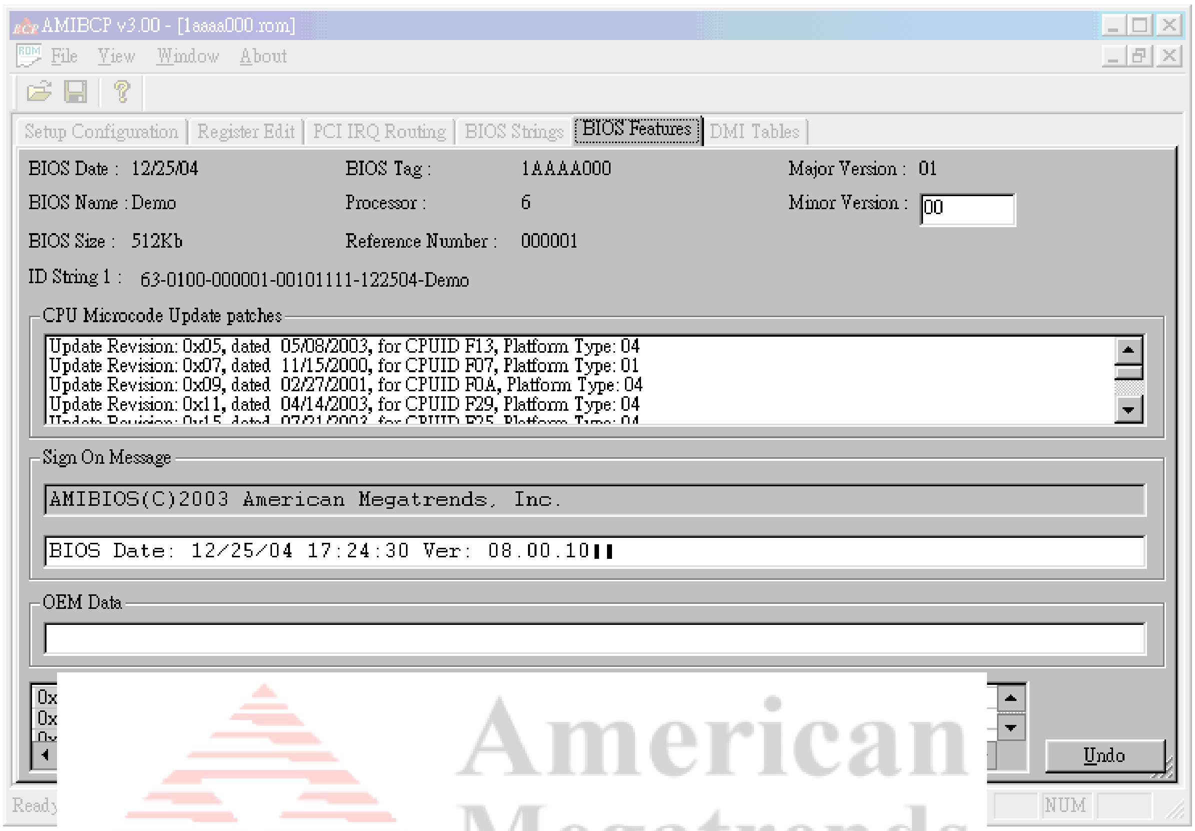
   Note: You can lick on the *Undo* button to restore the original setup settings before saving any new changes.

BIOS 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 Patchs
- Sign On Message
- OEM Data(If available)

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



BIOS Date : 12/25/04
BIOS Tag : 1AAAA000
Major Version : 01
BIOS Name : Demo
Processor : 6
Minor Version : 00
BIOS Size : 512Kb
Reference Number : 000001
ID String 1 : 63-0100-000001-00101111-122504-Demo

CPU Microcode Update patches

```
Update Revision: 0x05, dated 05/08/2003, for CPUID F13, Platform Type: 04
Update Revision: 0x07, dated 11/15/2000, for CPUID F07, Platform Type: 01
Update Revision: 0x09, dated 02/27/2001, for CPUID F0A, Platform Type: 04
Update Revision: 0x11, dated 04/14/2003, for CPUID F29, Platform Type: 04
```

Sign On Message

```
AMIBIOS(C)2003 American Megatrends, Inc.
BIOS Date: 12/25/04 17:24:30 Ver: 08.00.10
```

OEM Data

Fields

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

Field	Description
BIOS Date	This field displays the date when the AMIBIOS ROM file was built. The value in this field cannot be changed.
BIOS Name	This field displays the name associated with the AMIBIOS ROM file. The value in this field cannot be changed.
Processor	This field displays the number that is used to define the processor type. The value in this field cannot be changed.
Major Version	This field displays the main AMIBIOS revision number that is used with the AMIBIOS release. The value in this field cannot be changed.
ID String 1	This field displays the ID string that is associated with the AMIBIOS ROM file. The value in this field cannot be changed.
BIOS Size	This field displays the actual size of the AMIBIOS ROM file. The value in this field cannot be changed.
BIOS Tag	This field displays the eight-character tag that is associated with the AMIBIOS ROM file. The value in this field cannot be changed.
Minor Version	This field displays the minor AMIBIOS revision number that is used with the AMIBIOS release. This field is editable.

Field	Description
CPU Microcode Update Patches	This field displays processor patches that are contained in the AMIBIOS ROM file. The values in this field cannot be changed.
Sign On Message	This field displays the AMIBIOS sign-on message that is displayed during POST. The AMIBIOS copyright string is not editable. **Note:** This field must not be more than 175 characters in length.
OEM Data	This field displays the OEM data that is provided by the AMIBIOS. This field is 62-bytes long and displayed in two modes (text and hexadecimal). This field is editable. **Note:** The OEM data area is supported on the AMIBIOS 8.00.08 core and later releases.

Buttons

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

Icon	Description
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:

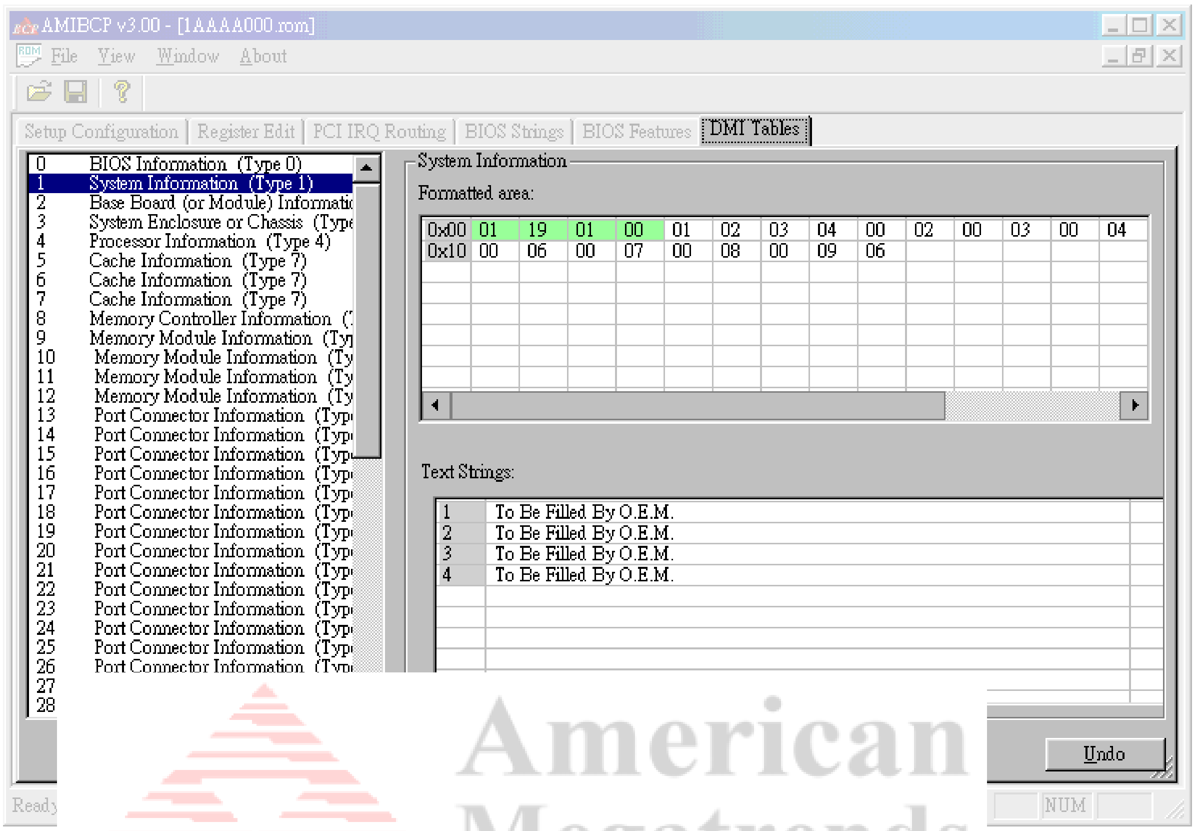
1. To modify any editable field, click on it and type in the new value.
2. To save the changes you have made to the AMIBIOS ROM file, click on the *File* menu bar and select *Save* menu item. You can also click 💾 icon on *Toolbar* to save the changes you have made.

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

DMI Tables Tab

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

An example of the *DMI Tables* tab is shown below:



AMIBCP v3.00 - [1AAAA000.rom]
File View Window About
Setup Configuration | Register Edit | PCI IRQ Routing | BIOS Strings | BIOS Features | DMI Tables
Undo
Ready
NUM

```
0   BIOS Information  (Type 0)
1   System Information  (Type 1)
2   Base Board (or Module) Informati
3   System Enclosure or Chassis  (Type
4   Processor Information  (Type 4)
5   Cache Information  (Type 7)
6   Cache Information  (Type 7)
7   Cache Information  (Type 7)
8   Memory Controller Information  (
9   Memory Module Information  (Ty
10  Memory Module Information  (Ty
11  Memory Module Information  (Ty
12  Memory Module Information  (Ty
13  Port Connector Information  (Typ
14  Port Connector Information  (Typ
15  Port Connector Information  (Typ
16  Port Connector Information  (Typ
17  Port Connector Information  (Typ
18  Port Connector Information  (Typ
19  Port Connector Information  (Typ
20  Port Connector Information  (Typ
21  Port Connector Information  (Typ
22  Port Connector Information  (Typ
23  Port Connector Information  (Typ
24  Port Connector Information  (Typ
25  Port Connector Information  (Typ
26  Port Connector Information  (Typ
27
28
```

System Information

Formatted area:

```
0x00  01  19  01  00  01  02  03  04  00  02  00  03  00  04
0x10  00  06  00  07  00  08  00  09  06
```

Text Strings:

```
1  To Be Filled By O.E.M.
2  To Be Filled By O.E.M.
3  To Be Filled By O.E.M.
4  To Be Filled By O.E.M.
```

Fields

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

Field	Description
Formatted area	This field displays the DMI Tables values (in hex). You can modify all DMI table values except the first two bytes. Note: The first two bytes of the DMI Tables are used to define the table type and size.
Test Strings	This field displays the DMI Tables strings. You can modify these strings but you cannot change the number of strings.

Buttons

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

Icon	Description
<< Back	This button allows you to go to the previous DMI table in the table list.
Next >>	This button allows you to go to the next DMI table in the table list.
Undo	This button allows you to restore the original setup settings. **Note:** This *Undo* button is used the same way throughout the AMIBCP program.

Using the DMI Tables Tab

You can use the *DMI Tables* tab by following steps:

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

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

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
- Microsoft® Windows® Vista 32/64

BIOS Requirements

System BIOS should have the followings:

- AMIBIOS CORE version 8.xx.xx.
- ***SMIFlash*** *eModule* with “8.00.00_SMIFlash-1.00.10” label or later.
- ***SMBIOS*** *eModule* with “8.00.08_SMB-3.1.02RC20” 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.
• ***DMI16.EXE***	Driver for Microsoft® Windows® 98/ME/NT/2000/XP

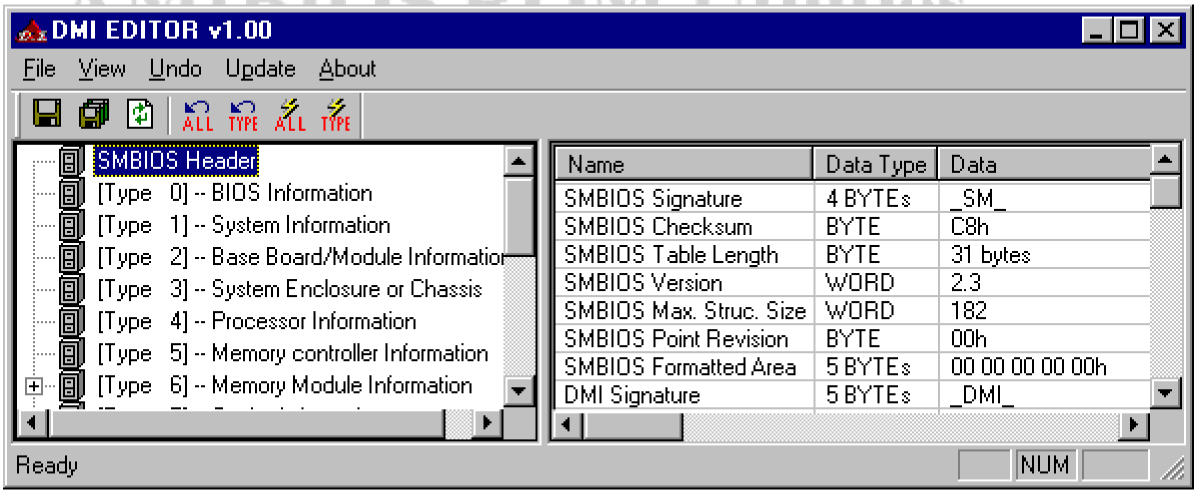
Notice: DMI16.EXE is used for BIOS which supports PNP function only instead of SMI protocol. If BIOS supports SMI protocol for updating SMBIOS data then DMI16.EXE is not needed when using DMIEditor.

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



SMBIOS Header
[Type 0] -- BIOS Information
[Type 1] -- System Information
[Type 2] -- Base Board/Module Informatio
[Type 3] -- System Enclosure or Chassis
[Type 4] -- Processor Information
[Type 5] -- Memory controller Information
[Type 6] -- Memory Module Information

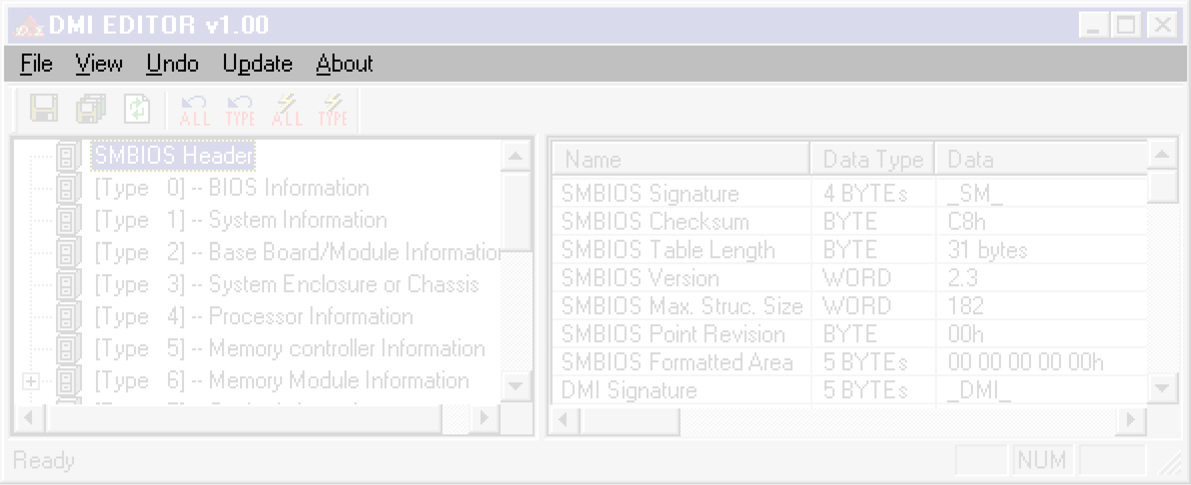
Name	Data Type	Data
SMBIOS Signature	4 BYTEs	_SM_
SMBIOS Checksum	BYTE	C8h
SMBIOS Table Length	BYTE	31 bytes
SMBIOS Version	WORD	2.3
SMBIOS Max. Struc. Size	WORD	182
SMBIOS Point Revision	BYTE	00h
SMBIOS Formatted Area	5 BYTEs	00 00 00 00 00h
DMI Signature	5 BYTEs	_DMI_

Ready NUM

Menu Bar

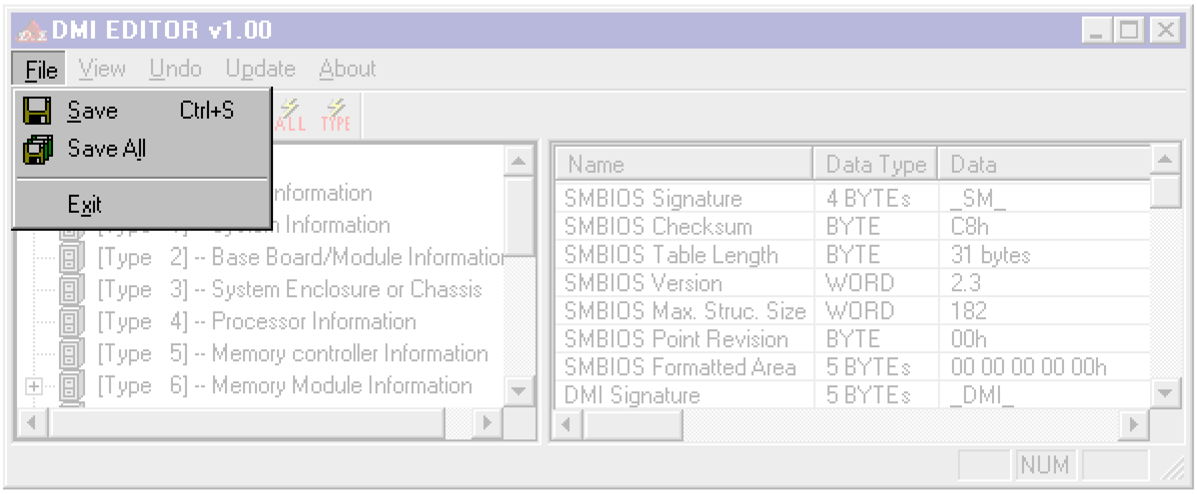
The *Menu bar* is located at the top of the DMI Editor window. The *Menu bar* contains the following:

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


DMI EDITOR v1.00
File View Undo Update About
ALL TYPE ALL TYPE
SMBIOS Header
[Type 0] -- BIOS Information
[Type 1] -- System Information
[Type 2] -- Base Board/Module Informatio
[Type 3] -- System Enclosure or Chassis
[Type 4] -- Processor Information
[Type 5] -- Memory controller Information
[Type 6] -- Memory Module Information
Ready
NUM

Name	Data Type	Data
SMBIOS Signature	4 BYTEs	_SM_
SMBIOS Checksum	BYTE	C8h
SMBIOS Table Length	BYTE	31 bytes
SMBIOS Version	WORD	2.3
SMBIOS Max. Struc. Size	WORD	182
SMBIOS Point Revision	BYTE	00h
SMBIOS Formatted Area	5 BYTEs	00 00 00 00 00h
DMI Signature	5 BYTEs	_DMI_

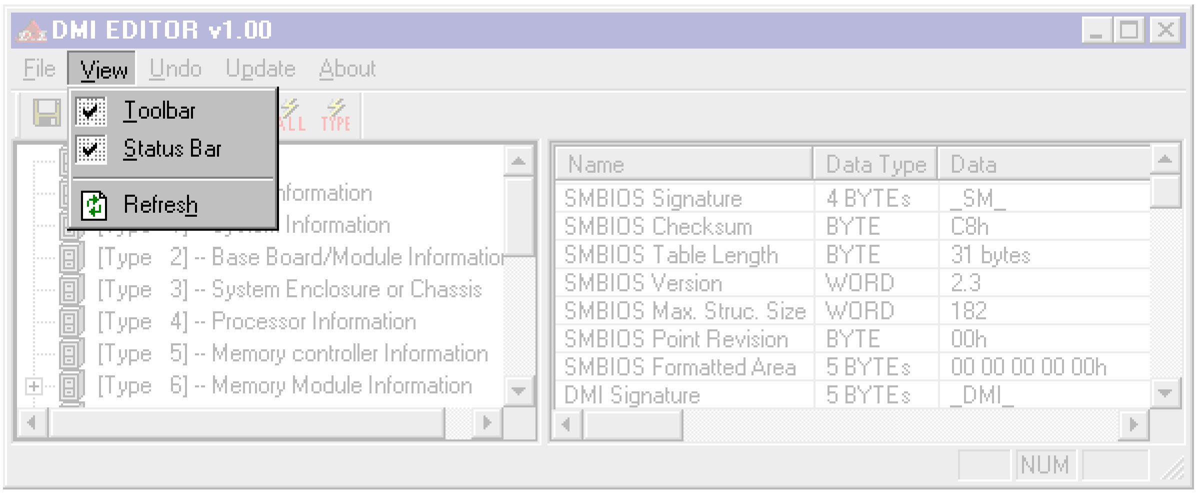
File drop-down menu options


DMI EDITOR v1.00
File View Undo Update About
Save Ctrl+S
Save All
Exit
ALL TYPE
nformation
n Information
[Type 2] -- Base Board/Module Informatio
[Type 3] -- System Enclosure or Chassis
[Type 4] -- Processor Information
[Type 5] -- Memory controller Information
[Type 6] -- Memory Module Information
NUM

Name	Data Type	Data
SMBIOS Signature	4 BYTEs	_SM_
SMBIOS Checksum	BYTE	C8h
SMBIOS Table Length	BYTE	31 bytes
SMBIOS Version	WORD	2.3
SMBIOS Max. Struc. Size	WORD	182
SMBIOS Point Revision	BYTE	00h
SMBIOS Formatted Area	5 BYTEs	00 00 00 00 00h
DMI Signature	5 BYTEs	_DMI_

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

View drop-down menu options

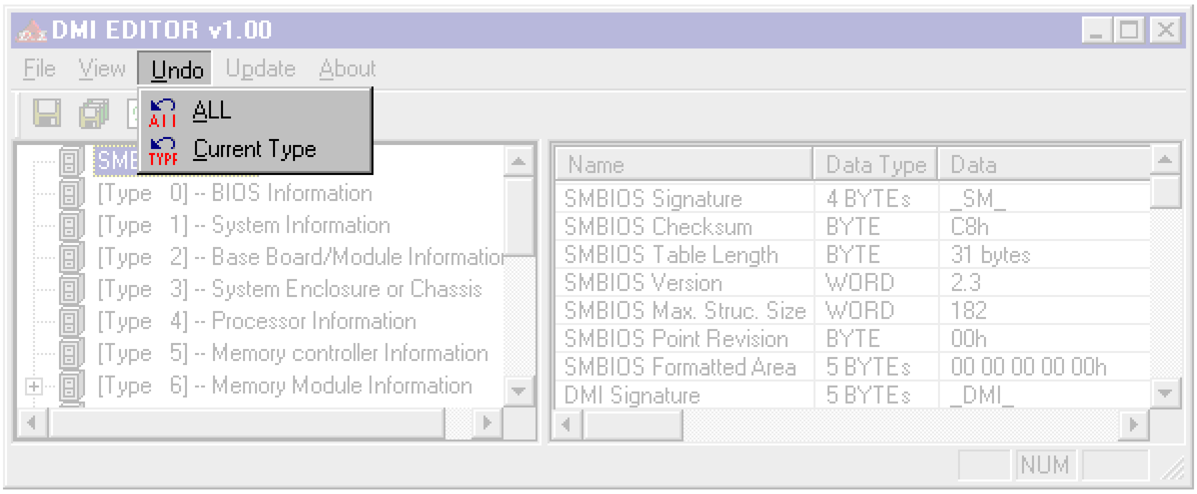


nformation
Information
[Type 2] -- Base Board/Module Informatio
[Type 3] -- System Enclosure or Chassis
[Type 4] -- Processor Information
[Type 5] -- Memory controller Information
[Type 6] -- Memory Module Information

Name	Data Type	Data
SMBIOS Signature	4 BYTEs	_SM_
SMBIOS Checksum	BYTE	C8h
SMBIOS Table Length	BYTE	31 bytes
SMBIOS Version	WORD	2.3
SMBIOS Max. Struc. Size	WORD	182
SMBIOS Point Revision	BYTE	00h
SMBIOS Formatted Area	5 BYTEs	00 00 00 00 00h
DMI Signature	5 BYTEs	_DMI_

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

Undo drop-down menu options

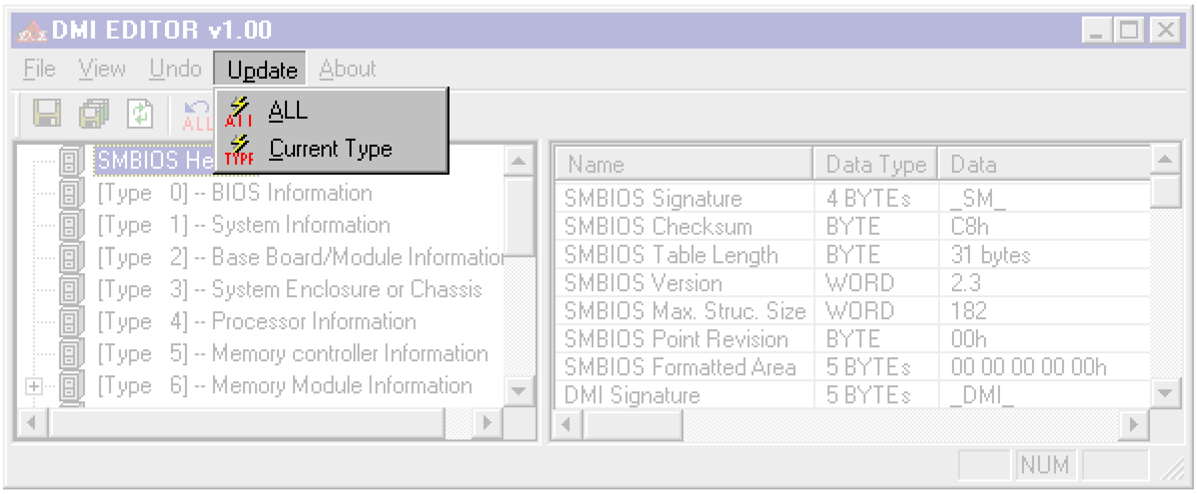


SMB
[Type 0] -- BIOS Information
[Type 1] -- System Information
[Type 2] -- Base Board/Module Informatio
[Type 3] -- System Enclosure or Chassis
[Type 4] -- Processor Information
[Type 5] -- Memory controller Information
[Type 6] -- Memory Module Information

Name	Data Type	Data
SMBIOS Signature	4 BYTEs	_SM_
SMBIOS Checksum	BYTE	C8h
SMBIOS Table Length	BYTE	31 bytes
SMBIOS Version	WORD	2.3
SMBIOS Max. Struc. Size	WORD	182
SMBIOS Point Revision	BYTE	00h
SMBIOS Formatted Area	5 BYTEs	00 00 00 00 00h
DMI Signature	5 BYTEs	_DMI_

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

Update drop-down menu options



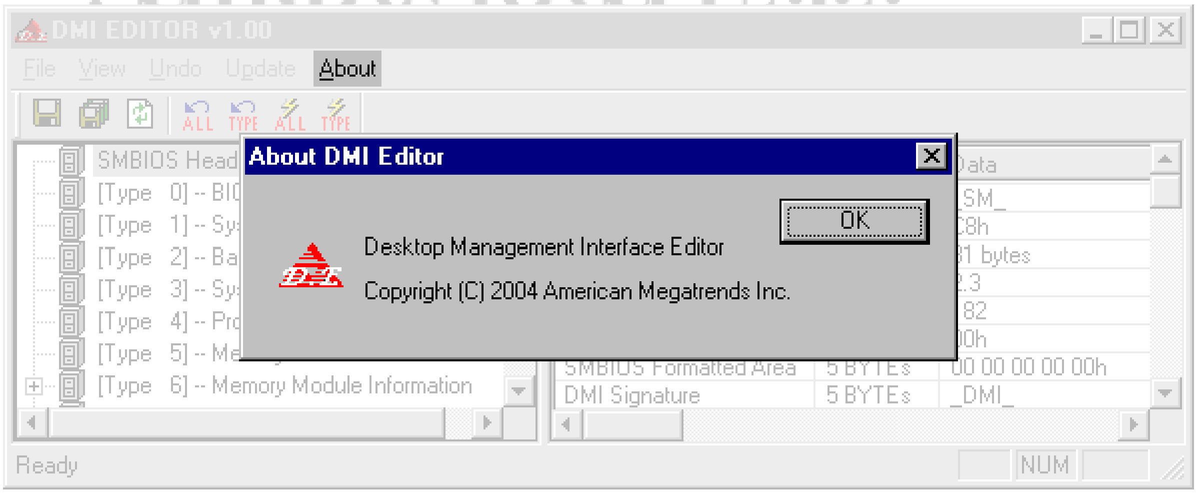
SMBIOS He
[Type 0] -- BIOS Information
[Type 1] -- System Information
[Type 2] -- Base Board/Module Informatio
[Type 3] -- System Enclosure or Chassis
[Type 4] -- Processor Information
[Type 5] -- Memory controller Information
[Type 6] -- Memory Module Information

Name	Data Type	Data
SMBIOS Signature	4 BYTEs	_SM_
SMBIOS Checksum	BYTE	C8h
SMBIOS Table Length	BYTE	31 bytes
SMBIOS Version	WORD	2.3
SMBIOS Max. Struc. Size	WORD	182
SMBIOS Point Revision	BYTE	00h
SMBIOS Formatted Area	5 BYTEs	00 00 00 00 00h
DMI Signature	5 BYTEs	_DMI_

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

About

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

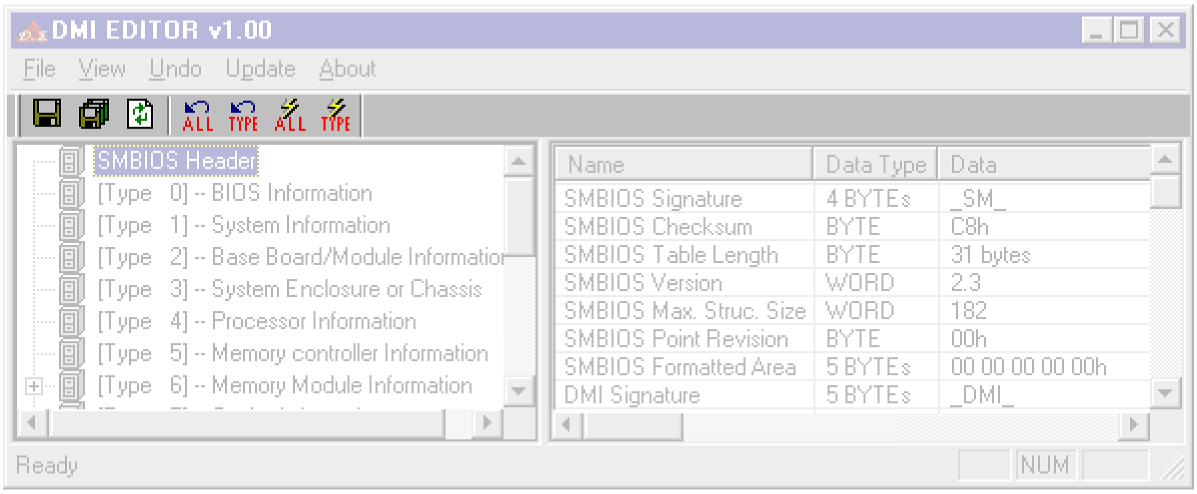


Desktop Management Interface Editor

Copyright (C) 2004 American Megatrends Inc.

Toolbar

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



SMBIOS Header
[Type 0] -- BIOS Information
[Type 1] -- System Information
[Type 2] -- Base Board/Module Informatio
[Type 3] -- System Enclosure or Chassis
[Type 4] -- Processor Information
[Type 5] -- Memory controller Information
[Type 6] -- Memory Module Information

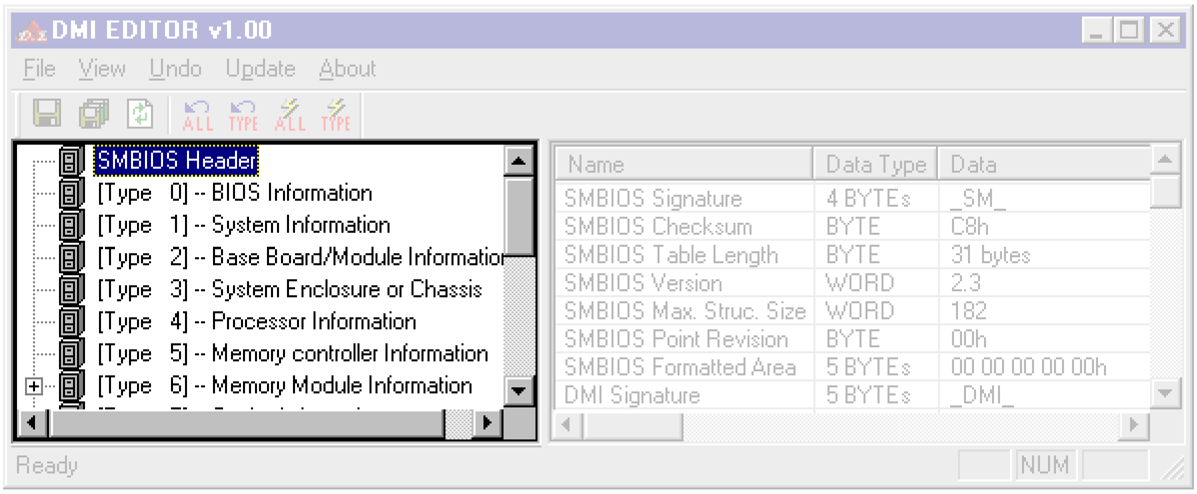
Name	Data Type	Data
SMBIOS Signature	4 BYTEs	_SM_
SMBIOS Checksum	BYTE	C8h
SMBIOS Table Length	BYTE	31 bytes
SMBIOS Version	WORD	2.3
SMBIOS Max. Struc. Size	WORD	182
SMBIOS Point Revision	BYTE	00h
SMBIOS Formatted Area	5 BYTEs	00 00 00 00 00h
DMI Signature	5 BYTEs	_DMI_

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

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

Type Frame

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

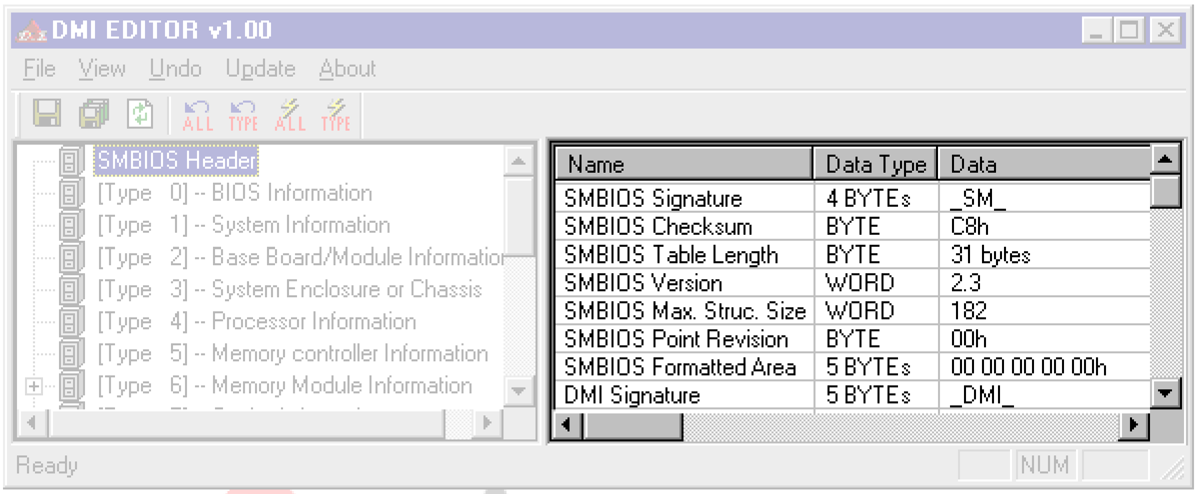


SMBIOS Header
[Type 0] -- BIOS Information
[Type 1] -- System Information
[Type 2] -- Base Board/Module Informatio
[Type 3] -- System Enclosure or Chassis
[Type 4] -- Processor Information
[Type 5] -- Memory controller Information
[Type 6] -- Memory Module Information

Name	Data Type	Data
SMBIOS Signature	4 BYTEs	_SM_
SMBIOS Checksum	BYTE	C8h
SMBIOS Table Length	BYTE	31 bytes
SMBIOS Version	WORD	2.3
SMBIOS Max. Struc. Size	WORD	182
SMBIOS Point Revision	BYTE	00h
SMBIOS Formatted Area	5 BYTEs	00 00 00 00 00h
DMI Signature	5 BYTEs	_DMI_

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.

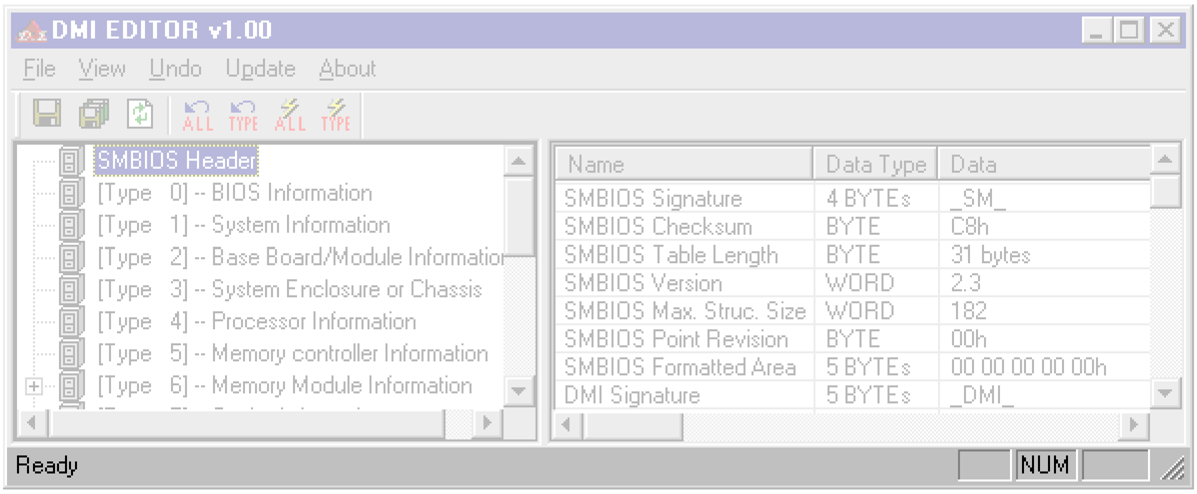

DMI EDITOR v1.00
File View Undo Update About
ALL TYPE ALL TYPE
SMBIOS Header
[Type 0] -- BIOS Information
[Type 1] -- System Information
[Type 2] -- Base Board/Module Informatio
[Type 3] -- System Enclosure or Chassis
[Type 4] -- Processor Information
[Type 5] -- Memory controller Information
[Type 6] -- Memory Module Information
Ready
NUM

Name	Data Type	Data
SMBIOS Signature	4 BYTEs	_SM_
SMBIOS Checksum	BYTE	C8h
SMBIOS Table Length	BYTE	31 bytes
SMBIOS Version	WORD	2.3
SMBIOS Max. Struc. Size	WORD	182
SMBIOS Point Revision	BYTE	00h
SMBIOS Formatted Area	5 BYTEs	00 00 00 00 00h
DMI Signature	5 BYTEs	_DMI_

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:

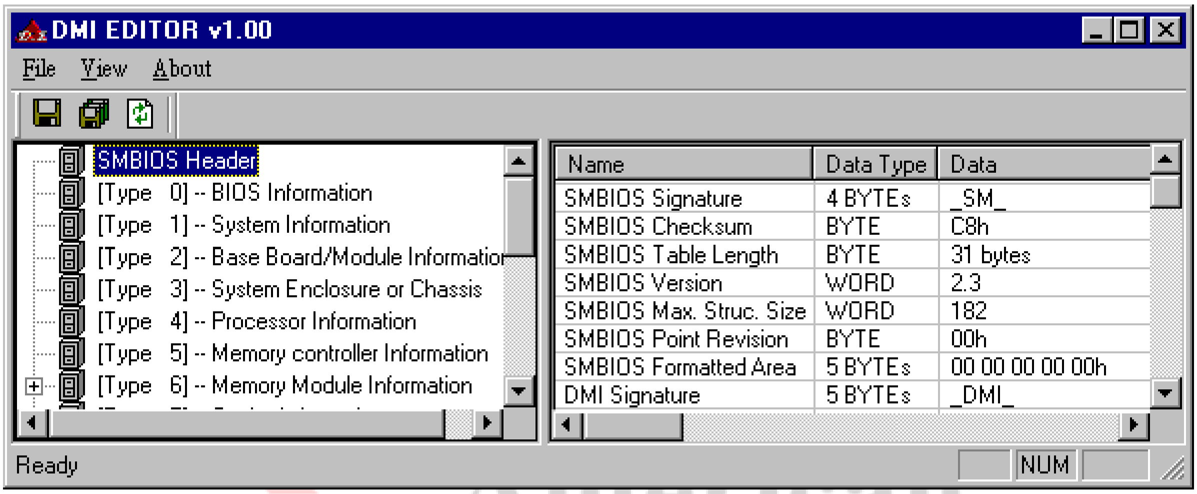
Item	Description
CAP	The Caps Lock key is latched down.
NUM	The Num Lock key is latched down.
SCRL	The Scroll Lock key is latched down.


DMI EDITOR v1.00
File View Undo Update About
ALL TYPE ALL TYPE
SMBIOS Header
[Type 0] -- BIOS Information
[Type 1] -- System Information
[Type 2] -- Base Board/Module Informatio
[Type 3] -- System Enclosure or Chassis
[Type 4] -- Processor Information
[Type 5] -- Memory controller Information
[Type 6] -- Memory Module Information
Ready
NUM

Name	Data Type	Data
SMBIOS Signature	4 BYTEs	_SM_
SMBIOS Checksum	BYTE	C8h
SMBIOS Table Length	BYTE	31 bytes
SMBIOS Version	WORD	2.3
SMBIOS Max. Struc. Size	WORD	182
SMBIOS Point Revision	BYTE	00h
SMBIOS Formatted Area	5 BYTEs	00 00 00 00 00h
DMI Signature	5 BYTEs	_DMI_

Functions

To use DMIEDIT, you can double-click the executable file icon to open Main Window. For non-AMIBIOS system, **Undo** and **Update** menu will be hidden as below:



SMBIOS Header
[Type 0] -- BIOS Information
[Type 1] -- System Information
[Type 2] -- Base Board/Module Informatio
[Type 3] -- System Enclosure or Chassis
[Type 4] -- Processor Information
[Type 5] -- Memory controller Information
[Type 6] -- Memory Module Information

Name	Data Type	Data
SMBIOS Signature	4 BYTEs	_SM_
SMBIOS Checksum	BYTE	C8h
SMBIOS Table Length	BYTE	31 bytes
SMBIOS Version	WORD	2.3
SMBIOS Max. Struc. Size	WORD	182
SMBIOS Point Revision	BYTE	00h
SMBIOS Formatted Area	5 BYTEs	00 00 00 00 00h
DMI Signature	5 BYTEs	_DMI_

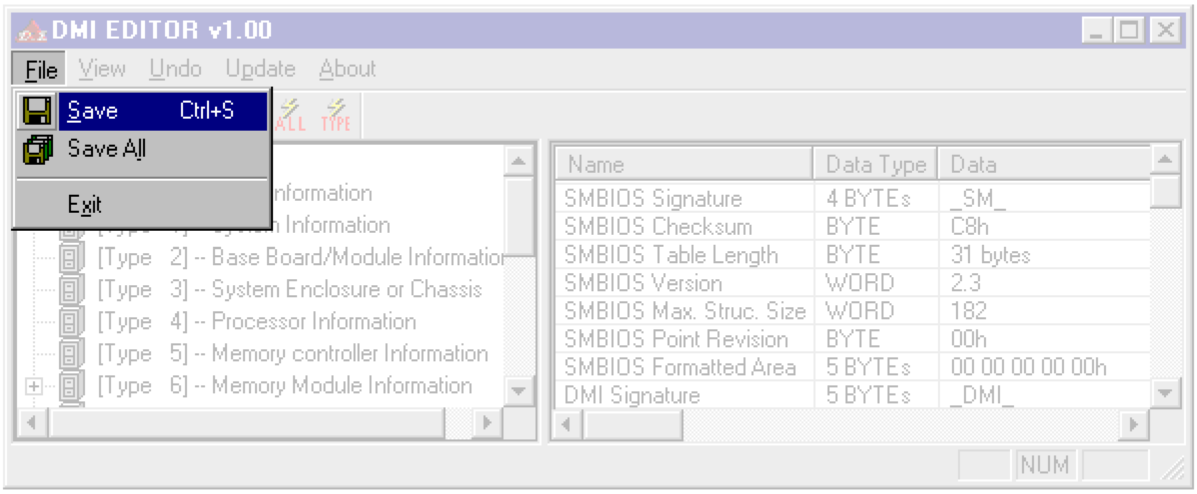
Ready NUM

Browsing SMBIOS information

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

Saving SMBIOS information to file

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



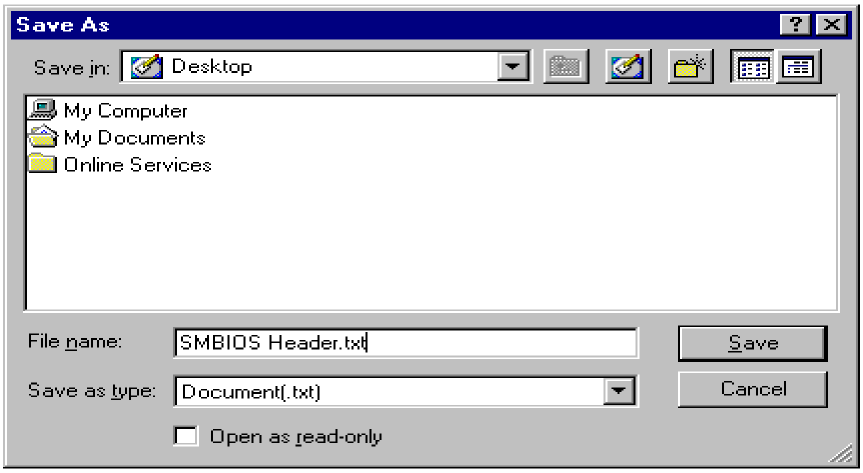
...nformation
...n Information
[Type 2] -- Base Board/Module Informatio
[Type 3] -- System Enclosure or Chassis
[Type 4] -- Processor Information
[Type 5] -- Memory controller Information
[Type 6] -- Memory Module Information

Name	Data Type	Data
SMBIOS Signature	4 BYTEs	_SM_
SMBIOS Checksum	BYTE	C8h
SMBIOS Table Length	BYTE	31 bytes
SMBIOS Version	WORD	2.3
SMBIOS Max. Struc. Size	WORD	182
SMBIOS Point Revision	BYTE	00h
SMBIOS Formatted Area	5 BYTEs	00 00 00 00 00h
DMI Signature	5 BYTEs	_DMI_

NUM

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

press Cancel to ignore the function.



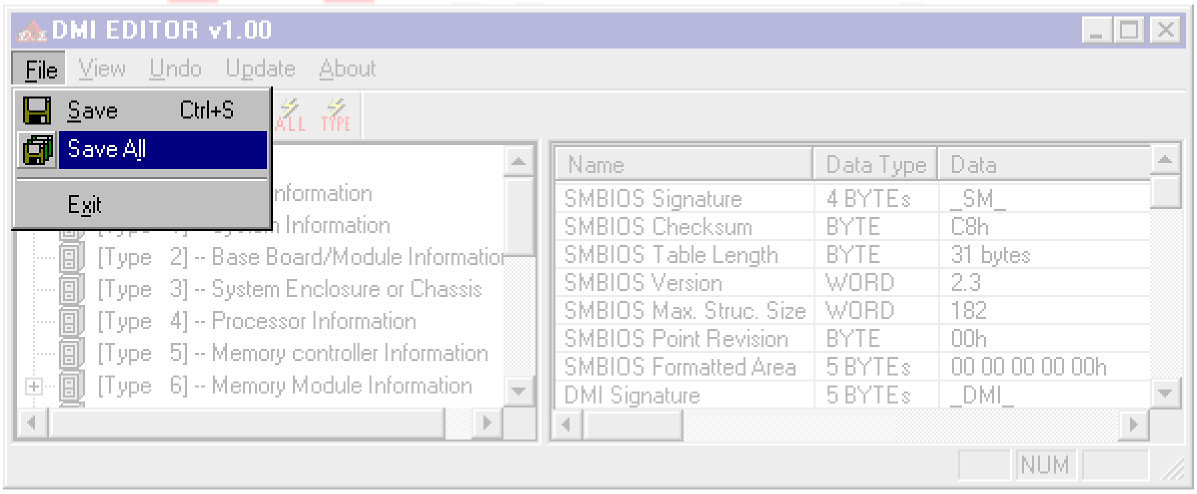
File name: SMBIOS Header.txt

Save as type: Document(.txt)

□ Open as read-only

Saving all SMBIOS information to file

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



...nformation
...n Information
[Type 2] -- Base Board/Module Informatio
[Type 3] -- System Enclosure or Chassis
[Type 4] -- Processor Information
[Type 5] -- Memory controller Information
[Type 6] -- Memory Module Information

Name	Data Type	Data
SMBIOS Signature	4 BYTEs	_SM_
SMBIOS Checksum	BYTE	C8h
SMBIOS Table Length	BYTE	31 bytes
SMBIOS Version	WORD	2.3
SMBIOS Max. Struc. Size	WORD	182
SMBIOS Point Revision	BYTE	00h
SMBIOS Formatted Area	5 BYTEs	00 00 00 00 00h
DMI Signature	5 BYTEs	_DMI_

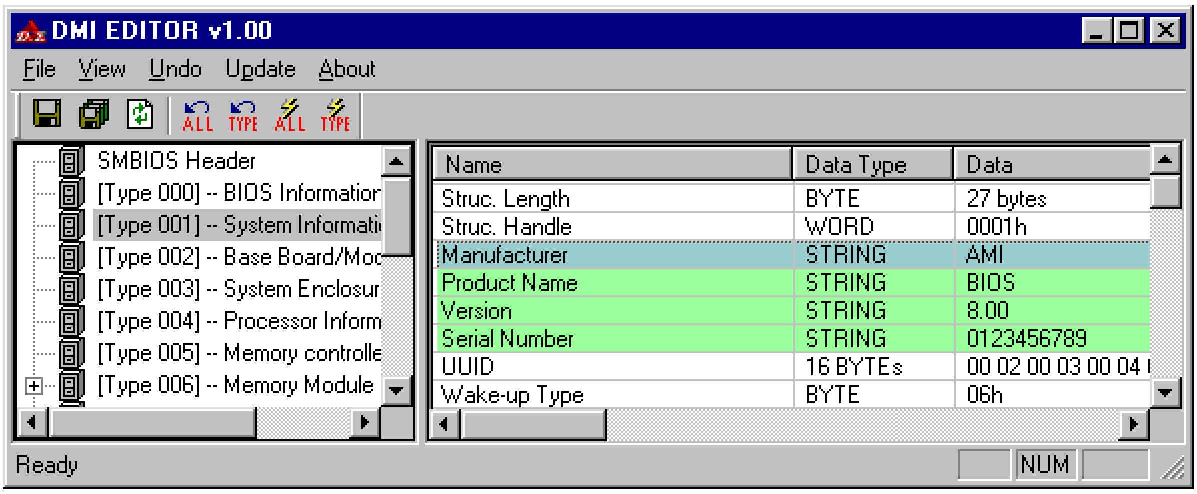
2. Input path/file name on dialog box and press Save to get information file Or

   press Cancel to ignore the function. The dialog box is same as above.

Updating SMBIOS type

This function is valid only on AMIBIOS system.

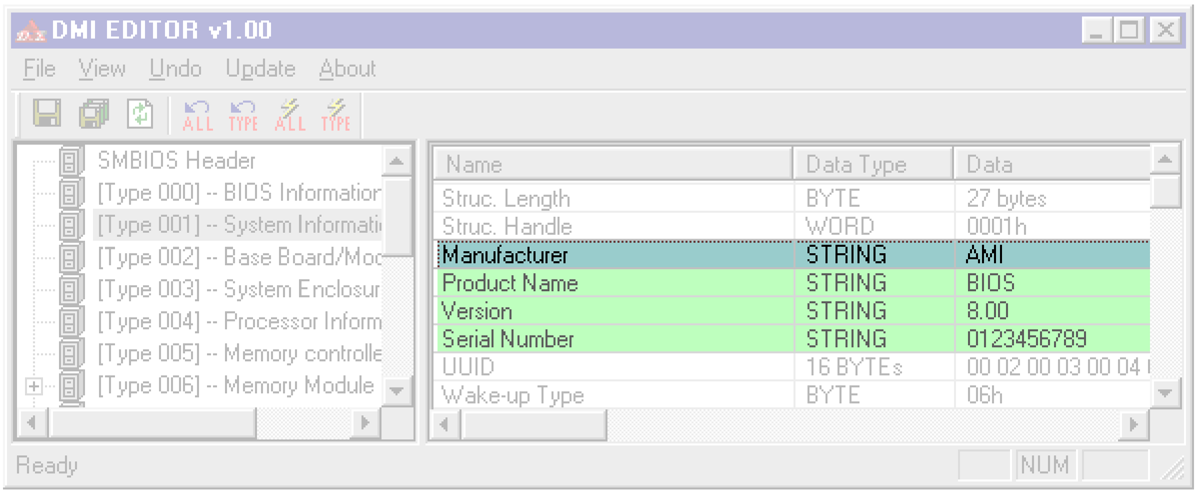
1. As SMBIOS Specification, not of all type can be edited. So you have to know which type contains editable item first.



- SMBIOS Header
- [Type 000] -- BIOS Information
- [Type 001] -- System Informati
- [Type 002] -- Base Board/Moc
- [Type 003] -- System Enclosur
- [Type 004] -- Processor Inform
- [Type 005] -- Memory controlle
- [Type 006] -- Memory Module

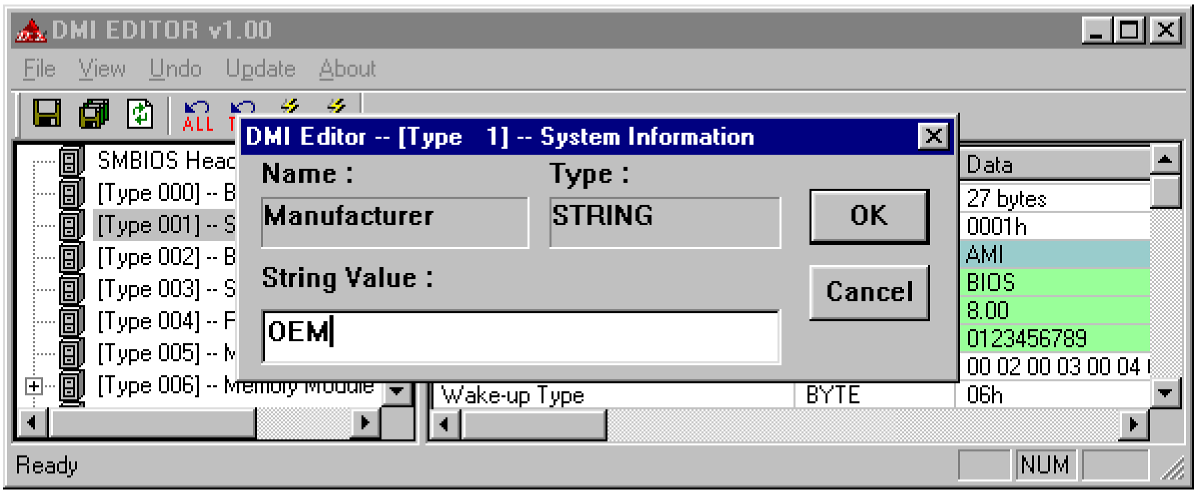
Name	Data Type	Data
Struc. Length	BYTE	27 bytes
Struc. Handle	WORD	0001h
Manufacturer	STRING	AMI
Product Name	STRING	BIOS
Version	STRING	8.00
Serial Number	STRING	0123456789
UUID	16 BYTEs	00 02 00 03 00 04
Wake-up Type	BYTE	06h

2. Select an item that will be modified.


DMI EDITOR v1.00

Name	Data Type	Data
Struc. Length	BYTE	27 bytes
Struc. Handle	WORD	0001h
Manufacturer	STRING	AMI
Product Name	STRING	BIOS
Version	STRING	8.00
Serial Number	STRING	0123456789
UUID	16 BYTEs	00 02 00 03 00 04
Wake-up Type	BYTE	06h

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.

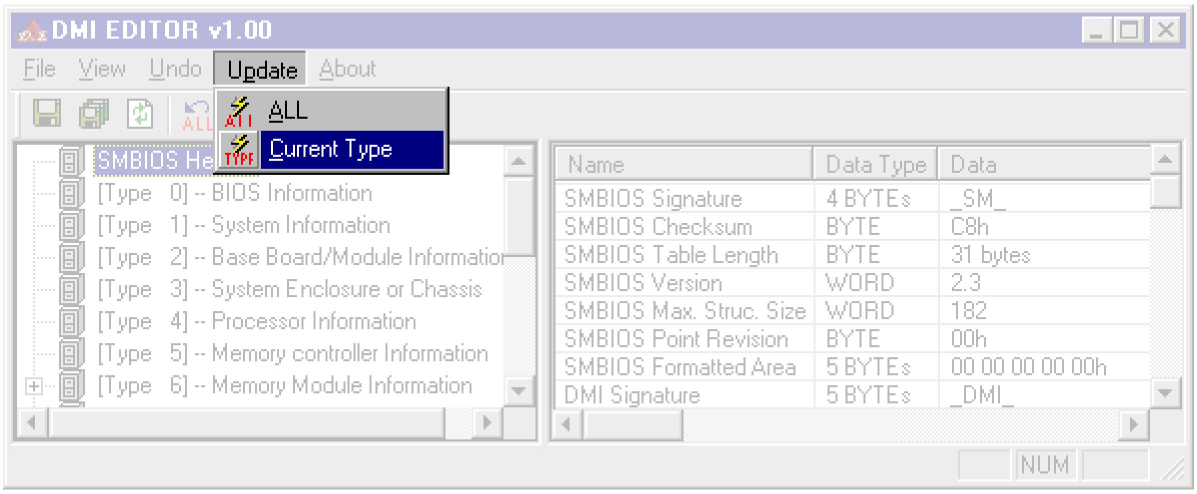

DMI EDITOR v1.00
File View Undo Update About
DMI Editor -- [Type 1] -- System Information
OK
Cancel
Ready
NUM

Name : Manufacturer

Type : STRING

String Value : OEM

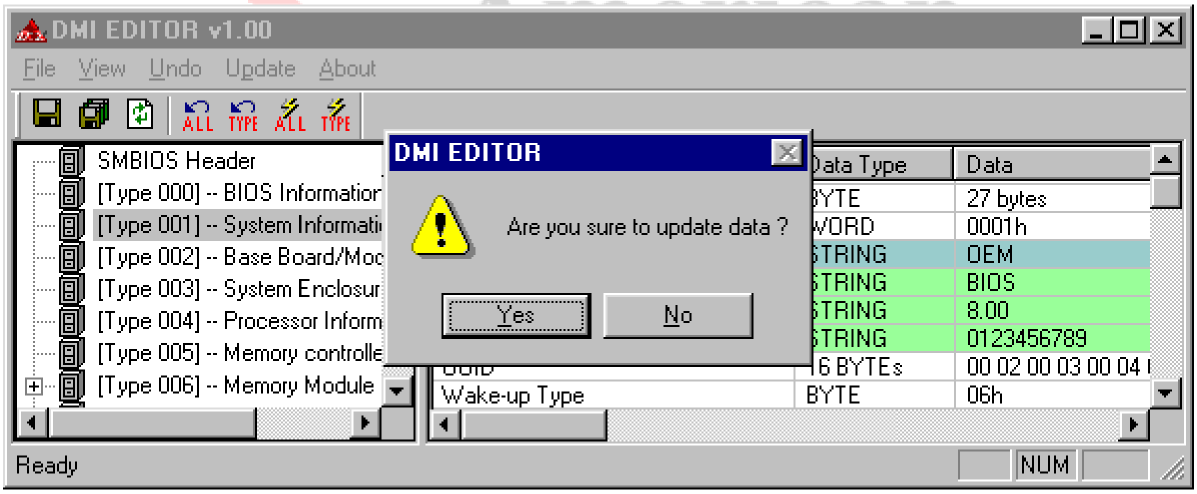
4. Open *Update* drop-down menu and select *Current Type* item Or single click TYPE icon on *Toolbar*.



SMBIOS He
[Type 0] -- BIOS Information
[Type 1] -- System Information
[Type 2] -- Base Board/Module Informatio
[Type 3] -- System Enclosure or Chassis
[Type 4] -- Processor Information
[Type 5] -- Memory controller Information
[Type 6] -- Memory Module Information

Name	Data Type	Data
SMBIOS Signature	4 BYTEs	_SM_
SMBIOS Checksum	BYTE	C8h
SMBIOS Table Length	BYTE	31 bytes
SMBIOS Version	WORD	2.3
SMBIOS Max. Struc. Size	WORD	182
SMBIOS Point Revision	BYTE	00h
SMBIOS Formatted Area	5 BYTEs	00 00 00 00 00h
DMI Signature	5 BYTEs	_DMI_

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

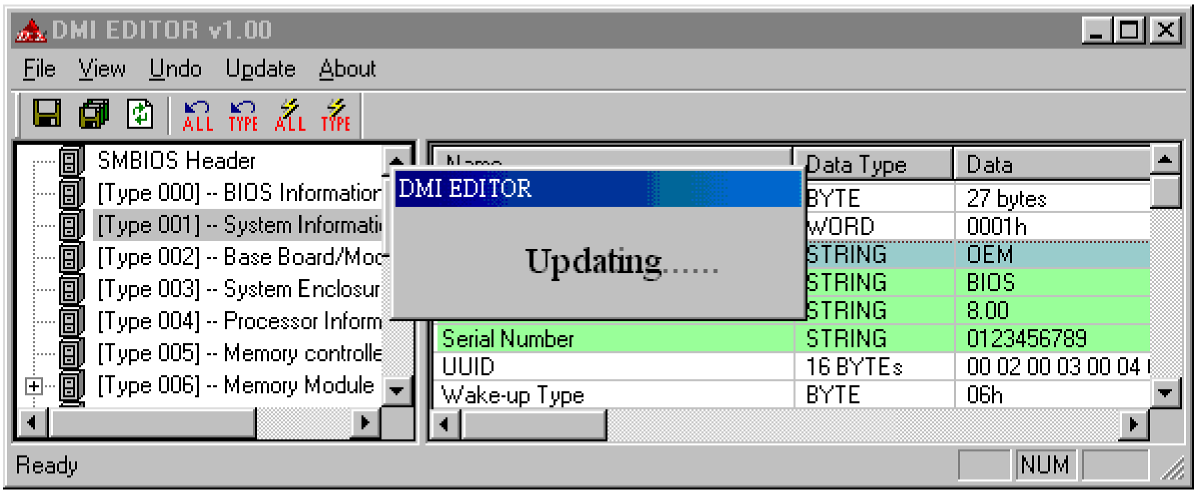


Are you sure to update data ?

SMBIOS Header
[Type 000] -- BIOS Information
[Type 001] -- System Informati
[Type 002] -- Base Board/Mod
[Type 003] -- System Enclosur
[Type 004] -- Processor Inform
[Type 005] -- Memory controlle
[Type 006] -- Memory Module

	Data Type	Data
	BYTE	27 bytes
	WORD	0001h
	STRING	OEM
	STRING	BIOS
	STRING	8.00
	STRING	0123456789
UUID	16 BYTEs	00 02 00 03 00 04
Wake-up Type	BYTE	06h

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



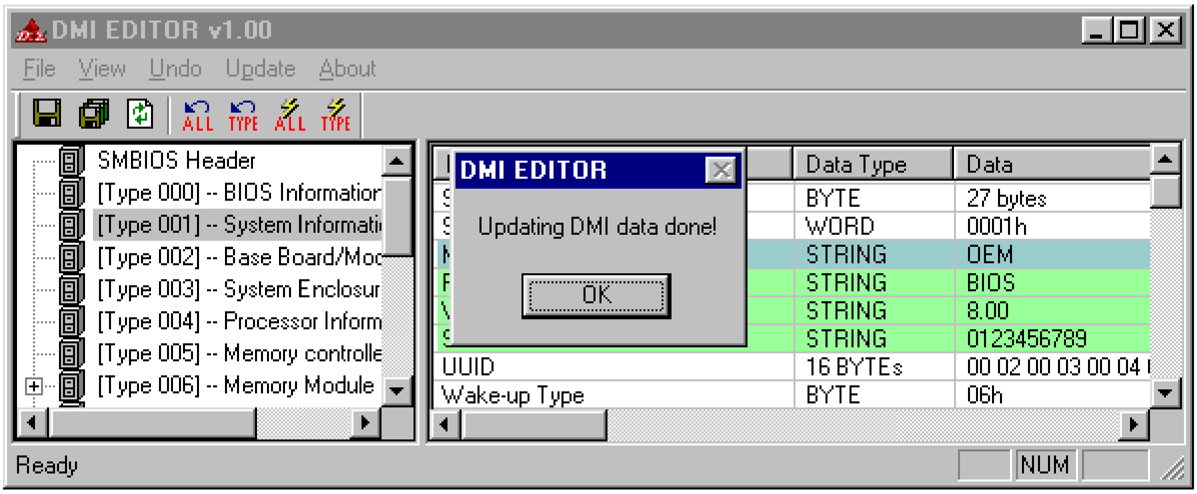
Updating......

SMBIOS Header
[Type 000] -- BIOS Information
[Type 001] -- System Informati
[Type 002] -- Base Board/Moc
[Type 003] -- System Enclosur
[Type 004] -- Processor Inform
[Type 005] -- Memory controlle
[Type 006] -- Memory Module

Name	Data Type	Data
	BYTE	27 bytes
	WORD	0001h
	STRING	OEM
	STRING	BIOS
	STRING	8.00
Serial Number	STRING	0123456789
UUID	16 BYTEs	00 02 00 03 00 04
Wake-up Type	BYTE	06h

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

update operation.


DMI EDITOR v1.00
File View Undo Update About
ALL TYPE ALL TYPE
SMBIOS Header
[Type 000] -- BIOS Information
[Type 001] -- System Informati
[Type 002] -- Base Board/Moc
[Type 003] -- System Enclosur
[Type 004] -- Processor Inform
[Type 005] -- Memory controlle
[Type 006] -- Memory Module
DMI EDITOR
OK
Ready
NUM

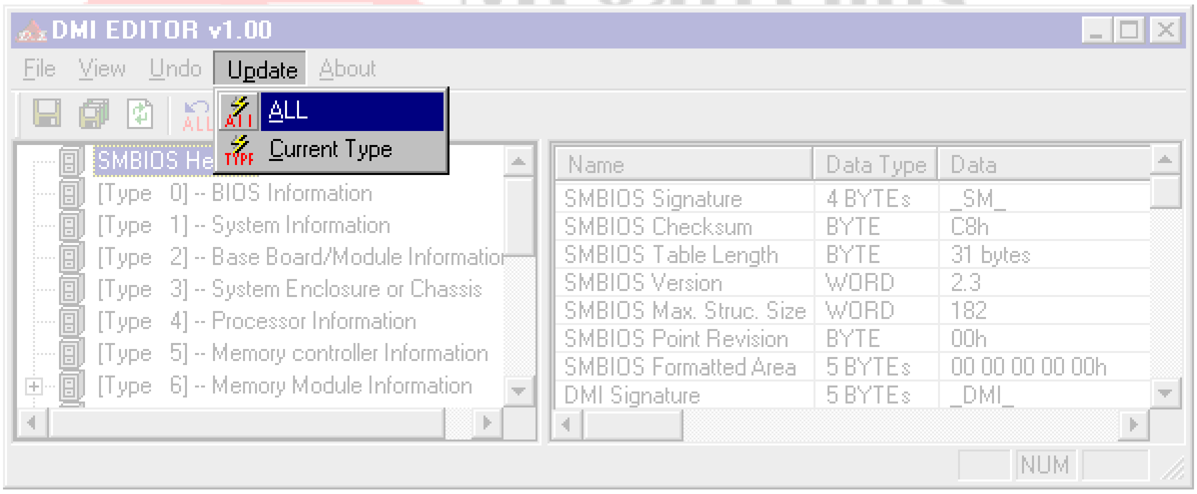
Updating DMI data done!

	Data Type	Data
S	BYTE	27 bytes
S	WORD	0001h
M	STRING	OEM
F	STRING	BIOS
V	STRING	8.00
S	STRING	0123456789
UUID	16 BYTEs	00 02 00 03 00 04
Wake-up Type	BYTE	06h

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 ALL icon on *Toolbar*.

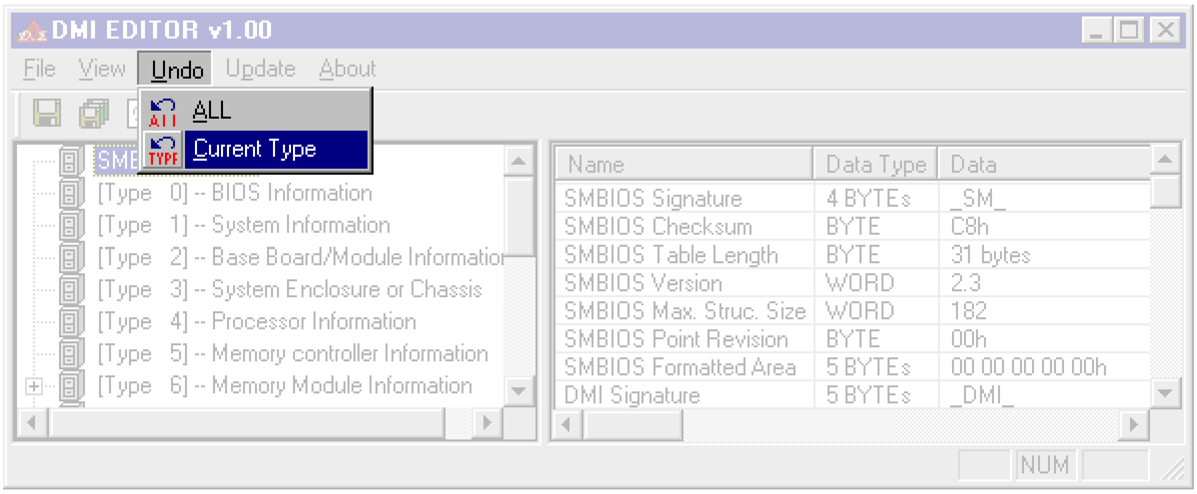

DMI EDITOR v1.00
File View Undo Update About
ALL
Current Type
SMBIOS He
[Type 0] -- BIOS Information
[Type 1] -- System Information
[Type 2] -- Base Board/Module Informatio
[Type 3] -- System Enclosure or Chassis
[Type 4] -- Processor Information
[Type 5] -- Memory controller Information
[Type 6] -- Memory Module Information
NUM

Name	Data Type	Data
SMBIOS Signature	4 BYTEs	_SM_
SMBIOS Checksum	BYTE	C8h
SMBIOS Table Length	BYTE	31 bytes
SMBIOS Version	WORD	2.3
SMBIOS Max. Struc. Size	WORD	182
SMBIOS Point Revision	BYTE	00h
SMBIOS Formatted Area	5 BYTEs	00 00 00 00 00h
DMI Signature	5 BYTEs	_DMI_

3. See step.5-7 of Update SMBIOS Type to finish update operation.

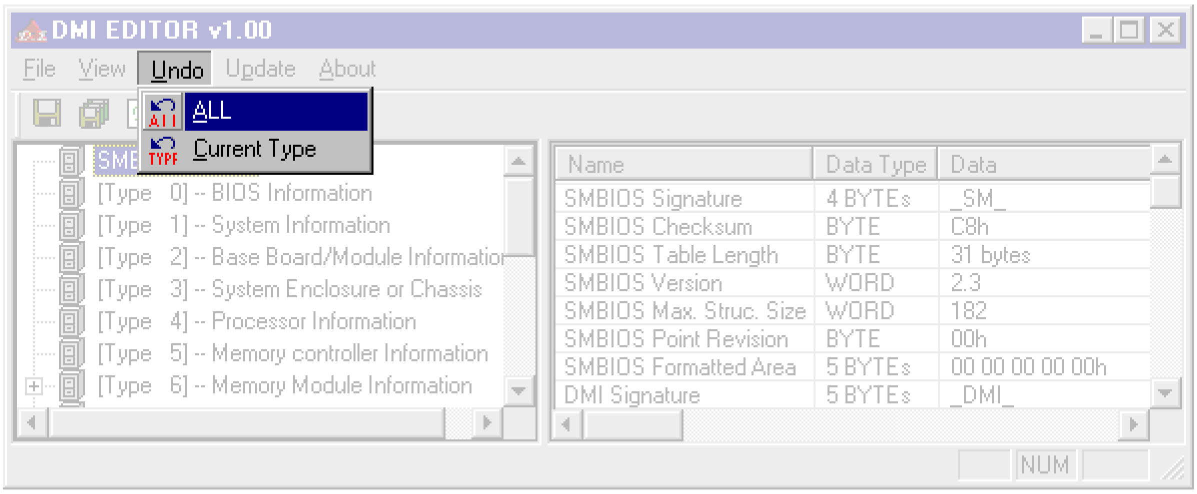
Undoing current type and Undoing all

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 TYPE icon on *Toolbar*.


DMI EDITOR v1.00
File
View
Undo
Update
About
ALL
Current Type
SMB
[Type 0] -- BIOS Information
[Type 1] -- System Information
[Type 2] -- Base Board/Module Informatio
[Type 3] -- System Enclosure or Chassis
[Type 4] -- Processor Information
[Type 5] -- Memory controller Information
[Type 6] -- Memory Module Information
NUM

Name	Data Type	Data
SMBIOS Signature	4 BYTEs	_SM_
SMBIOS Checksum	BYTE	C8h
SMBIOS Table Length	BYTE	31 bytes
SMBIOS Version	WORD	2.3
SMBIOS Max. Struc. Size	WORD	182
SMBIOS Point Revision	BYTE	00h
SMBIOS Formatted Area	5 BYTEs	00 00 00 00 00h
DMI Signature	5 BYTEs	_DMI_

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


DMI EDITOR v1.00
File
View
Undo
Update
About
ALL
Current Type
SMB
[Type 0] -- BIOS Information
[Type 1] -- System Information
[Type 2] -- Base Board/Module Informatio
[Type 3] -- System Enclosure or Chassis
[Type 4] -- Processor Information
[Type 5] -- Memory controller Information
[Type 6] -- Memory Module Information
NUM

Name	Data Type	Data
SMBIOS Signature	4 BYTEs	_SM_
SMBIOS Checksum	BYTE	C8h
SMBIOS Table Length	BYTE	31 bytes
SMBIOS Version	WORD	2.3
SMBIOS Max. Struc. Size	WORD	182
SMBIOS Point Revision	BYTE	00h
SMBIOS Formatted Area	5 BYTEs	00 00 00 00 00h
DMI Signature	5 BYTEs	_DMI_

Chapter 5 AMIUCP v1.xx

Overview

Utility Configuration Program (hereafter referred to as UCP) is designed for the people who intend to customize AMIBIOS ROM Utility's capabilities for specific product or user. This is a utility to configure default settings/behaviors of certain compatible utilities, pointed below. User can use it to get target utility information named Utility Identification Information (hereafter referred to as UII), manipulate Utility Auxiliary File (hereafter referred to as UAF) and adjust the settings for target utility without asking AMI to rebuild a new program.

This specification defines a method to standardize necessary structures presented in AMIBIOS ROM Utility and referred by UCP, moreover, the basic functionalities will be also defined here.

Compatible Utilities

Following utilities support AMIUCP standard and supported options will be described after the AMIUCP standard user guide.

• AFUDOS	v4.24 or later
• AFUWIN	v4.34 or later
• AFULNX2	v4.22 or later
• AFUBSD	v3.08 or later

Features

This utility offers following features:

- Display Utility Identification Information
- Insert
- Delete
- Extract
- Replace
- Insert ROM image
- Exchange ROM image
- Insert Default Command String
- Exchange Default Command String

- Edit OEM Version

Requirements

Supported Operating System

AMIBCP Utility is supported in the following operating systems:

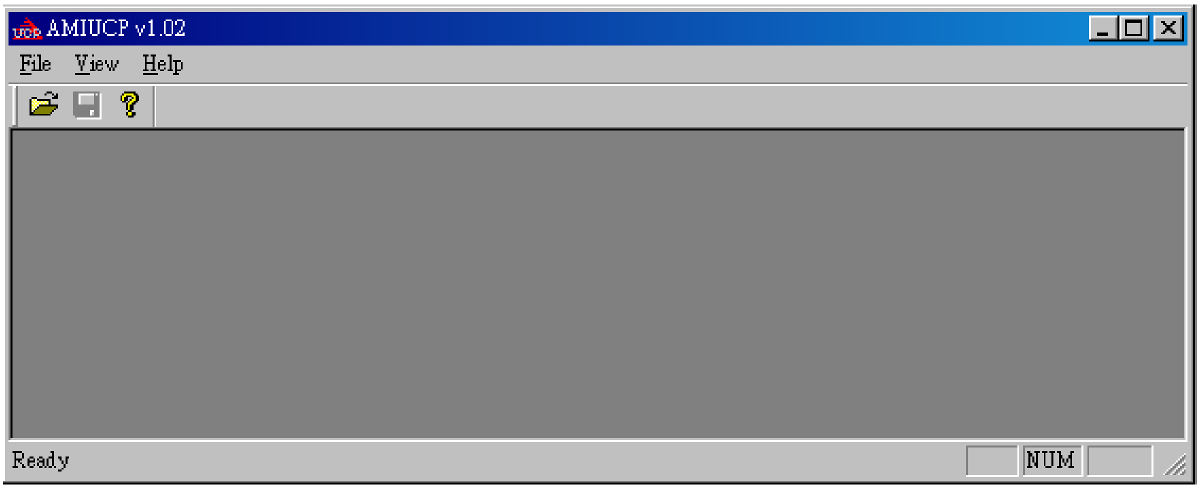
- Microsoft® Windows® 98 (v3.30 and above will not support any more.)
- Microsoft® Windows® ME
- Microsoft® Windows® NT 4.0
- Microsoft® Windows® 2000
- Microsoft® Windows® XP/XP64
- Microsoft® Windows® PE
- Microsoft® Windows® Vista 32/64

Getting Started

Installation

*Copies the **AMIUCP.EXE** executable file to any storage location accessible by the host system and then double-click **AMIUCP** icon to run.*

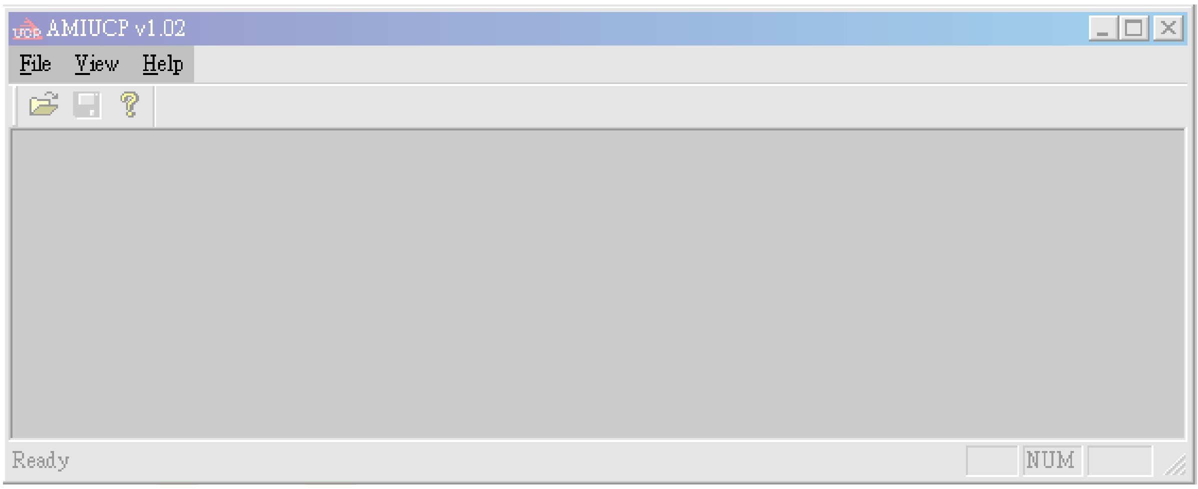
Main Window



Menu Bar

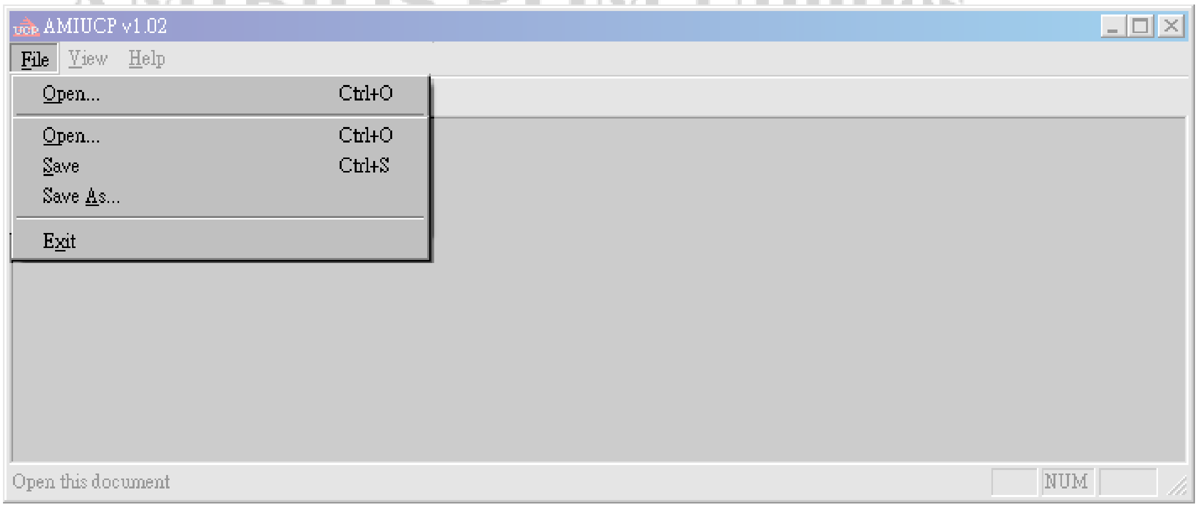
The *Menu bar* is located at the top of the AMIUCP window. The *Menu bar* contains the following:

- File drop-down menu
- View drop-down menu
- Help



File drop-down menu options

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

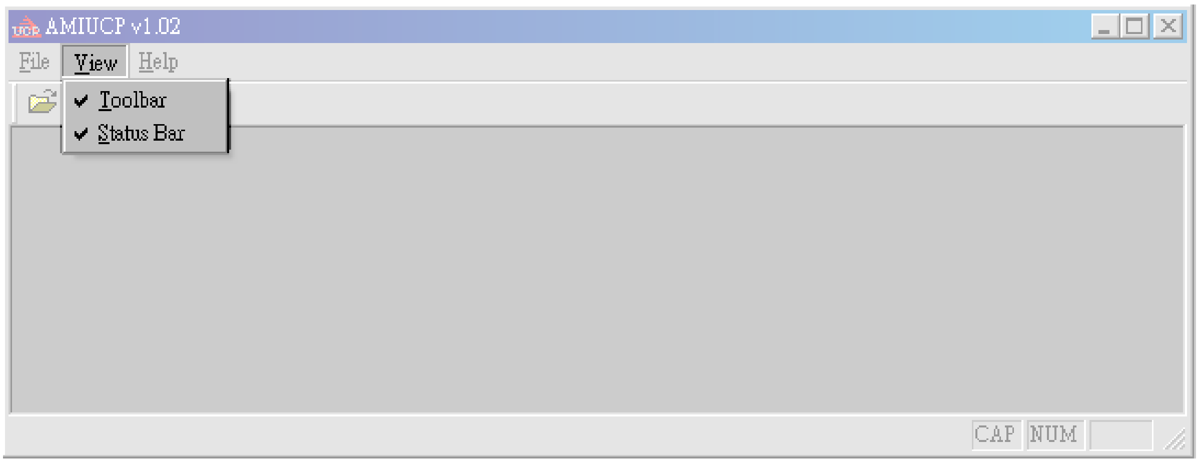


The *File* drop-down menu items are explained in the following table:

File Menu Item List	
Name	**Description**
Open	Open an existing executable file.
Save	Save any changes you have made to the existing executable file.
Save As	Same feature as *Save* menu item. In addition, it also allows you to specify the location and to change the existing file name.
Exit	Quit the application.

View drop-down menu options

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

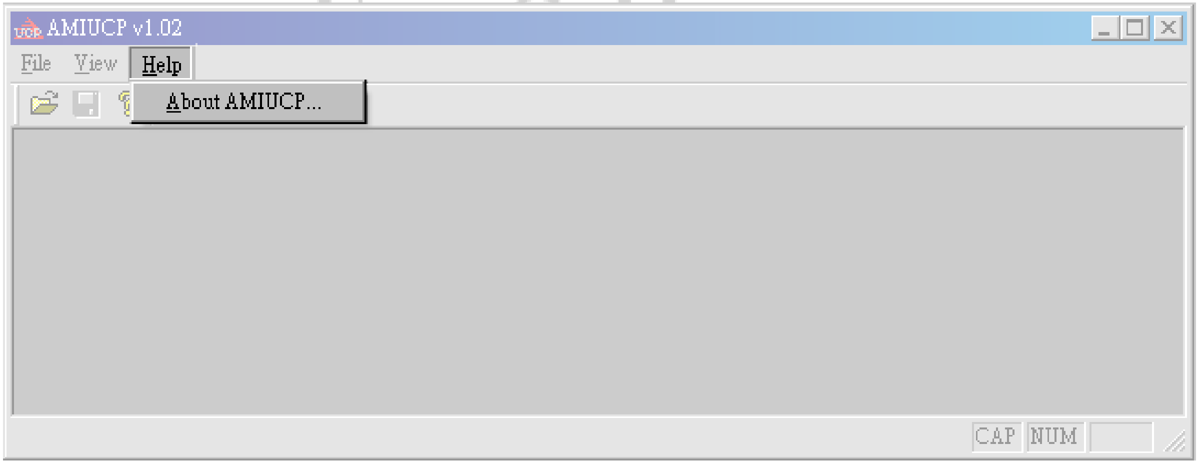


The *View* drop-down menu items are explained in the following table:

View Menu Item List	
Name	**Description**
Toolbar	Display or hide the *Toolbar*. The *Toolbar* is displayed under the *Menu bar*.
Status Bar	Display or hide the *Status Bar*. The *Status Bar* is displayed at the bottom of the AMIUCP window.

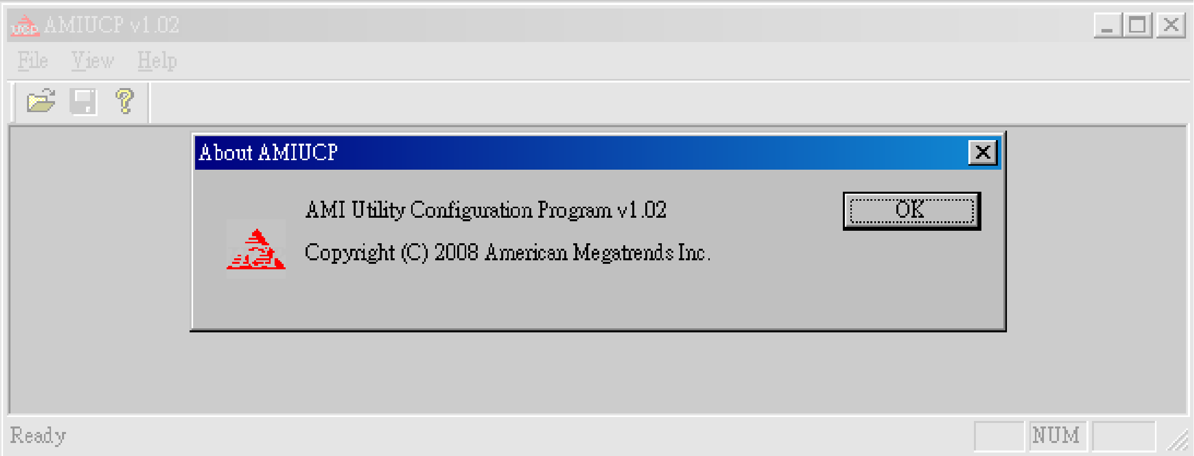
Help drop-down menu options

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



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

View Menu Item List	
Name	**Description**
About AMIUCP	AMIUCP copyrights information.



About AMIUCP

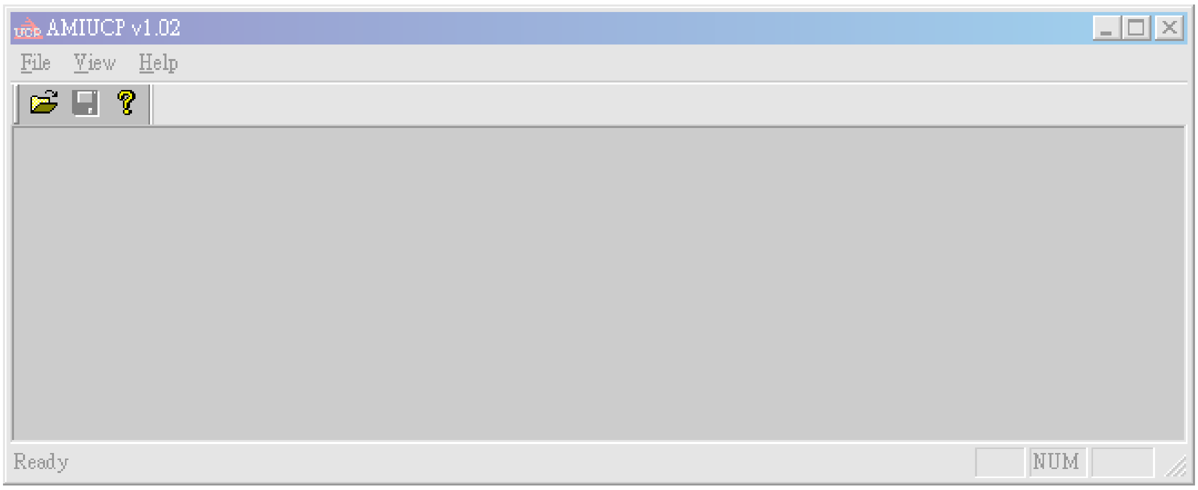
AMI Utility Configuration Program v1.02

Copyright (C) 2008 American Megatrends Inc.

OK

Toolbar

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

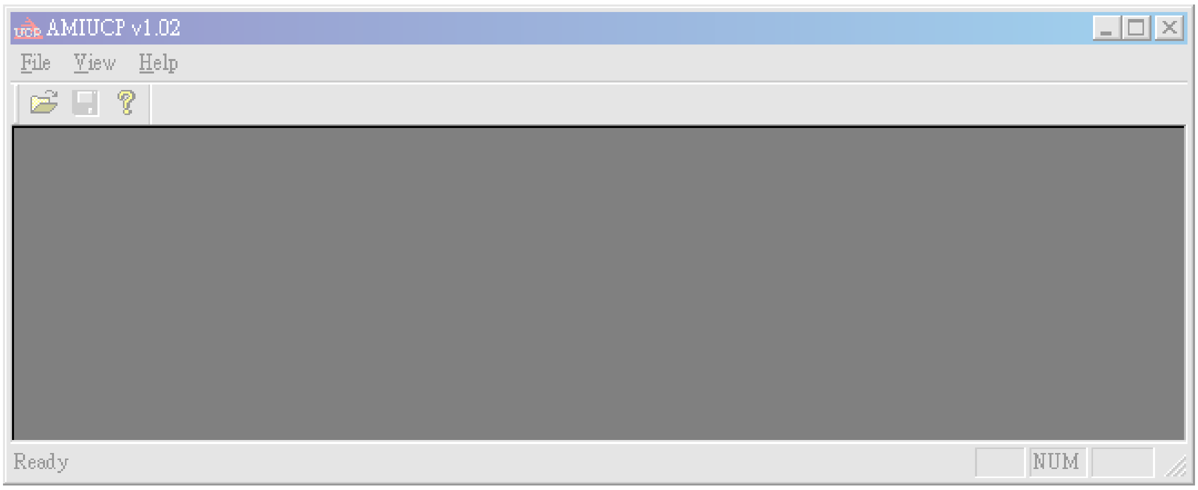


The *Toolbar* icons are explained in the following table:

Toolbar Icon List	
Icon	**Description**
📂	Open an existing executable file.
💾	Save any changes you have made to the executable file.
?	Display AMIUCP copyrights information.

Body Frame

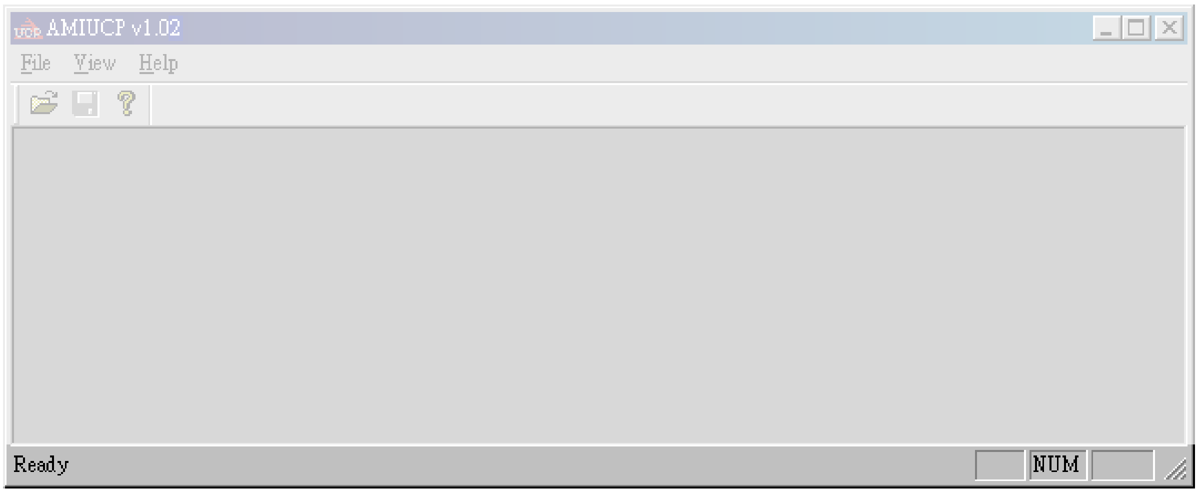
The *Body Frame* is the main frame of AMIUCP. It is located under the *Menu bar* and *Toolbar*. The AMIUCP main functions are displayed in the body frame screen.



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:

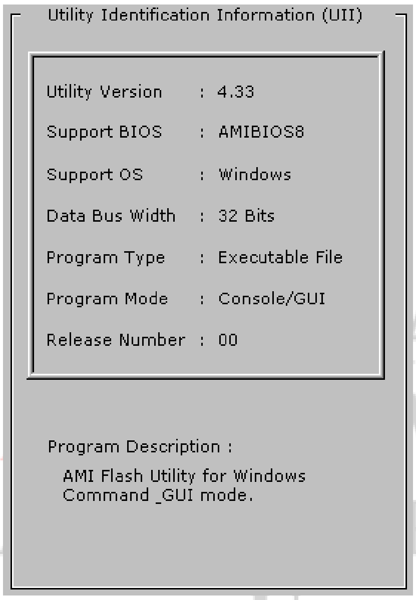
Item	Description
CAP	The Caps Lock key is latched down.
NUM	The Num Lock key is latched down.
SCRL	The Scroll Lock key is latched down.



Function Frame

Utility Identification Information (UII)

UII provides enough information to our customers to control their projects with suitable version of utilities. With UII, customers can leave the problem that build faithless ROM image with unsuitable building tools or figure out the cause more quickly, and further, customer can also understand what kind of platform supported by ROM utilities on hand, this will increase the their confidence in using AMITOOLs.



Utility Identification Information (UII)

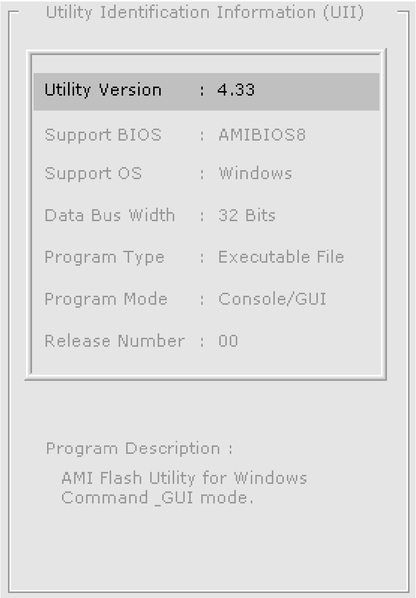
```
Utility Version  :  4.33
Support BIOS     :  AMIBIOS8
Support OS       :  Windows
Data Bus Width   :  32 Bits
Program Type     :  Executable File
Program Mode     :  Console/GUI
Release Number   :  00
```

Program Description :
AMI Flash Utility for Windows
Command _GUI mode.

Field

- Utility Version

  This value holds current utility version number.



Utility Identification Information (UII)

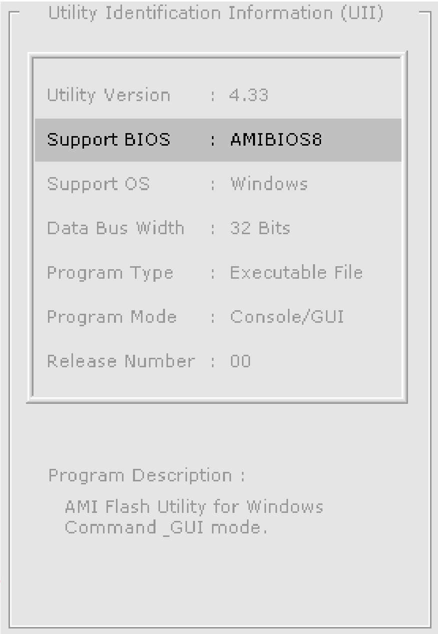
```
Utility Version  :  4.33
Support BIOS     :  AMIBIOS8
Support OS       :  Windows
Data Bus Width   :  32 Bits
Program Type     :  Executable File
Program Mode     :  Console/GUI
Release Number   :  00
```

Program Description :
AMI Flash Utility for Windows
Command _GUI mode.

- Supporting BIOS

  This field indicates what BIOS is supporting. The definition is listed in following:

  1. ALL
  2. AMIBIOS8
  3. UEFI
  4. AMIBIOS8 and UEFI

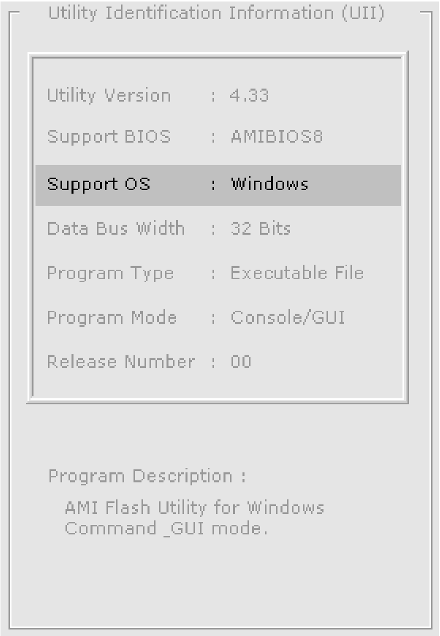


```
Utility Identification Information (UII)

Utility Version   :  4.33
Support BIOS      :  AMIBIOS8
Support OS        :  Windows
Data Bus Width    :  32 Bits
Program Type      :  Executable File
Program Mode      :  Console/GUI
Release Number    :  00

Program Description :
  AMI Flash Utility for Windows
  Command _GUI mode.
```

- Supporting Operating System

  This field indicates what O/S is supporting. The definition is listed in following:

  1. DOS
  2. EFI Shell
  3. Windows
  4. Linux
  5. FreeBSD
  6. MAC OS
  7. Multi-Platform

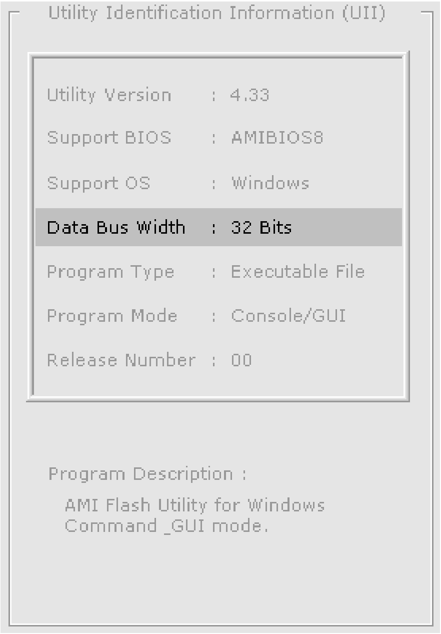


Utility Identification Information (UII)

```
Utility Version   :  4.33
Support BIOS      :  AMIBIOS8
Support OS        :  Windows
Data Bus Width    :  32 Bits
Program Type      :  Executable File
Program Mode      :  Console/GUI
Release Number    :  00
```

Program Description :
AMI Flash Utility for Windows
Command _GUI mode.

- Data Bus Width
  This field indicates what CPU Data Bus width is supporting. The definition is listed in following:
  1. 16 Bits
  2. 16/32 Bits(For DOS Extended Program only)
  3. 32 Bits
  4. 64 Bits



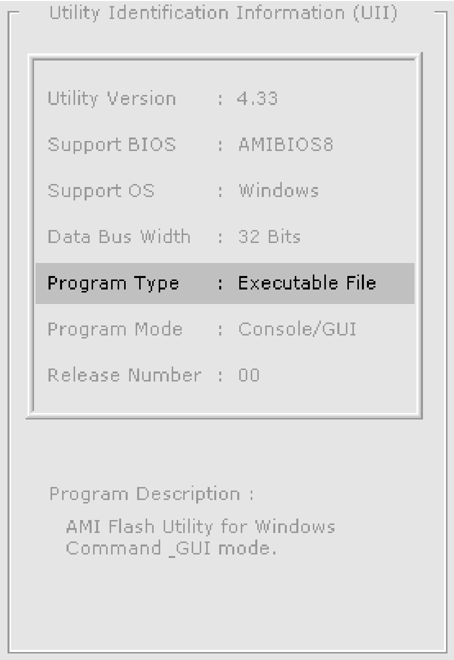
Utility Identification Information (UII)

```
Utility Version   :  4.33
Support BIOS      :  AMIBIOS8
Support OS        :  Windows
Data Bus Width    :  32 Bits
Program Type      :  Executable File
Program Mode      :  Console/GUI
Release Number    :  00
```

Program Description :
AMI Flash Utility for Windows
Command _GUI mode.

- Program Type
  This field indicates program file type. The definition is listed in following:

1. Executable File
2. Dynamically Link Library(Windows only)
3. Driver File



Utility Identification Information (UII)

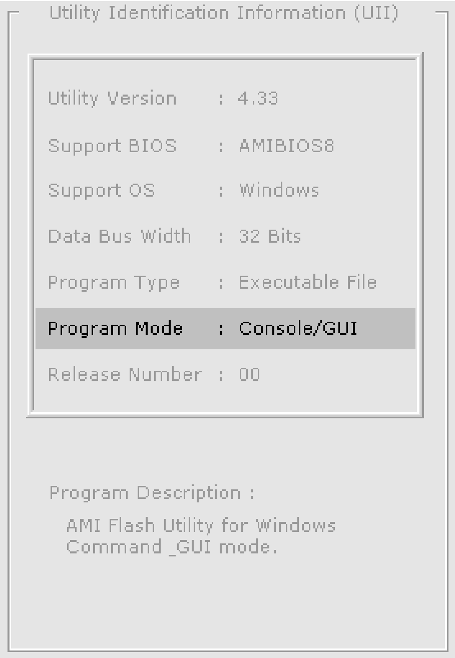
Utility Version : 4.33
Support BIOS : AMIBIOS8
Support OS : Windows
Data Bus Width : 32 Bits
Program Type : Executable File
Program Mode : Console/GUI
Release Number : 00

Program Description :
AMI Flash Utility for Windows
Command _GUI mode.

- Program Mode

  This field indicates program mode. The definition is listed in following:

  1. API
  2. Console
  3. GUI
  4. Console/GUI



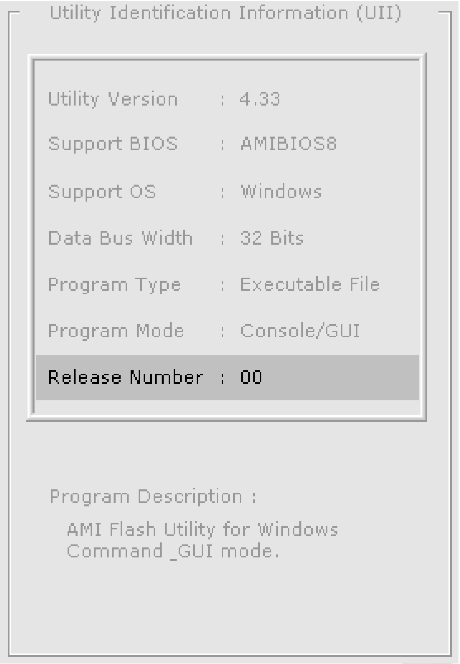
Utility Identification Information (UII)

Utility Version : 4.33
Support BIOS : AMIBIOS8
Support OS : Windows
Data Bus Width : 32 Bits
Program Type : Executable File
Program Mode : Console/GUI
Release Number : 00

Program Description :
AMI Flash Utility for Windows
Command _GUI mode.

- Release Number

  This field indicates release number labeled in SourceSafe.



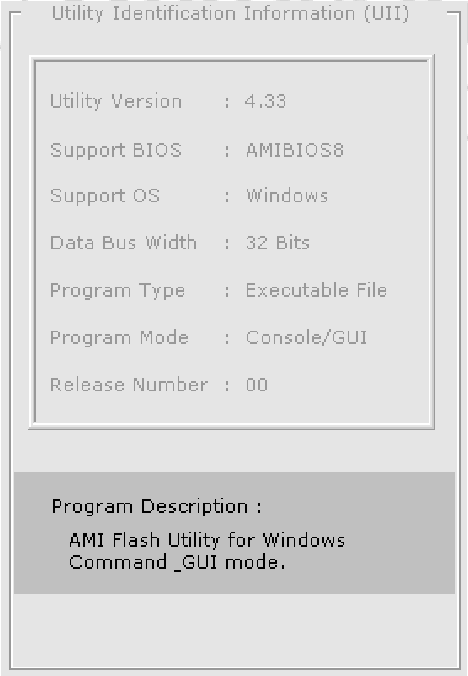
Utility Identification Information (UII)

Utility Version : 4.33
Support BIOS : AMIBIOS8
Support OS : Windows
Data Bus Width : 32 Bits
Program Type : Executable File
Program Mode : Console/GUI
Release Number : 00

Program Description :
AMI Flash Utility for Windows
Command _GUI mode.

- Program Description

  This field indicates the utility identification information in the program.



Utility Identification Information (UII)

Utility Version : 4.33
Support BIOS : AMIBIOS8
Support OS : Windows
Data Bus Width : 32 Bits
Program Type : Executable File
Program Mode : Console/GUI
Release Number : 00

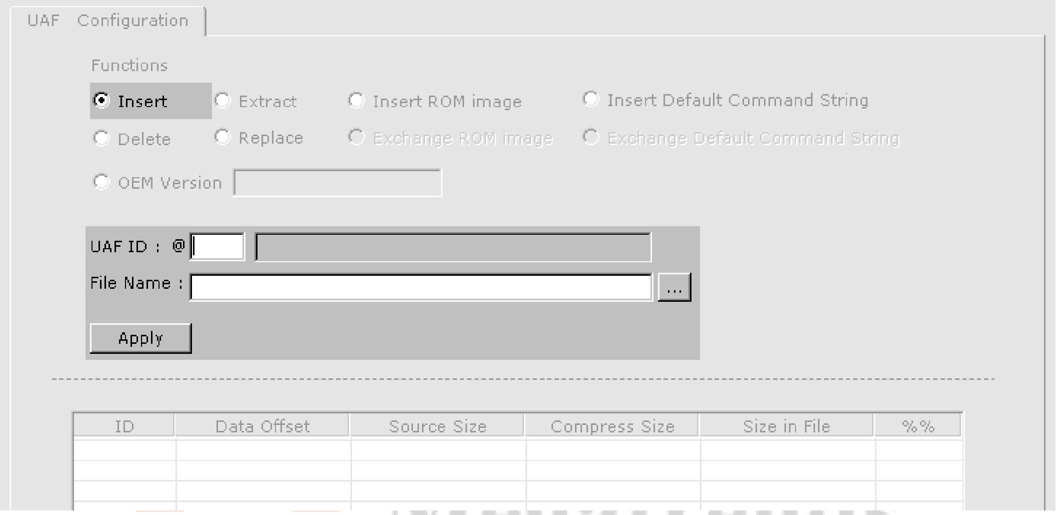
Program Description :
AMI Flash Utility for Windows
Command _GUI mode.

Utility Auxiliary File (UAF) Manipulation

UAF manipulation should be included in UCP as default functionality. UCP must offer following features at least for UAF:

- **Insert**

  This function allows user to insert a new UAF into target program.



● Insert ○ Extract ○ Insert ROM image ○ Insert Default Command String
○ Delete ○ Replace ○ Exchange ROM image ○ Exchange Default Command String
○ OEM Version

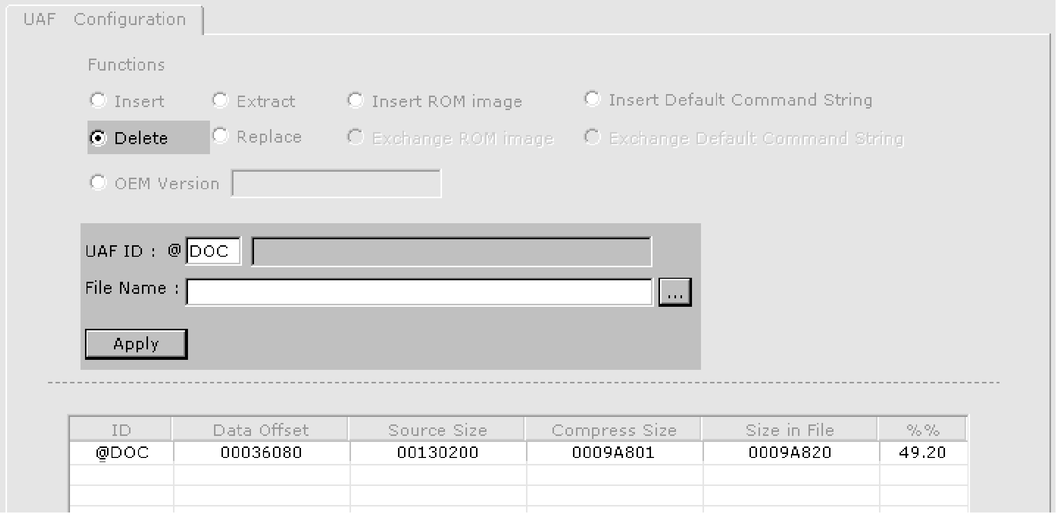
UAF ID : @

File Name :

ID	Data Offset	Source Size	Compress Size	Size in File	%%

- **Delete**

  This function allows user to delete a UAF from target program.



○ Insert ○ Extract ○ Insert ROM image ○ Insert Default Command String
● Delete ○ Replace ○ Exchange ROM image ○ Exchange Default Command String
○ OEM Version

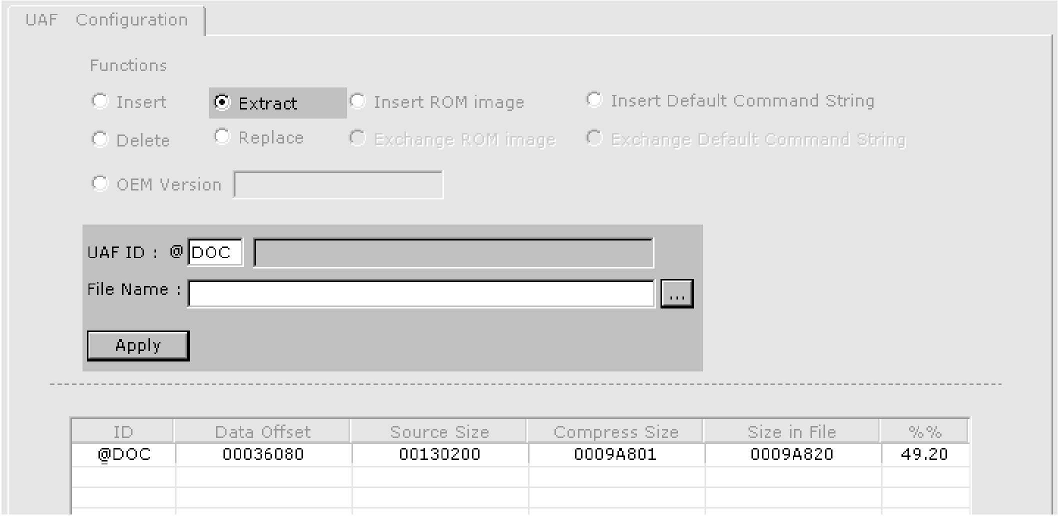
UAF ID : @DOC

File Name :

ID	Data Offset	Source Size	Compress Size	Size in File	%%
@DOC	00036080	00130200	0009A801	0009A820	49.20

- **Extract**

  This function allows user to extract a UAF from target program to any storage.



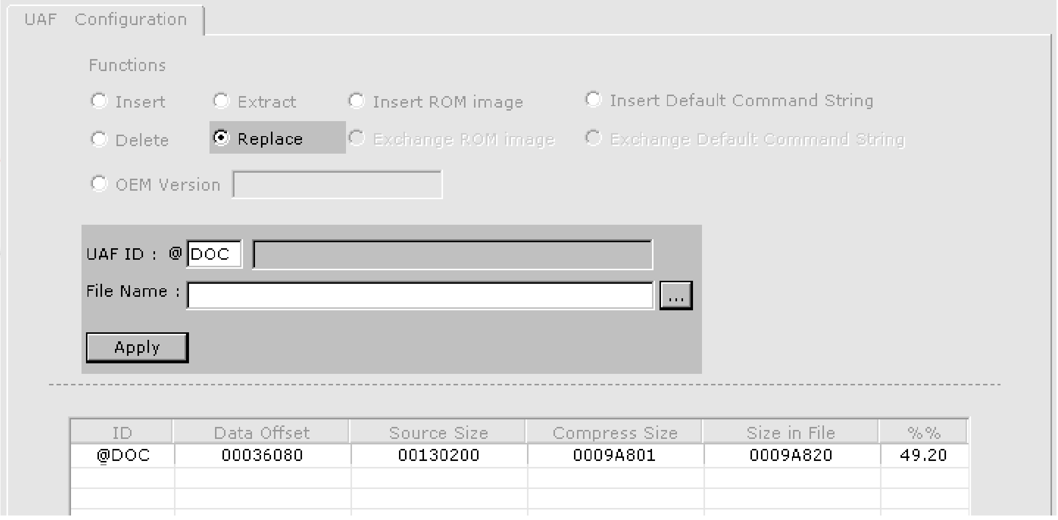
○ Insert ● Extract ○ Insert ROM image ○ Insert Default Command String
○ Delete ○ Replace ○ Exchange ROM image ○ Exchange Default Command String
○ OEM Version

UAF ID : @ DOC
File Name :

ID	Data Offset	Source Size	Compress Size	Size in File	%%
@DOC	00036080	00130200	0009A801	0009A820	49.20

- **Replace**

  This function allows user to substitute an existing UAF into target program with a new one.



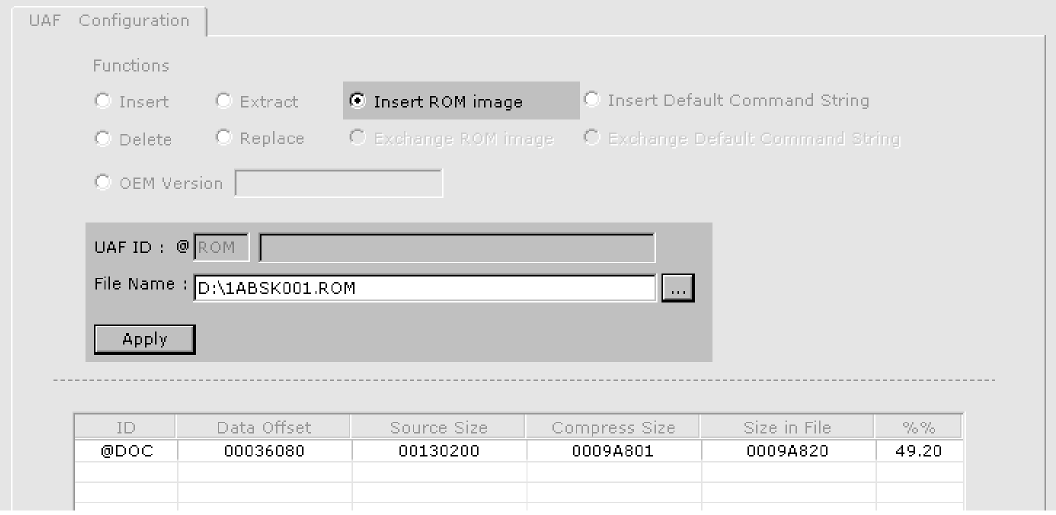
○ Insert ○ Extract ○ Insert ROM image ○ Insert Default Command String
○ Delete ● Replace ○ Exchange ROM image ○ Exchange Default Command String
○ OEM Version

UAF ID : @ DOC
File Name :

ID	Data Offset	Source Size	Compress Size	Size in File	%%
@DOC	00036080	00130200	0009A801	0009A820	49.20

- **Insert ROM image**

  This function allows user to insert a ROM image into target program.



○ Insert ○ Extract ● Insert ROM image ○ Insert Default Command String

○ Delete ○ Replace ○ Exchange ROM image ○ Exchange Default Command String

○ OEM Version

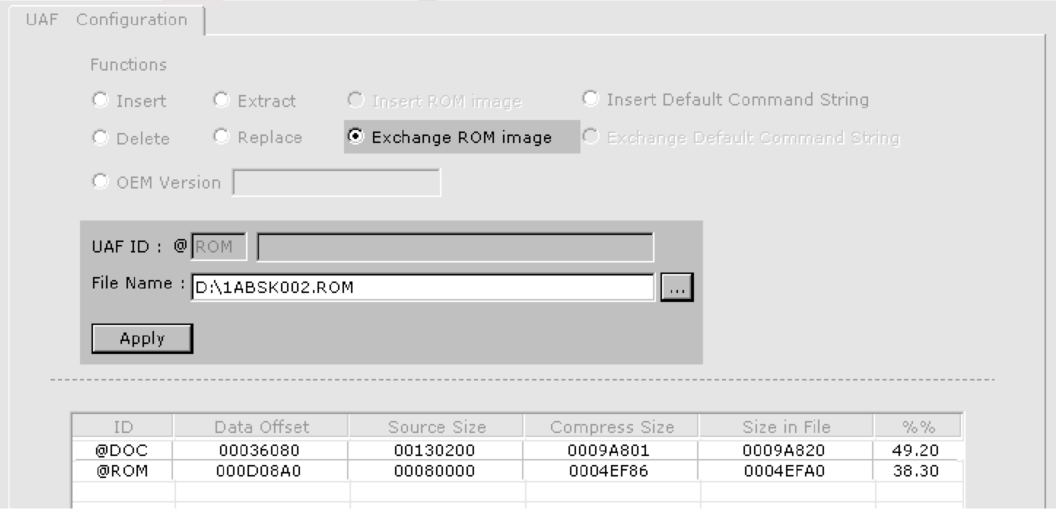
UAF ID : @ROM

File Name : D:\1ABSK001.ROM

ID	Data Offset	Source Size	Compress Size	Size in File	%%
@DOC	00036080	00130200	0009A801	0009A820	49.20

- **Exchange ROM image**

  This function allows user to substitute an existing @ROM UAF into target program with a new one.



○ Insert ○ Extract ○ Insert ROM image ○ Insert Default Command String

○ Delete ○ Replace ● Exchange ROM image ○ Exchange Default Command String

○ OEM Version

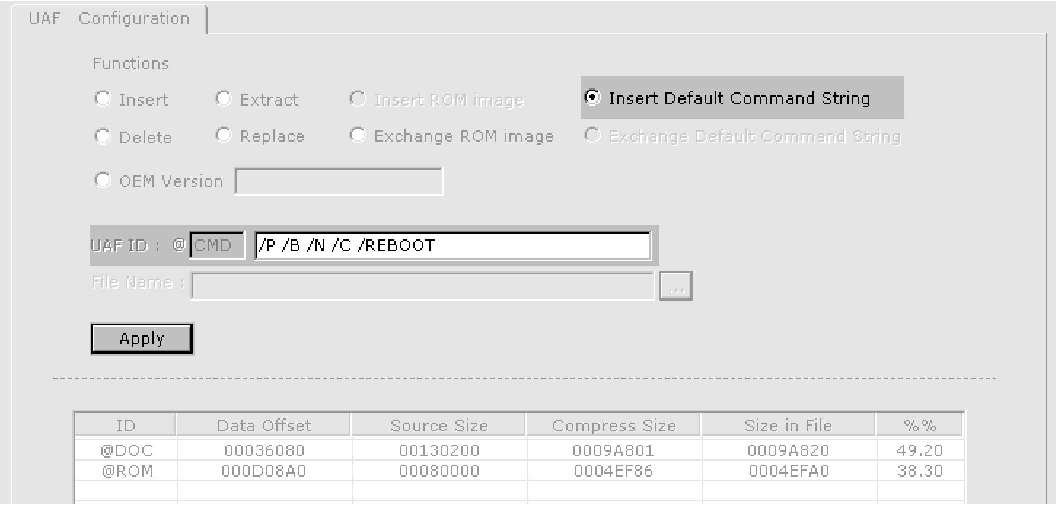
UAF ID : @ROM

File Name : D:\1ABSK002.ROM

ID	Data Offset	Source Size	Compress Size	Size in File	%%
@DOC	00036080	00130200	0009A801	0009A820	49.20
@ROM	000D08A0	00080000	0004EF86	0004EFA0	38.30

- **Insert Default Command String**

  This function allows user to insert default string into target program.



UAF Configuration

Functions

○ Insert ○ Extract ○ Insert ROM image ● Insert Default Command String
○ Delete ○ Replace ○ Exchange ROM image ○ Exchange Default Command String
○ OEM Version

UAF ID : @ CMD /P /B /N /C /REBOOT

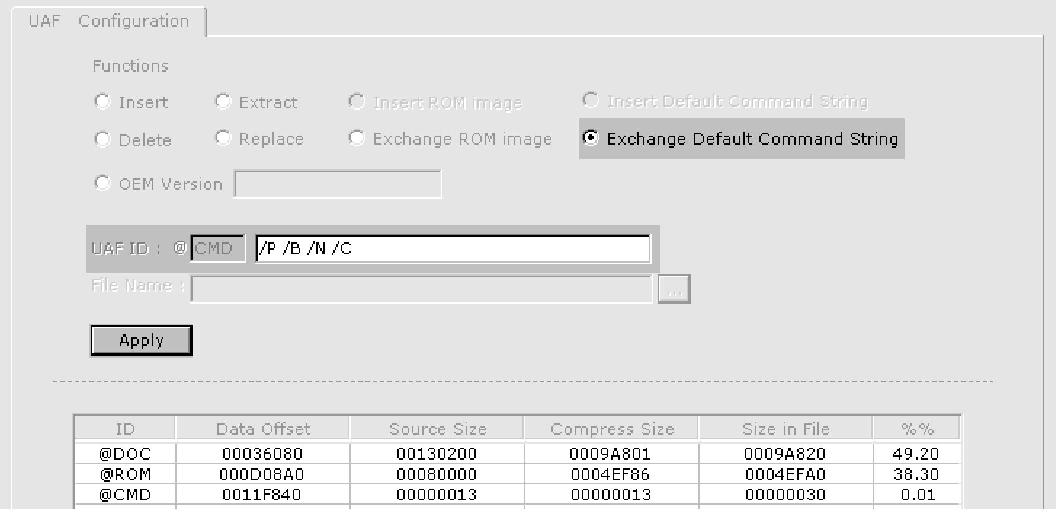
File Name :

Apply

ID	Data Offset	Source Size	Compress Size	Size in File	%%
@DOC	00036080	00130200	0009A801	0009A820	49.20
@ROM	000D08A0	00080000	0004EF86	0004EFA0	38.30

- **Exchange Default Command String**

  This function allows user to substitute an existing @CMD UAF into target program.



UAF Configuration

Functions

○ Insert ○ Extract ○ Insert ROM image ○ Insert Default Command String
○ Delete ○ Replace ○ Exchange ROM image ● Exchange Default Command String
○ OEM Version

UAF ID : @ CMD /P /B /N /C

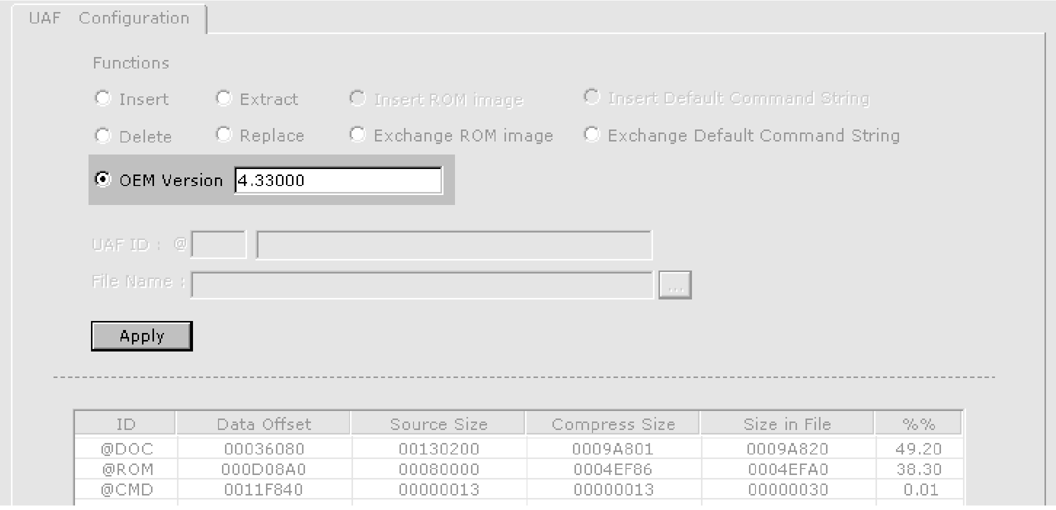
File Name :

Apply

ID	Data Offset	Source Size	Compress Size	Size in File	%%
@DOC	00036080	00130200	0009A801	0009A820	49.20
@ROM	000D08A0	00080000	0004EF86	0004EFA0	38.30
@CMD	0011F840	00000013	00000013	00000030	0.01

- **Enter OEM Version**

  This function allows user to enter OEM Version into target program.



UAF Configuration

Functions

○ Insert ○ Extract ○ Insert ROM image ○ Insert Default Command String
○ Delete ○ Replace ○ Exchange ROM image ○ Exchange Default Command String
● OEM Version 4.33000

UAF ID : @

File Name :

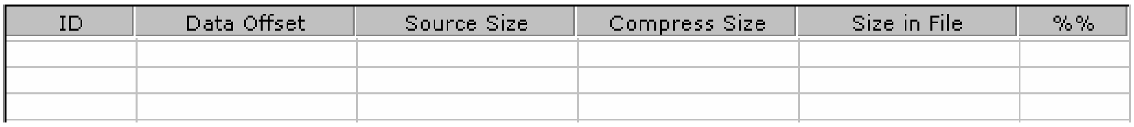
Apply

ID	Data Offset	Source Size	Compress Size	Size in File	%%
@DOC	00036080	00130200	0009A801	0009A820	49.20
@ROM	000D08A0	00080000	0004EF86	0004EFA0	38.30
@CMD	0011F840	00000013	00000013	00000030	0.01

Buttons

Icon	Description
...	This button is used to search for a new input file from any storage location.
Apply	This button is used to insert a new UAF into the target program.

UAF Info Frame



ID	Data Offset	Source Size	Compress Size	Size in File	%%

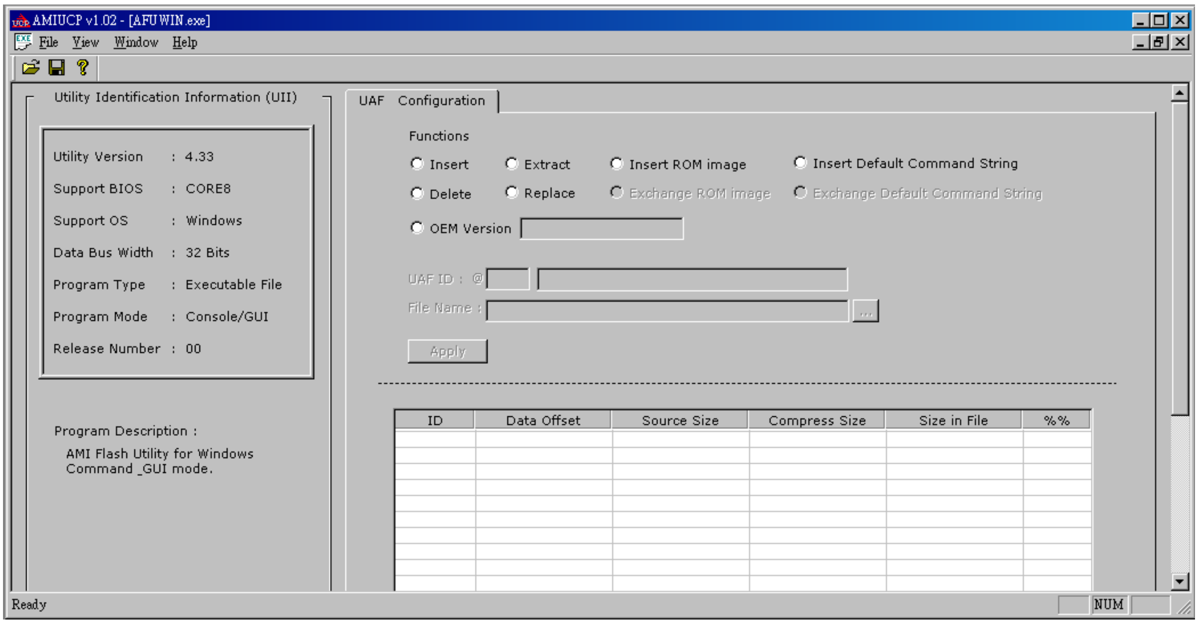
Field	Description
ID	The signature is a four-byte long ASCII string. It hods the user defined signature here to be identified by program file.
Data Offset	This field displays the UAF block data address offset in the program file.
Source Size	Original input file source size in unit of bytes.
Compress Size	Compressed input file source size in unit of bytes.
Size in File	The UAF block data size in unit of bytes.
%%	This field displays the file’s compressed ratio. Usually, 0.00 means the file is uncompressed and -- means it is a linked file.

Functions

To use AMIUCP, you can double-click the executable file icon to open Main Window and press on *Toolbar* to open an existing executable file.

AMIUCP allows you to get target utility information named Utility Identification Information (hereafter referred to as UII), manipulate Utility Auxiliary File (hereafter referred to as UAF) and adjust the settings for target utility without asking AMI to rebuild a new program. You can perform various actions using the following configuration tabs:

- Utility Identification Information (UII)
- UAF Configuration
-



Utility Identification Information (UII)

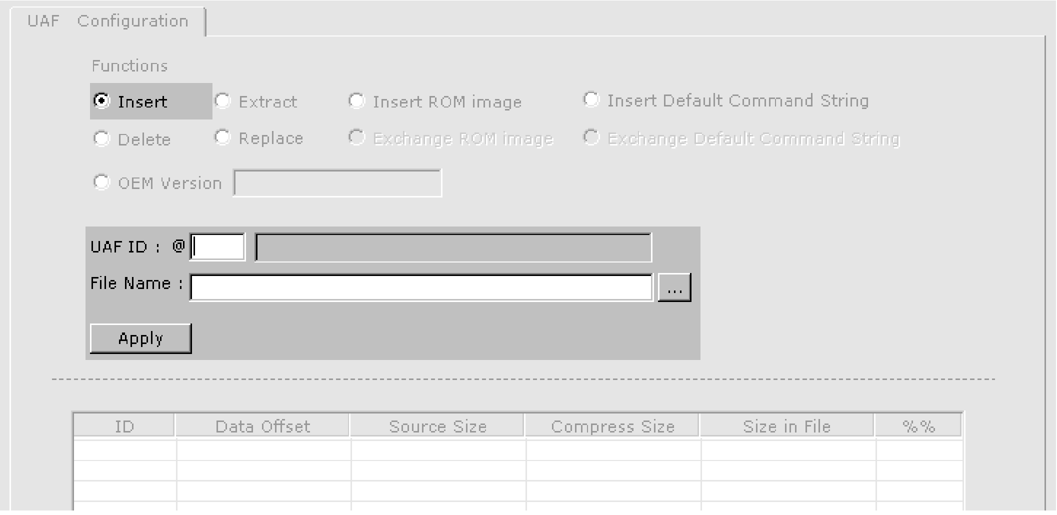
Utility Version : 4.33
Support BIOS : CORE8
Support OS : Windows
Data Bus Width : 32 Bits
Program Type : Executable File
Program Mode : Console/GUI
Release Number : 00

Program Description :
AMI Flash Utility for Windows
Command _GUI mode.

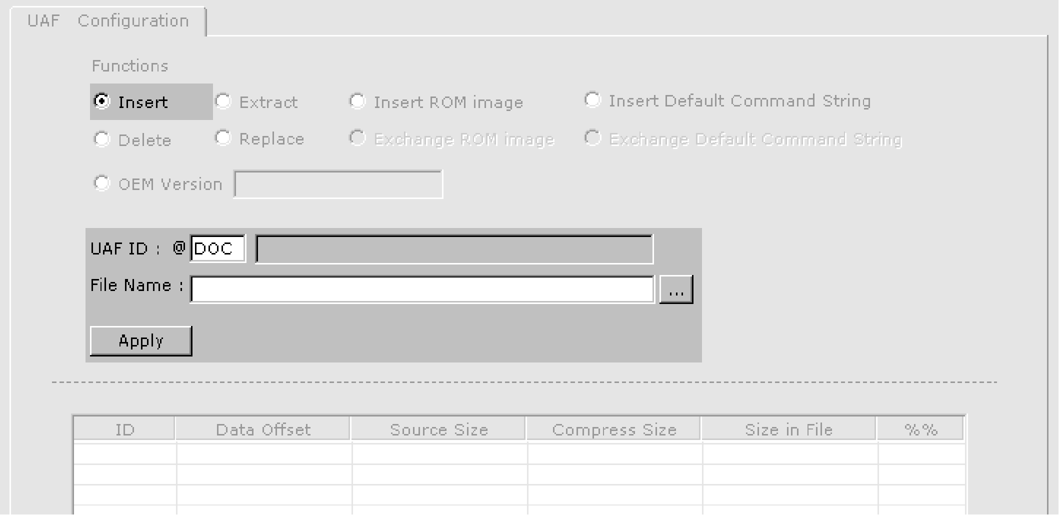
Insert

You can insert a new UAF into target program by following steps:

1. Enable *Insert* at Functions block.

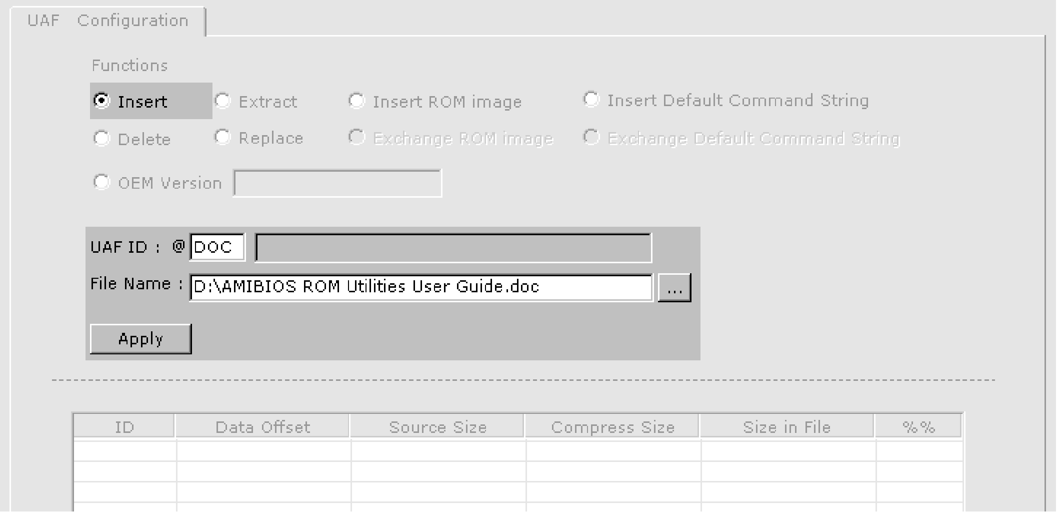


2. Type the new UAF ID into *UAF ID: @* field.



UAF ID : @DOC

3. Click ... button to select input file location Or type the path and the file name in the *File Name* field.



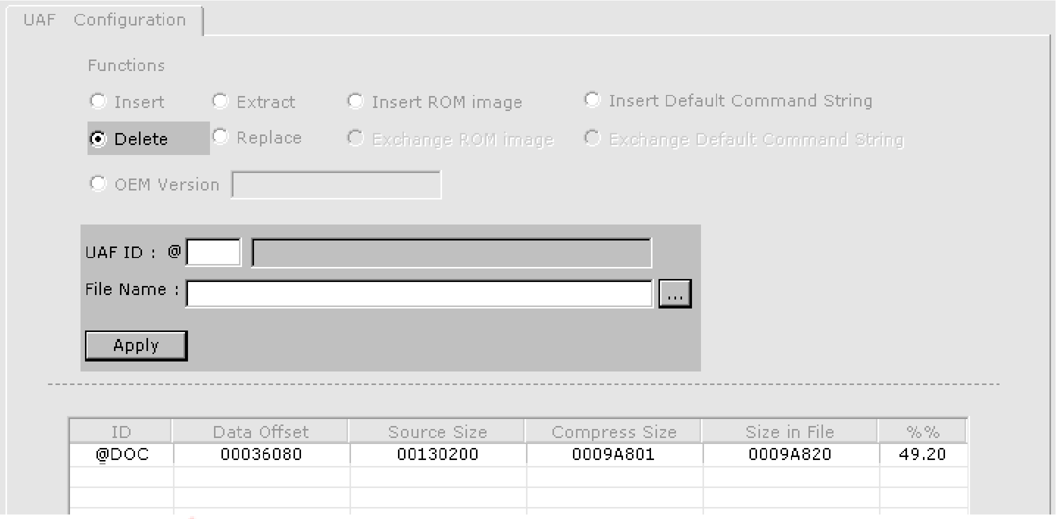
UAF ID : @DOC
File Name : D:\AMIBIOS ROM Utilities User Guide.doc

4. Click Apply button to insert target input file.

Delete

You can delete a UAF from target program by following steps:

1. Enable *Delete* at Functions block.

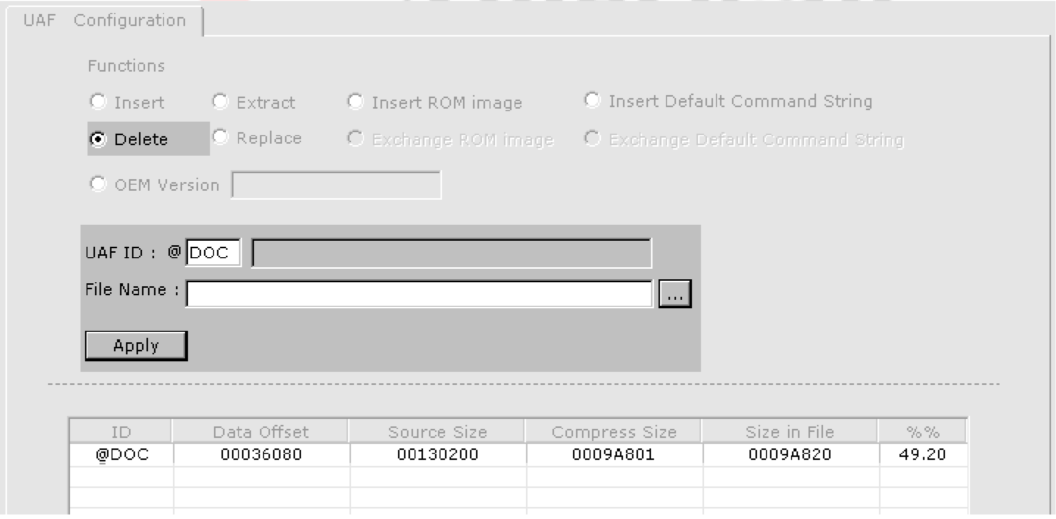


○ Insert ○ Extract ○ Insert ROM image ○ Insert Default Command String
● Delete ○ Replace ○ Exchange ROM image ○ Exchange Default Command String
○ OEM Version

UAF ID : @
File Name :

ID	Data Offset	Source Size	Compress Size	Size in File	%%
@DOC	00036080	00130200	0009A801	0009A820	49.20

2. Type the UAF ID into *UAF ID: @* field.



○ Insert ○ Extract ○ Insert ROM image ○ Insert Default Command String
● Delete ○ Replace ○ Exchange ROM image ○ Exchange Default Command String
○ OEM Version

UAF ID : @DOC
File Name :

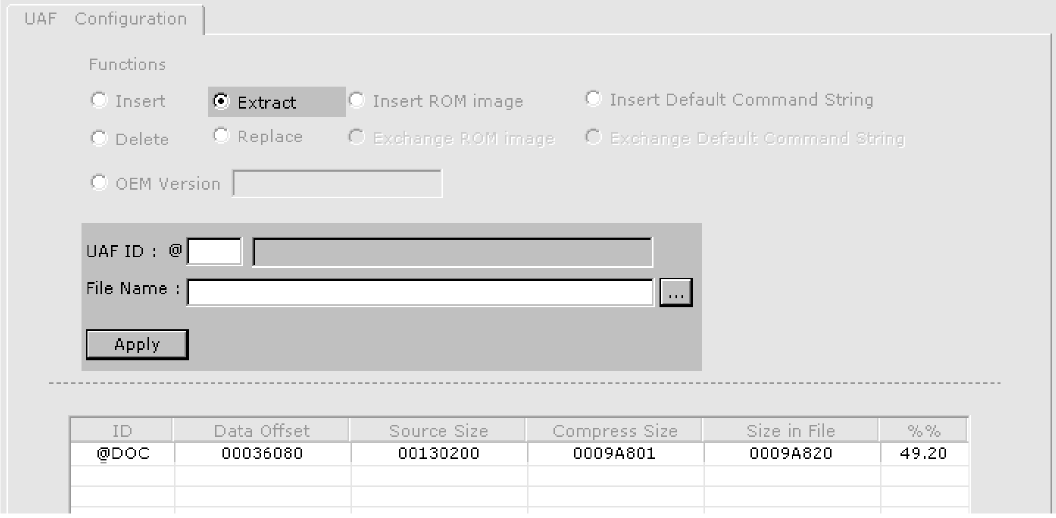
ID	Data Offset	Source Size	Compress Size	Size in File	%%
@DOC	00036080	00130200	0009A801	0009A820	49.20

3. Click Apply button to delete the target UAF.

Extract

You can extract a UAF from target program by following steps:

1. Enable *Extract* at Functions block.



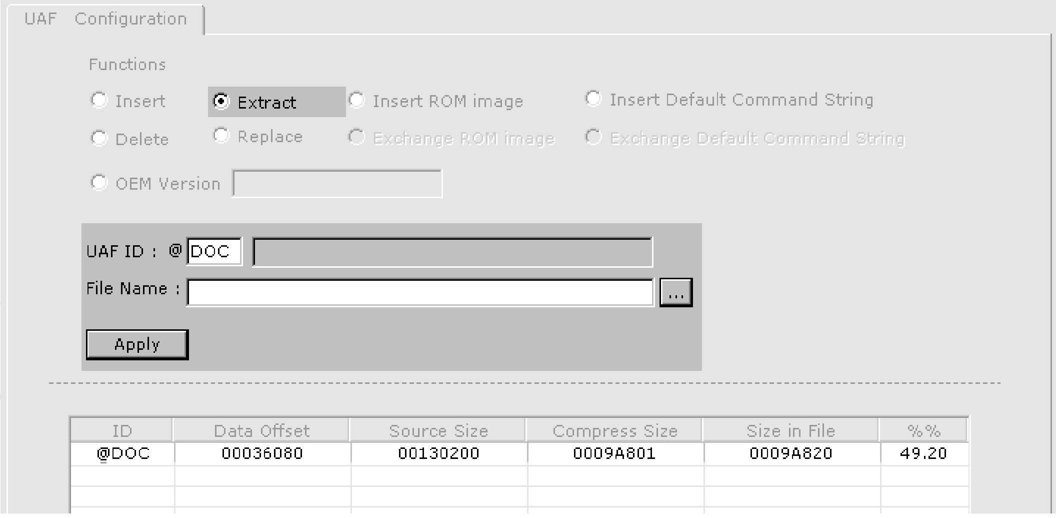
● Extract

UAF ID : @

File Name :

ID	Data Offset	Source Size	Compress Size	Size in File	%%
@DOC	00036080	00130200	0009A801	0009A820	49.20

2. Type the UAF ID into *UAF ID: @* field.



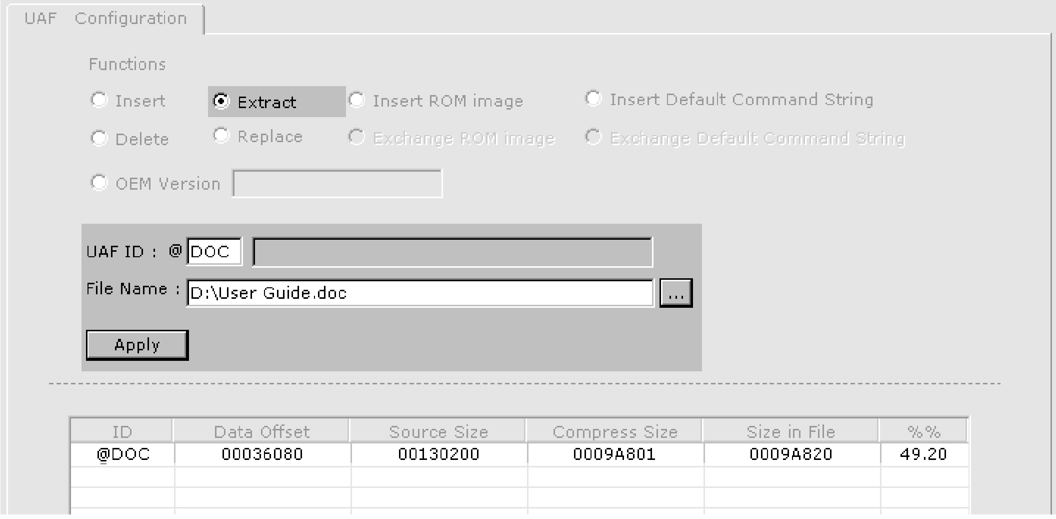
● Extract

UAF ID : @DOC

File Name :

ID	Data Offset	Source Size	Compress Size	Size in File	%%
@DOC	00036080	00130200	0009A801	0009A820	49.20

3. Click ... button to select output file location Or type the path and the file name in the *File Name* field.



● Extract

UAF ID : @DOC

File Name : D:\User Guide.doc

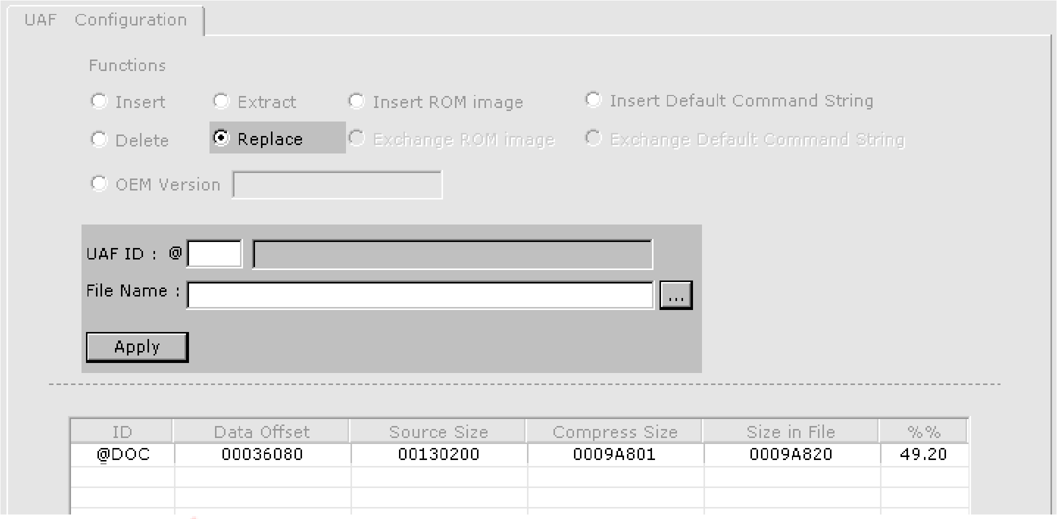
ID	Data Offset	Source Size	Compress Size	Size in File	%%
@DOC	00036080	00130200	0009A801	0009A820	49.20

4. Click Apply button to extract the target UAF to output file.

Replace

You can substitute an existing UAF into target program with a new one by following steps:

1. Enable *Replace* at Functions block.

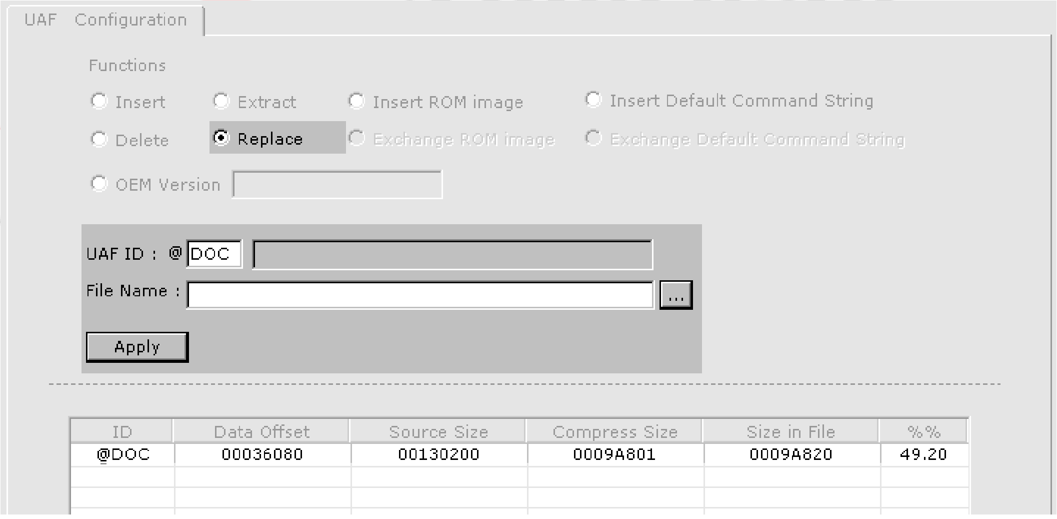


○ Insert ○ Extract ○ Insert ROM image ○ Insert Default Command String
○ Delete ● Replace ○ Exchange ROM image ○ Exchange Default Command String
○ OEM Version

UAF ID : @
File Name :

ID	Data Offset	Source Size	Compress Size	Size in File	%%
@DOC	00036080	00130200	0009A801	0009A820	49.20

2. Type the UAF ID into *UAF ID: @* field.

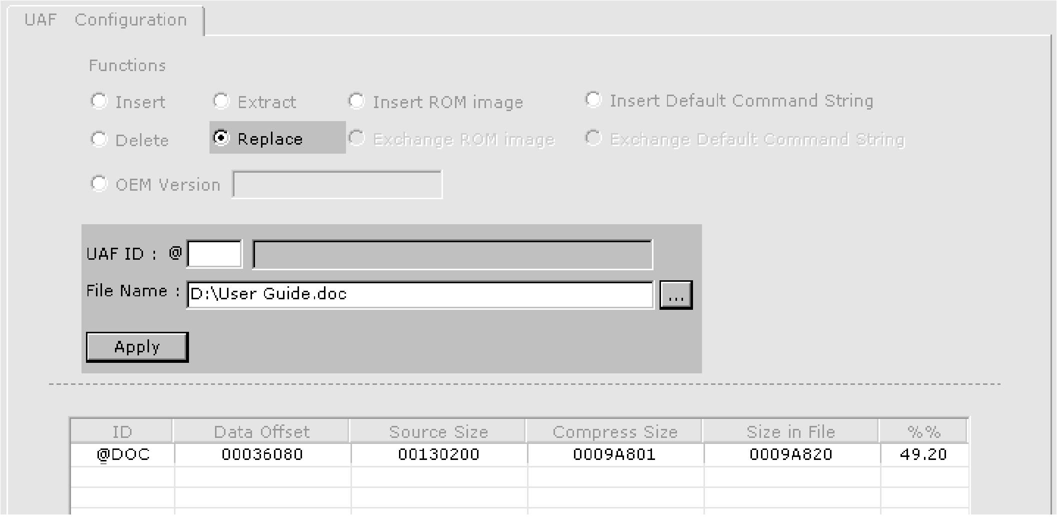


○ Insert ○ Extract ○ Insert ROM image ○ Insert Default Command String
○ Delete ● Replace ○ Exchange ROM image ○ Exchange Default Command String
○ OEM Version

UAF ID : @DOC
File Name :

ID	Data Offset	Source Size	Compress Size	Size in File	%%
@DOC	00036080	00130200	0009A801	0009A820	49.20

3. Click ... button to select input file location Or type the path and the file name in the *File Name* field.



UAF Configuration

Functions

○ Insert ○ Extract ○ Insert ROM image ○ Insert Default Command String

○ Delete ● Replace ○ Exchange ROM image ○ Exchange Default Command String

○ OEM Version

UAF ID : @

File Name : D:\User Guide.doc

Apply

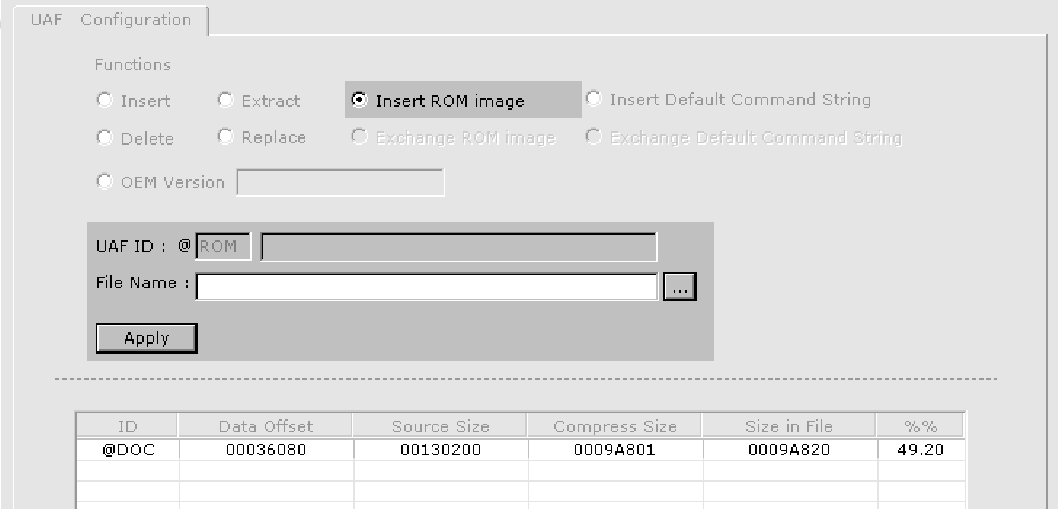
ID	Data Offset	Source Size	Compress Size	Size in File	%%
@DOC	00036080	00130200	0009A801	0009A820	49.20

4. Click Apply button to replace the existing UAF with new input file. The new input file will be inserted into target program.

Insert ROM image

You can insert the @ROM UAF for specify ROM image into target program by following steps:

1. Enable *Insert ROM image* at Functions block.



UAF Configuration

Functions

○ Insert ○ Extract ● Insert ROM image ○ Insert Default Command String

○ Delete ○ Replace ○ Exchange ROM image ○ Exchange Default Command String

○ OEM Version

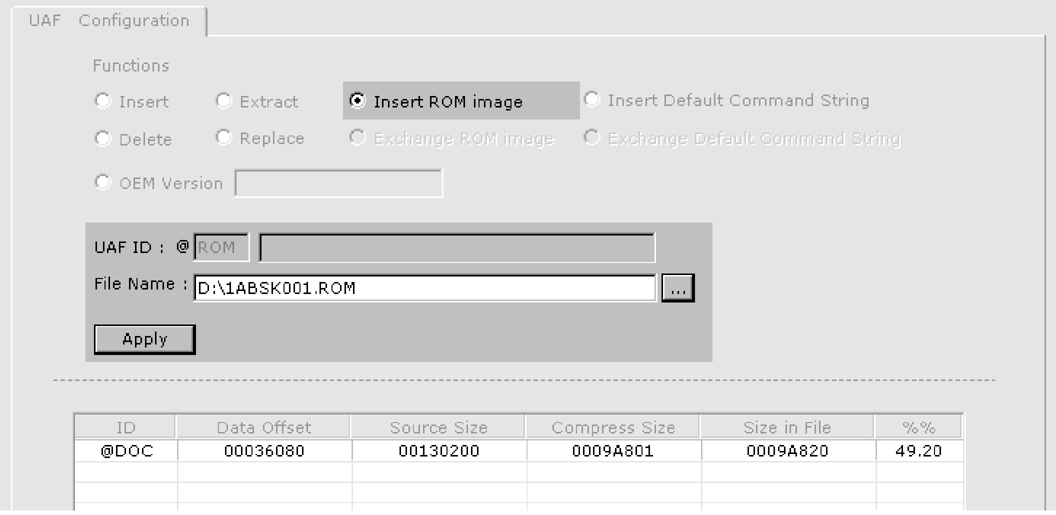
UAF ID : @ROM

File Name :

Apply

ID	Data Offset	Source Size	Compress Size	Size in File	%%
@DOC	00036080	00130200	0009A801	0009A820	49.20

2. Click ... button to select ROM file location Or type the path and the ROM file name in the *File Name* field.



UAF Configuration

Functions

○ Insert ○ Extract ● Insert ROM image ○ Insert Default Command String

○ Delete ○ Replace ○ Exchange ROM image ○ Exchange Default Command String

○ OEM Version

UAF ID : @ROM

File Name : D:\1ABSK001.ROM

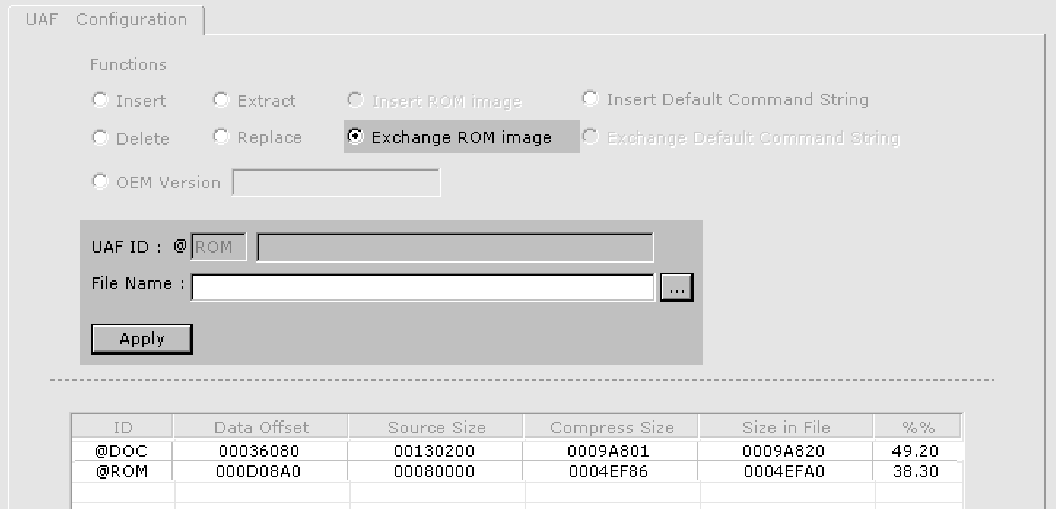
ID	Data Offset	Source Size	Compress Size	Size in File	%%
@DOC	00036080	00130200	0009A801	0009A820	49.20

3. Click Apply button to insert ROM file into target program.

Exchange ROM image

You can substitute the existing @ROM UAF into target program with a new one by following steps:

1. Enable *Exchange ROM image* at Functions block.



UAF Configuration

Functions

○ Insert ○ Extract ○ Insert ROM image ○ Insert Default Command String

○ Delete ○ Replace ● Exchange ROM image ○ Exchange Default Command String

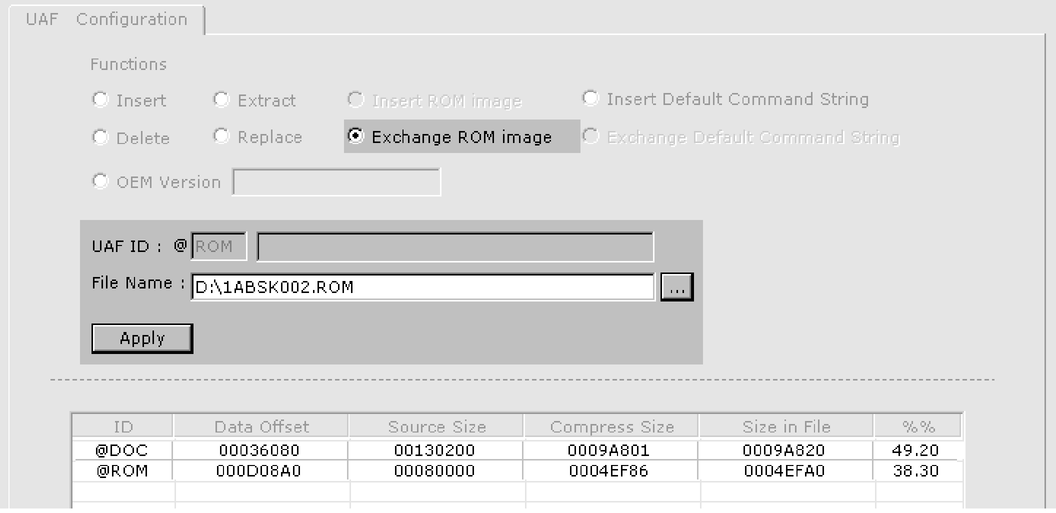
○ OEM Version

UAF ID : @ROM

File Name :

ID	Data Offset	Source Size	Compress Size	Size in File	%%
@DOC	00036080	00130200	0009A801	0009A820	49.20
@ROM	000D08A0	00080000	0004EF86	0004EFA0	38.30

2. Click ... button to select ROM file location Or type the path and the ROM file name in the *File Name* field.



UAF Configuration

Functions

○ Insert ○ Extract ○ Insert ROM image ○ Insert Default Command String

○ Delete ○ Replace ● Exchange ROM image ○ Exchange Default Command String

○ OEM Version

UAF ID : @ROM

File Name : D:\1ABSK002.ROM

Apply

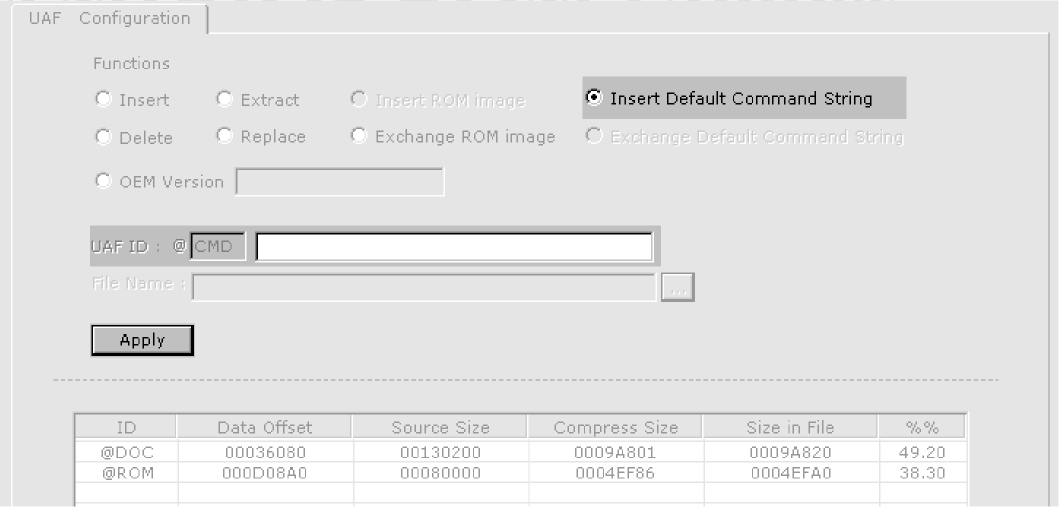
ID	Data Offset	Source Size	Compress Size	Size in File	%%
@DOC	00036080	00130200	0009A801	0009A820	49.20
@ROM	000D08A0	00080000	0004EF86	0004EFA0	38.30

3. Click Apply button to replace the existing @ROM UAF with new ROM file. The new ROM file will be inserted into target program.

Insert Default Command String

You can insert the @CMD UAF for specify Default Command String into target program by following steps:

1. Enable *Insert Default Command String* at Functions block.



UAF Configuration

Functions

○ Insert ○ Extract ○ Insert ROM image ● Insert Default Command String

○ Delete ○ Replace ○ Exchange ROM image ○ Exchange Default Command String

○ OEM Version

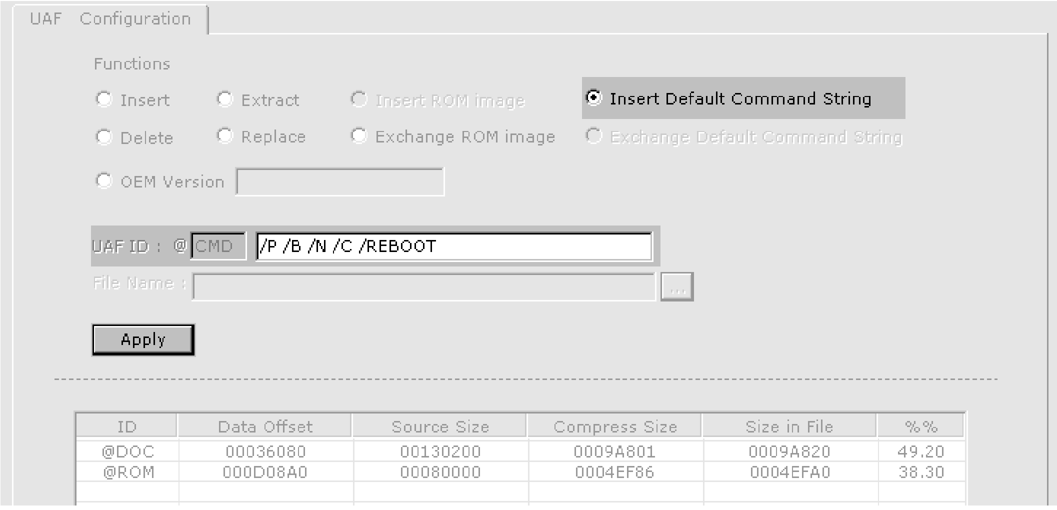
UAF ID : @CMD

File Name :

Apply

ID	Data Offset	Source Size	Compress Size	Size in File	%%
@DOC	00036080	00130200	0009A801	0009A820	49.20
@ROM	000D08A0	00080000	0004EF86	0004EFA0	38.30

2. Enter the Default Command String in the *UAF ID: @CMD* field.



UAF Configuration

Functions

○ Insert ○ Extract ○ Insert ROM image ● Insert Default Command String

○ Delete ○ Replace ○ Exchange ROM image ○ Exchange Default Command String

○ OEM Version

UAF ID : @ CMD /P /B /N /C /REBOOT

File Name :

Apply

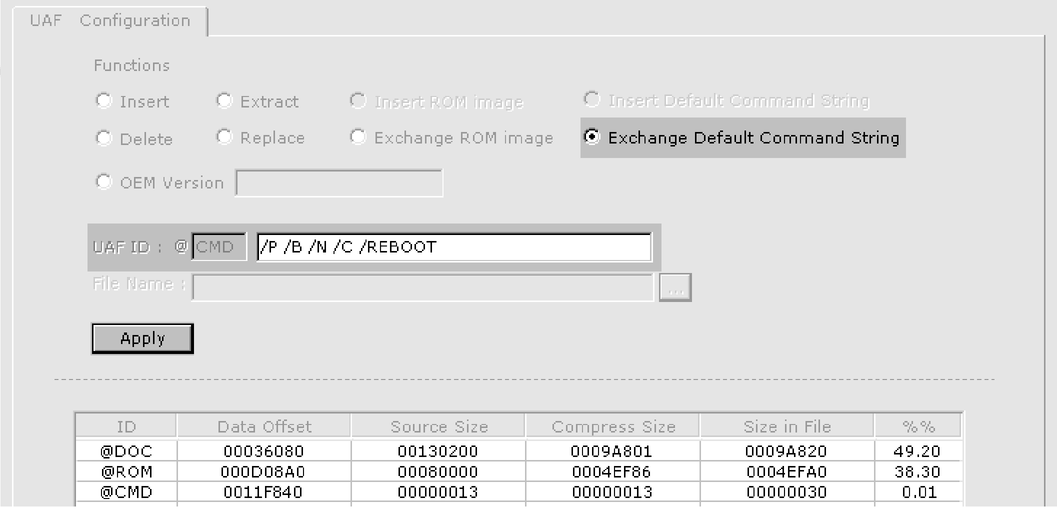
ID	Data Offset	Source Size	Compress Size	Size in File	%%
@DOC	00036080	00130200	0009A801	0009A820	49.20
@ROM	000D08A0	00080000	0004EF86	0004EFA0	38.30

3. Click Apply button to insert Default Command String into target program.

Exchange Default Command String

You can substitute the existing @CMD UAF into target program with a new one by following steps:

1. Enable *Exchange Default Command String* at Functions block.



UAF Configuration

Functions

○ Insert ○ Extract ○ Insert ROM image ○ Insert Default Command String

○ Delete ○ Replace ○ Exchange ROM image ● Exchange Default Command String

○ OEM Version

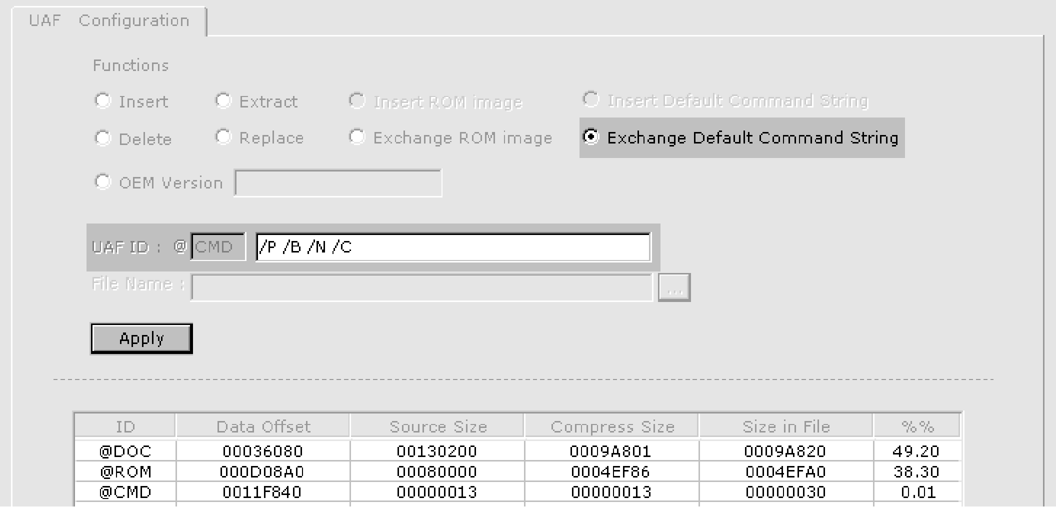
UAF ID : @ CMD /P /B /N /C /REBOOT

File Name :

Apply

ID	Data Offset	Source Size	Compress Size	Size in File	%%
@DOC	00036080	00130200	0009A801	0009A820	49.20
@ROM	000D08A0	00080000	0004EF86	0004EFA0	38.30
@CMD	0011F840	00000013	00000013	00000030	0.01

2. Modify the Default Command String in the *UAF ID: @CMD* field.



○ Insert ○ Extract ○ Insert ROM image ○ Insert Default Command String
○ Delete ○ Replace ○ Exchange ROM image ● Exchange Default Command String
○ OEM Version

UAF ID : @ CMD /P /B /N /C
File Name :

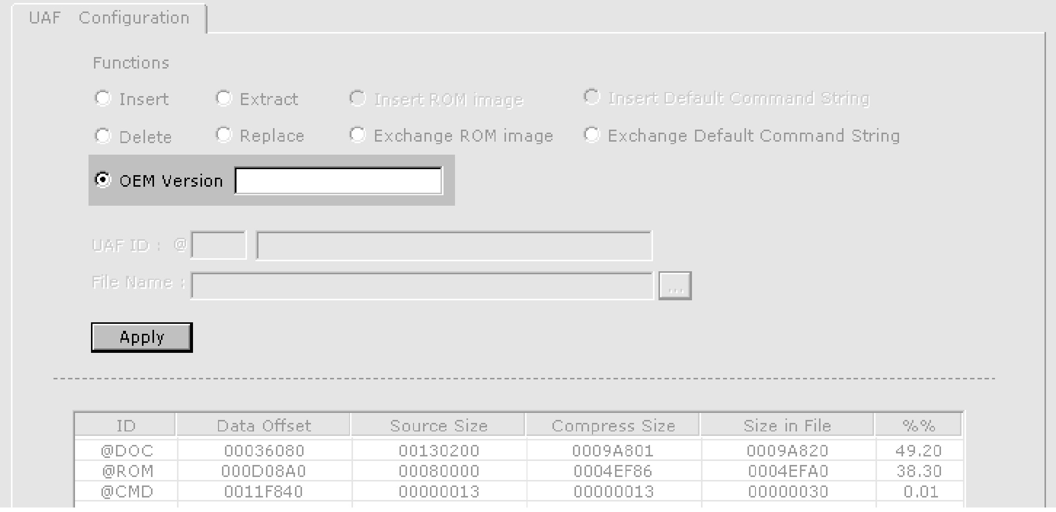
ID	Data Offset	Source Size	Compress Size	Size in File	%%
@DOC	00036080	00130200	0009A801	0009A820	49.20
@ROM	000D08A0	00080000	0004EF86	0004EFA0	38.30
@CMD	0011F840	00000013	00000013	00000030	0.01

3. Click Apply button to replace the existing @CMD UAF with Default Command String. The Default Command String will be inserted into target program.

OEM Version

You can insert the OEM Version into target program by following steps:

1. Enable *OEM Version* at Functions block.

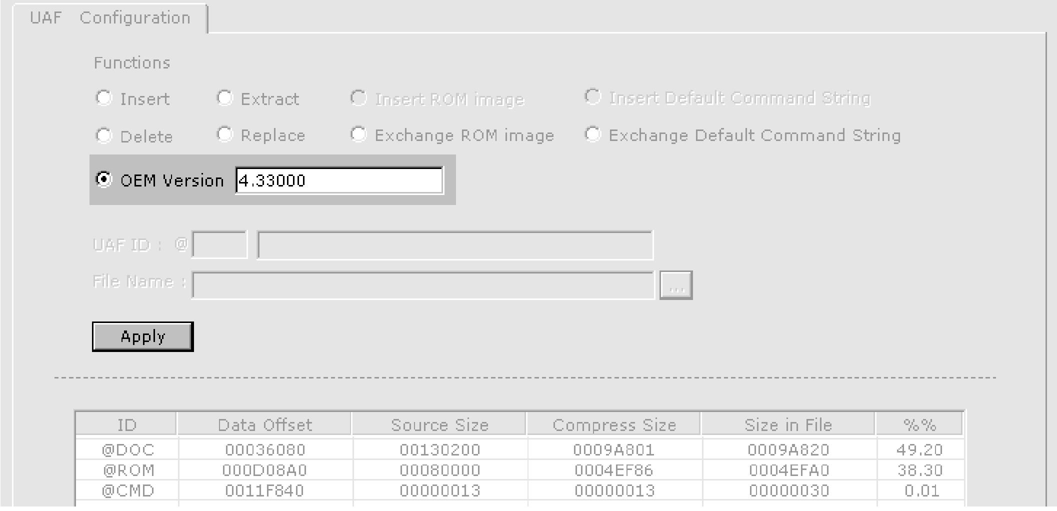


○ Insert ○ Extract ○ Insert ROM image ○ Insert Default Command String
○ Delete ○ Replace ○ Exchange ROM image ○ Exchange Default Command String
● OEM Version

UAF ID : @
File Name :

ID	Data Offset	Source Size	Compress Size	Size in File	%%
@DOC	00036080	00130200	0009A801	0009A820	49.20
@ROM	000D08A0	00080000	0004EF86	0004EFA0	38.30
@CMD	0011F840	00000013	00000013	00000030	0.01

2. Enter the value in the *OEM Version* field.



○ Insert ○ Extract ○ Insert ROM image ○ Insert Default Command String

○ Delete ○ Replace ○ Exchange ROM image ○ Exchange Default Command String

● OEM Version 4.33000

UAF ID : @

File Name :

ID	Data Offset	Source Size	Compress Size	Size in File	%%
@DOC	00036080	00130200	0009A801	0009A820	49.20
@ROM	000D08A0	00080000	0004EF86	0004EFA0	38.30
@CMD	0011F840	00000013	00000013	00000030	0.01

3. Click Apply button to enter OEM Version into target program.

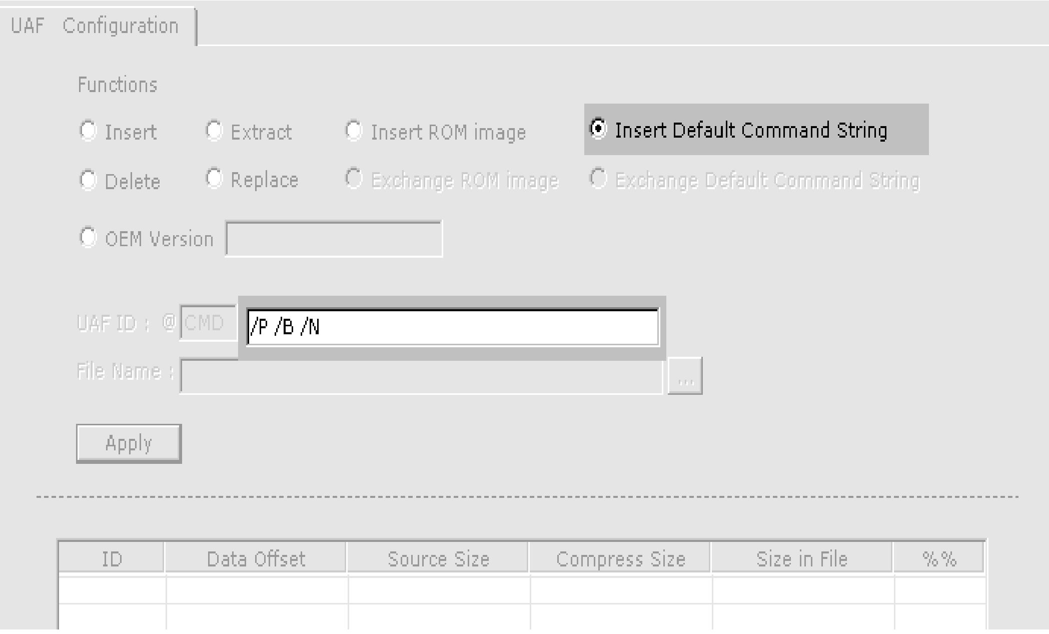
AFU & AMIUCP Configuration Guide

This section will demonstrate how AFU Utilities support AMIUCP standard in following fields.

- Default Command Configuration.
- ROM Image Insertion
- OEM version Control.

Default Command Configuration

In UAF Configuration area, click “Insert Default Command String” then you can set launch up default commands and fill out options in edit box. Please see below figure for usage.



UAF | Configuration

Functions

○ Insert ○ Extract ○ Insert ROM image ● Insert Default Command String

○ Delete ○ Replace ○ Exchange ROM image ○ Exchange Default Command String

○ OEM Version

UAF ID : @CMD /P /B /N

File Name : ...

Apply

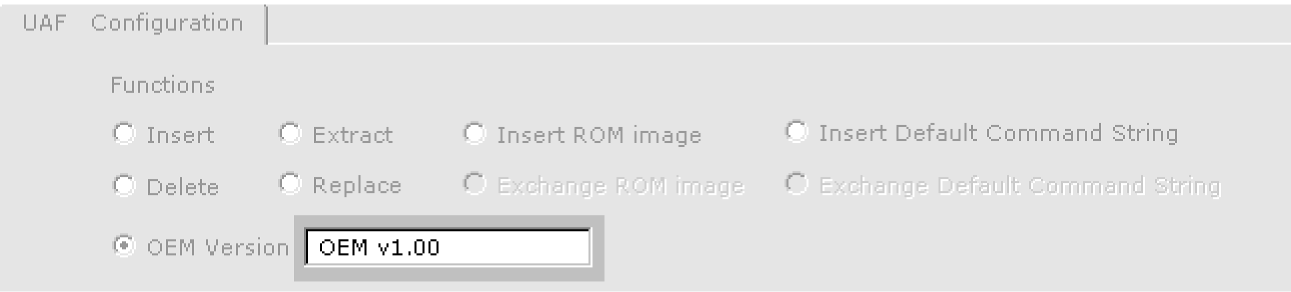
ID	Data Offset	Source Size	Compress Size	Size in File	%%

ROM Image Insertion

Rom file insertion function allows OEM to release click once executable to customers. AFU Utilities will detect executable contains embedded ROM or not. If embedded ROM existed then AFU will issued command according to default command settings. Please see “Exchange ROM Image” to know the usage.

OEM Version Control

AFU Utilities allows OEM to have their version and extended information control. String put in OEM Version edit box will be displayed right after AFU release version. Eg. “AFUWIN 4.xx. OEM v1.00”.



UAF | Configuration

Functions

○ Insert ○ Extract ○ Insert ROM image ○ Insert Default Command String

○ Delete ○ Replace ○ Exchange ROM image ○ Exchange Default Command String

● OEM Version OEM v1.00

Appendix A Module ID Codes

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

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

Appendix B AFUDOS v3.xx Commands

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

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

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