



AMI Software Utility User Guide

AptioV AFU User Guide

Document Revision 1.09

Mar 25, 2016



Confidential, NDA Required
Copyright © 2016

American Megatrends, Inc.
5555 Oakbrook Parkway
Suite 200
Norcross, GA 30093 (USA)

All Rights Reserved
Property of American Megatrends, Inc.

Legal

Disclaimer

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

For Additional Information

Call American Megatrends, Inc. at 1-800-828-9264 for additional information.

Limitations of Liability

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

Limited Warranty

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

Trademark and Copyright Acknowledgments

Copyright © 2016 American Megatrends, Inc. All Rights Reserved.

American Megatrends, Inc.
5555 Oakbrook Parkway
Suite 200
Norcross, GA 30093 (USA)

All product names used in this publication are for identification purposes only and are trademarks of their respective companies.

Table of Contents

| | |
|--|-----------|
| AptioV AFU User Guide | 1 |
| Document Revision 1.09 | 1 |
| Mar 25, 2016 | 1 |
| Legal | 2 |
| Table of Contents | 3 |
| Document Information | 5 |
| Purpose | 5 |
| Audience | 5 |
| Change History | 5 |
| Introduction | 6 |
| Overview | 6 |
| AFU APTIO Features | 6 |
| Requirements | 6 |
| <i>Supported Operating System</i> | 6 |
| <i>Firmware Requirements</i> | 7 |
| Getting Started | 8 |
| Installation | 8 |
| AFUAPTIO Operation | 9 |
| Overview | 9 |
| Commands and Options | 10 |
| <i>Usage</i> | 10 |
| <i>Commands</i> | 10 |
| <i>Options</i> | 11 |
| <i>Rules</i> | 12 |
| Usage | 13 |
| Overview..... | 13 |
| AfuEfix64 <Input or Output File Name> [Option 1] [Option 2] ... | 13 |
| AfuEfix64 <Input or Output File Name> <Command>..... | 15 |
| AfuEfix64 <Command> | 15 |
| AfuEfix64 <ROM Hole File Name> <ROM Hole Option>:<ROM Hole GUID> | 16 |
| AfuEfix64 <BIOS ROM File Name> <Option><Number> | 16 |
| AfuEfix64 <Option /A> <OEM Activation Key Bin File Name> | 17 |
| Remarks | 18 |
| Overview | 18 |
| Preserving Setup Setting – /SP..... | 18 |
| Preserving SMBIOS – /R and /Rn | 19 |
| Programming NVRAM Region – /N | 19 |

| | |
|---|-----------|
| Programming Specific NCB Block – /Kn | 19 |
| Programming Specific ROM Hole – /Ln | 19 |
| Secured Flash Update – /CAPSULE and /RECOVERY | 20 |
| Send special command to BIOS – /CMD:{xxx} | 21 |
| Support Table..... | 22 |
| Command/Option Support in Each Mode | 22 |
| Error Codes | 24 |
| Error Code Definition | 24 |

Document Information

Purpose

This document provides information to use the AptioV AFU for updating system BIOS.

Audience

Generic BIOS Engineers, OEM Engineers, and Aptio Customers.

Change History

| Date | Revision | Description |
|------------|----------|--|
| 2013-11-19 | 1.00 | Initial document created and update content to latest released of Afu |
| 2014-03-20 | 1.01 | Modify 0x18 、0xB6 、0xB7 、0xBF 、0xD0 error message text. |
| 2014-04-24 | 1.02 | Add error message 0x34 、0x35. |
| 2014-08-22 | 1.03 | Add new commands for /meul and /JBC. Need to be updated the SmiFlash module to 5.001_SmiFlash_13. |
| 2014-11-11 | 1.04 | Add Windows 2012 R2 in support list. |
| 2015-01-30 | 1.05 | Add new's command /cmd:. |
| 2015-06-22 | 1.06 | Remove Microsoft®DOS support. |
| 2015-11-18 | 1.07 | Add Linux Xen note. |
| 2016-02-02 | 1.08 | Add 0x4A 、0x4B error message text. Add DOS not support note. |
| 2016-03-25 | 1.09 | Add an announcement: Linux does not support Secure Boot |

Introduction

Overview

AFU (AMI Firmware Update) is a package of utilities used to update the system BIOS under various operating systems. AFU only works for APTIO with SMI FLASH support.

AFU APTIO Features

This list of features is supported by command line, command prompt, EFI Shell, or BSD/Linux shell.

- Read system ROM image
- Flash ROM image
- Command line operating

Requirements

Supported Operating System

AFU is supported by the following operating systems:

- Microsoft® Windows® 2000
- Microsoft® Windows® XP
- Microsoft® Windows® 2003
- Microsoft® Windows® Server 2008 R2
- Microsoft® Windows® Server 2012 R2
- Microsoft® Windows® Vista
- Microsoft® Windows® 7
- Microsoft® Windows® 8
- Microsoft® Windows® 8.1
- Microsoft® Windows® 10
- Microsoft® Windows® PE
- EFI Shell Environment
- BSD
- Linux(*1)(*3)
- MS-DOS(*2)

Note:

- *1. On Linux Xen environment, AFULNX must be executed in host desktop (Domain 0) of virtual machine.
- *2. DOS version is stopped supporting in AFU 5.08 or later version.
- *3. AFU does not support Linux of Secure Boot. If any further requirement, please contact us directly.

Firmware Requirements

- Compatible with AptioV.
- Requires that the current installed firmware has SMI flashing support enabled.
- For supporting Secure Flash, the following eModules are required:
 - *SecureFlash/NIST SP800-147 (5.004_SecureFlash_06 or later)*
 - *AmiCryptoPkg (5_004.CryptoPkg_010 or later)*
 - *SMIFlash (5.001_SmiFlash_13 or later)*
 - *On Flash Block Description (APTIO) (5.001_OFBD_03 or later)*
 - *On Flash Block Description (APTIO) - Secure Flash (5.005_OfbdSecureFlash_000 or later)*
- For supporting Linux Xen, the following eModule are required:
 - *Runtime Memory Hole (5.007_RuntimeMemoryHole_02.1 or later)*

Getting Started

Installation

To run, extract all of the files from the folder with the name corresponding to the desired operating system.

AFUAPTIO Operation

Overview

This mostly involves documenting all the SDL tokens and eLinks. This chapter explains the operation of AFUAPTIO.

The AFUAPTIO operation mode includes all of the AFUAPTIO features such as saving current ROM image to file, getting and displaying ROM ID from BIOS ROM file.

An example of AFUEFIX64 that getting and displaying ROM ID from BIOS ROM file command screen are shown below:

```
UEFI Interactive Shell v2.0
EDK II
UEFI v2.40 (American Megatrends, 0x0005000B)
Mapping table
  FS0: Alias(s):F6:
      VenHw(58C518B1-76F3-11D4-BCEA-0080C73C8881)/VenHw(0C95A935-A006-11D4-B
CFA-0080C73C8881,00000000)
  FS1: Alias(s):F7:
      VenHw(58C518B1-76F3-11D4-BCEA-0080C73C8881)/VenHw(0C95A935-A006-11D4-B
CFA-0080C73C8881,01000000)
  BLK0: Alias(s):
      VenHw(58C518B1-76F3-11D4-BCEA-0080C73C8881)/VenHw(0C95A928-A006-11D4-B
CFA-0080C73C8881,00000000)
  BLK1: Alias(s):
      VenHw(58C518B1-76F3-11D4-BCEA-0080C73C8881)/VenHw(0C95A92F-A006-11D4-B
CFA-0080C73C8881,01000000)
Press ESC in 5 seconds to skip startup.nsh or any other key to continue.
Shell> fs0:
FS0:\> AFUEFIX64 BIOS.rom /U_
```

Commands and Options

The following list is to offer you an overview of the commands and options provided by AFUAPTIO. The content can also be found in AFUAPTIO's help information. A more detailed usage of the commands and options will be explained on the next chapter.

Usage

AfuEfix64 <BIOS ROM File Name> [Option 1] [Option 2] ...

Or

AfuEfix64 < Input or Output File Name > <Command>

Or

AfuEfix64 <Command>

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
- /S Refer to Option: /S
- /D Verification test of given ROM File without flashing BIOS.
- /A Refer to Option: /A
- /OAD Refer to Option: /OAD
- /CLNEVNLOG Refer to Option: /CLNEVNLOG

Options

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

- /MEUL: Program ME Entire Firmware Block, which supports Production.BIN and PreProduction.BIN files.
- /Q Silent execution
- /X Do not check ROM ID
- /CAF Compare ROM file's data with Systems is different or not, if not then cancel related update.
- /S Display current system's ROMID
- /JBC Don't Check AC adapter and battery.
- /HOLEOUT: Save specific ROM Hole according to given RomHole GUID.
- /SP Preserve Setup setting.
- /R Preserve all SMBIOS structures during programming.
- /Rn Preserve SMBIOS type N during programming.(n=0-255)
- /B Program Boot Block
- /P Program main bios image
- /N Program NVRAM
- /K Program all non-critical blocks
- /Kn Program n'th non-critical block (n=0-15)
- /HOLE: Update specific ROM Hole according to RomHole GUID.
- /L Program all ROM Holes
- /Ln Program n'th ROM Hole only (n=0-15)
- /ECUF Update EC BIOS when newer version is detected.
- /E Program Embedded Controller block
- /ME Program ME Entire Firmware Block.
- /MEUF Program ME Ignition Firmware Block.
- /A OEM Activation file.
- /OAD Delete OEM Activation Key
- /CLNEVNLOG Clear Event Log.

- /CAPSULE Override Secure Flash policy by Capsule
- /RECOVERY Override Secure Flash policy by Recovery
- /EC Program Embedded Controller Block. (Flash Type)
- /CMD: Send special command to BIOS. /CMD:{xxx}
- /REBOOT Reboot after programming.
- /SHUTDOWN Shutdown after programming.
- /FDR Flash Flash-Descriptor Region. (*1)
- /GBER Flash GBE Region. (*1)
- /MER Flash Entire ME Region. (*1)
- /OPR Flash Operation Region of SPS. (*1)
- /PDR Flash PDR Region. (*1)

Note:

*1: If BIOS ME Module reports these commands, AFU will show this command.

Rules

- Any parameter enclosed by < > is a mandatory field.
- Any parameter enclosed by [] is an optional field.
- <Commands> cannot co-exist with any [Options].
- Main BIOS image is default flashing area if no any option present.
- [/REBOOT], [/X], and [/S] will enable [/P] function automatically.
- If [/B] present alone, there is only the Boot Block area to be updated.
- If [/N] present alone, there is only the NVRAM area to be updated.
- If [/E] present alone, there is only the Embedded Controller block to be updated.

Overview

The AFUAPTIO offers the following basic command and option usages:

- AfuEfix64 <Input or Output File Name> [Option 1] [Option 2] ...
- AfuEfix64 <Input or Output File Name> <Command>
- AfuEfix64 <Command>

Other usages which are not mentioned in help are:

- AfuEfix64 <ROM Hole File Name> <ROM Hole Option>:<ROM Hole GUID>
- AfuEfix64 <BIOS ROM File Name> <Option><Number>
- AfuEfix64 <Option /A> <OEM Activation Key Bin File Name>

These usages are explained in more detail on this chapter.

AfuEfix64 <Input or Output File Name> [Option 1] [Option 2] ...

User could put no option or combine multiple options in one command line. Commands cannot be combined in command line like options unless the command is categorized as both a command and an option, such as /S and /A.

For option combination case, AFUAPTIO will check its option priority list and execute the options according to the priority order. Three examples of this usage are provided below.

AfuEfix64 < BIOS ROM File Name >

Where BIOS ROM File Name, the mandatory field is used to specify path/filename of the BIOS ROM file with extension. This command line would trigger AFUAPTIO to run the default setting which flashes the system Main Block with the specified BIOS ROM File.

AfuEfix64 <Output BIOS ROM File Name > /D /S

Where Output BIOS ROM File Name, the mandatory field is used to specify path/filename of the BIOS ROM file with extension. /D is to verify the current BIOS and the BIOS ROM File, and /S, which is categorized as a command and also an option, gets and displays the current system's ROM ID.

AfuEfix64 <Output BIOS ROM File Name > /P /B /N /REBOOT

Where BIOS ROM File Name, the mandatory field is used to specify path/filename of the BIOS ROM file with extension. This command line is to flashing current BIOS by BIOS ROM file. /P /B /N are to specify that the flashing regions are Main Block, Boot Block and NVRAM. /REBOOT is to specify that reboot action will be performed in this execution. AFUAPTIO would execute the options in the order of /B, /P, /N and then reboot the system at the end. The order of execution is determined by AFUAPTIO design.

AfuEfix64 <ME File Name > /ME

Where ME File Name is used to specify path/filename of the ME file with extension. This command line programs entire ME block with the specified ME file.

AfuEfix64 <Input or Output File Name> <Command>

AFUAPTIO can only execute one command at a time and it does not accept combinations of command and option in one command line except those can be both command and option. Three examples of this usage are provided below.

AfuEfix64 <Output BIOS ROM File Name> /O

Where BIOS ROM File Name, the mandatory field is used to specify path/filename of the BIOS ROM file with extension. This command line saves the current ROM image to a file.

AfuEfix64 <Output BIOS ROM File Name> /U

Where BIOS ROM File Name is used to specify path/filename of the BIOS ROM file with extension. This command line gets and displays the ROM ID from the specified BIOS ROM file.

AfuEfix64 <Command>

This command usage is for some commands which do not require inputting any file to complete the execution. Usually this type of commands accesses the current BIOS only. An example of this usage is provided:

AfuEfix64 /S

This command line gets and displays the ROM ID of the current BIOS in system.

AfuEfix64 <ROM Hole File Name> <ROM Hole Option>:<ROM Hole GUID>

This command usage is for outputting or flashing a certain ROM hole. For example, the command line for outputting a certain ROM hole whose GUID is 01234567- 89ab- cdef- 0123- 456789abcdef is as following:

AfuEfix64 <Output ROM Hole File Name> /HOLEOUT:0123456789abcdef0123456789abcdef

Where Output ROM Hole File Name is used to specify path/filename of the output ROM hole file with extension. The GUID after the option should not contain dashes or spaces in between.

Another example of flashing a certain ROM Hole whose GUID is 01234567- 89ab- cdef- 0123- 456789abcdef is as following:

AfuEfix64 <ROM Hole File Name> /HOLE:0123456789abcdef0123456789abcdef

Where ROM Hole File Name is used to specify path/filename of the ROM hole file with extension. Please discard dashes and spaces inside GUID line while typing.

AfuEfix64 <BIOS ROM File Name> <Option><Number>

This command usage is for /Kn and /Ln commands where n is indicating the numeric order of a certain no-critical block or ROM hole. For example, to program the 4th ROM hole, the command line could be:

AfuEfix64 <BIOS ROM File Name> /L4

Where BIOS ROM File Name is used to specify path/filename of the BIOS ROM file with extension, and 4 is to specify that the 4th ROM hole is the one to perform /L operation.

The next chapter has more detail of the numbering rule of non-critical blocks and ROM holes.

AfuEfix64 <Option /A> <OEM Activation Key Bin File Name>

This command usage is for /A command which insert a specific OEM activation key into the empty key inside current system BIOS. The command line is as follows:

AfuEfix64 /A <OEM Activation Key Bin File Name>

Where OEM Activation Key Bin File Name is used to specify path/filename of the OEM activation key file with extension. Please make sure that the OEM Activation Key region is empty before inserting the key, or please perform /OAD command before insertion.

Overview

This chapter is to describe commands/options which require extra attention and to explain cases which may occur in certain unique scenarios.

Preserving Setup Setting – /SP

/SP command is designed specifically for “OEM NVRAM/Setup Variable Preserve” module part of OFBD. If /SP is called, AFUAPTIO would send SMI 0x26 twice to save setup setting before starting updating NVRAM and to restore setup setting after finishing updating NVRAM. Customer can customize their OFBD module to preserve certain NVRAM data when AFUAPTIO flashes the NVRAM area. For example, there are two methods for preserving Setup Password:

Method 1

Enable PRESERVE_PASSWORDS token – The BIOS will preserve its Setup password when AFUAPTIO calls the SMIFlash module.

Method 2

Control through /SP command – Customer can port PreserveSetupPassword in OFBDSETUPStoreHandle and RestoreSetupPassword in OFBDSETUPRestoreHandle, and use /SP command to keep or not to keep the Setup Password while updating the NVRAM:

AfuEfix64 xxx.ROM /N /SP - keep Setup password

AfuEfix64 xxx.ROM /N - don't keep Setup password.

This feature needs BIOS' cooperation. To learn more about preserving setup data, please consult with your BIOS provider.

Preserving SMBIOS – /R and /Rn

If the SMBIOS data is stored in Main Block or Boot Block, AFUAPTIO /R and /Rn options would take the responsibility to preserve the SMBIOS data. If the SMBIOS data is stored in NVRAM and BIOS project's token SMBIOS_PRESERVE_NVRAM=0, the preservation process would take place at OFBD module. To know more about the detail of preserved data, please consult with your BIOS provider.

/R is used to preserve the whole SMBIOS data. To preserve a certain type of SMBIOS, please use /Rn. For example, to preserve SMBIOS Type 2 and Type 41 during BIOS flashing and the SMBIOS data is located in Boot Block, the command could be:

AfuEfix64 <BIOS ROM File Name> /B /R2 /R41

Programming NVRAM Region – /N

Erasing NVRAM may cause important variables lose.

Programming Specific NCB Block – /Kn

/Kn command is designed to program a specific non-critical block, or NCB block. AFUAPTIO would search ROM and identify the first NCB Block found as K0, and the second one as K1, etc. Therefore, command /K2 would program the third NCB Block found by AFU.

Programming Specific ROM Hole – /Ln

/Ln command is designed to program a specific ROM Hole. Each ROM Hole is identified in the following way: AFUAPTIO would search for ROM Holes in the order of Boot Block area and Main Block area, and identify each ROM Hole in consecutive integers from 0 to 15. So, for example, /L1 is used to program the second ROM Hole found in ROM.

Scenarios:

- If a ROM contains two ROM Holes in Boot Block area and two in Main Block area, AFUAPTIO would identify L0 and L1 for the two in Boot Block area and L2 and L3 for the two in Main Block area.

- If a ROM contains 2 ROM Holes in Boot Block area and none in Main Block area, AFUAPTIO would only find 2 ROM Holes in total and identify them as L0 and L1.

- If a ROM contains no ROM Holes in Boot Block area and three in Main Block area, AFUAPTIO would find nothing in Boot Block area and identify L0, L1 and L2 for the three ROM Holes in Main Block area.

Secured Flash Update – /CAPSULE and /RECOVERY

For Secured BIOS, the command rule for programming the current BIOS is different. There are two more modes, Capsule Mode and Recovery Mode, which are different from the regular Runtime Mode mentioned in the previous contents. Unlike Runtime Mode where all the commands/options are supported, Capsule Mode and Recovery Mode only support /P, /B, /N, and /E options, or depending on the BIOS design. The following description explains how to program BIOS under these two modes.

To override Secure Flash policy and program the BIOS image in Capsule Mode, please use the command:

AfuEfix64 <BIOS ROM File Name> /CAPSULE /P /B /N /E

And to override Secure Flash policy and program the BIOS image in Recovery Mode, please use this command:

AfuEfix64 <BIOS ROM File Name> /RECOVERY /P /B /N /E

Where BIOS ROM File Name is used to specify path/filename of the BIOS ROM file with extension.

For more detail on Secure Flash, please consult with your BIOS provider.

Send special command to BIOS – /CMD:{xxx}

Send the string between brackets to OFBD OEM CMD Checking Module. The string is corresponding to the string which is defined in BIOS by user.

Support Table

Command/Option Support in Each Mode

| Command | Runtime Mode | Capsule Mode | Recovery Mode |
|------------|--------------|---------------|---------------|
| /O | Supported | Not Supported | Not Supported |
| /U | Supported | Not Supported | Not Supported |
| /S | Supported | Not Supported | Not Supported |
| /D | Supported | Not Supported | Not Supported |
| /A | Supported | Not Supported | Not Supported |
| /OAD | Supported | Not Supported | Not Supported |
| /CLNEVNLOG | Supported | Not Supported | Not Supported |

| Option | Runtime Mode | Capsule Mode | Recovery Mode |
|-----------|--------------|------------------|---------------|
| /MEUL | Supported | Not Supported | Not Supported |
| /Q | Supported | Not Supported | Not Supported |
| /X | Supported | Not Supported | Not Supported |
| /CAF | Supported | Not Supported | Not Supported |
| /S | Supported | Not Supported | Not Supported |
| /JBC | Supported | Not Supported | Not Supported |
| /SP | Supported | Not Supported | Not Supported |
| /R | Supported | Supported (*1) | Not Supported |
| /Rn | Supported | Supported (*1) | Not Supported |
| /B | Supported | Supported | Supported |
| /P | Supported | Supported | Supported |
| /N | Supported | Supported | Supported |
| /K | Supported | Not Supported | Not Supported |
| /Kn | Supported | Not Supported | Not Supported |
| /HOLE: | Supported | Not Supported | Not Supported |
| /HOLEOUT: | Supported | Not Supported | Not Supported |
| /L | Supported | Not Supported | Not Supported |
| /Ln | Supported | Not Supported | Not Supported |
| /ECUF | Supported | Not Supported | Not Supported |
| /E | Supported | Supported | Supported |
| /ME | Supported | Not Supported | Not Supported |
| /MEUF | Supported | Not Supported | Not Supported |
| /A | Supported | Not Supported | Not Supported |
| /OAD | Supported | Not Supported | Not Supported |

| Option | Runtime Mode | Capsule Mode | Recovery Mode |
|------------|--------------|---------------|---------------|
| /CLNEVNLOG | Supported | Not Supported | Not Supported |
| /EC | Supported | Not Supported | Not Supported |
| /REBOOT | Supported | Not Supported | Not Supported |
| /SHUTDOWN | Supported | Not Supported | Not Supported |

Note:

*** 1: This option must use with either /P or /B in order to be supported under Capsule Mode.**

Error Codes

Error Code Definition

| CODE | Definition |
|------|--|
| 0x01 | Error: Unknown command. |
| 0x02 | Error: BIOS has no flash information available. |
| 0x03 | Error: ROM file size does not match existing BIOS size. |
| 0x04 | Error: ROM file ROMID is not compatible with existing BIOS ROMID. |
| 0x05 | Error: Bootblock error. |
| 0x06 | Error: This BIOS version has more Non-Critical blocks than supported. |
| 0x07 | Error: BIOS checksum error. |
| 0x08 | Error: Invalid option |
| 0x09 | Error: Size of ROM file does not match the size of system ROM |
| 0x0A | Error: Unable to update ROM hole |
| 0x0B | Error: ROMHOLE not exist |
| 0x0C | Error: BIOS update cancelled by user. |
| 0x0D | Error: BIOS Report Error. |
| 0x0E | Error: Kernel source files cannot be found. |
| 0x10 | Error: Unable to load driver. |
| 0x11 | Error: Unable to unload driver. |
| 0x12 | Error: No non-critical blocks found in ROM file. |
| 0x13 | Error: Requested non-critical block not available in ROM file. |
| 0x14 | Error: Non-critical blocks in ROM image file do not match those in the system. |
| 0x15 | Error: Secure Flash function is not supported on this platform. |
| 0x16 | Error: Unable to get Secure Flash policy from BIOS. |
| 0x17 | Error: Unsupported Secure Flash policy. |
| 0x18 | Error: Secure Flash Rom Verify fail. |
| 0x19 | Error: Failed to erase flash chip (at Runtime Secure Flash). |
| 0x1A | Error: Failed to update flash chip (at Runtime Secure Flash). |
| 0x1B | Error: Failed to read flash chip (at Runtime Secure Flash). |
| 0x1C | Error: Failed to verify flash chip (at Runtime Secure Flash). |
| 0x1D | Error: Failed to load image into memory. |
| 0x1E | Error: Secure Flash function is not supported on this file. |
| 0x1F | Error: Reserved for Secure Flash. |
| 0x20 | Error: Unable to initialize memory manager. |
| 0x21 | Error: Unable to close memory manager. |
| 0x22 | Error: Problem allocating memory. |

| CODE | Definition |
|------|---|
| 0x23 | Error: Problem freeing memory. |
| 0x24 | Error: Problem allocating BIOS buffer. |
| 0x25 | Error: Problem freeing BIOS buffer. |
| 0x26 | Error: Problem freeing mapping BIOS. |
| 0x27 | Error: Problem freeing unmapping BIOS. |
| 0x28 | Error: Problem mapping BIOS data. |
| 0x29 | Error: Problem unmapping BIOS data. |
| 0x30 | Error: Problem opening file for reading. |
| 0x31 | Error: Problem reading file. |
| 0x32 | Error: Problem opening file to write. |
| 0x33 | Error: Problem writing file. |
| 0x34 | Error: Using the wrong AFU version, Please use Aptio 4 AFU. |
| 0x35 | Error: Using the wrong AFU version, Please use Aptio 5 AFU. |
| 0x40 | Error: BIOS is write-protected. |
| 0x41 | Error: Can not close flash interface. |
| 0x42 | Error: Problem reading flash. |
| 0x43 | Error: Problem erasing flash. |
| 0x44 | Error: Problem writing flash. |
| 0x45 | Error: Problem verifying flash. |
| 0x46 | Error: Problem getting flash information. |
| 0x47 | Error: No firmware id. |
| 0x48 | Error: Power cord not connected. Plug in power cord to flash. |
| 0x49 | Error: A platform condition has prevented flashing. |
| 0x4A | Error: Platform data is not empty, And data address is not Alignment Block Address. |
| 0x4B | Error: SLP key is not empty at all. |
| 0x50 | Error: This program must be run in MS-DOS mode. |
| 0x60 | Error: Accessing registry. |
| 0x61 | Error: Program already running. |
| 0x70 | Error: BSD access IO. |
| 0x80 | Error: Size of system ROM mismatches size of ROM file |
| 0x81 | Error: ROM ID mismatch |
| 0x82 | Error: Bootblock checksum error |
| 0x90 | Error: Error to shutdown |
| 0x91 | Error: Error to restart... |
| 0x92 | Error: Can't open ROM ID file |
| 0x93 | Error: ROM ID file is not a ROM file. |
| 0x94 | Error: Invalid MAC address |
| 0x95 | Error: Invalid load current CMOS option |
| 0x96 | Error: Invalid retry count |
| 0x97 | Error: Invalid defined ROM ID length |
| 0x98 | Error: Invalid SMI |
| 0x99 | Error: ROM File ID don't exist |
| 0x9A | Error: System ROM ID don't exist |
| 0x9B | Error: Password Retry count exceeded. |
| 0x9C | Error: BIOS don't support NVRAM/SETUP preserve function |

| CODE | Definition |
|------|---|
| 0x9D | Error: Store SETUP setting error |
| 0x9E | Error: Restore SETUP setting error |
| 0x9F | Error: Cannot analyze ROM file. ROM file may be corrupted |
| 0xA0 | Error: Cannot analyze the ME Data. ROM file may be corrupted |
| 0xA1 | Error: BIOS does not support ME Entire Firmware update |
| 0xA2 | Error: BIOS does not support ME Ignition Firmware update |
| 0xA3 | Error: Invalid EC ROM file |
| 0xA4 | Error: EC ROM file checksum error |
| 0xA5 | Error: Can't enter EC flash mode |
| 0xA6 | Error: Erasing EC flash memory fail |
| 0xA7 | Error: Initial EC programming fail |
| 0xA8 | Error: EC flash data transmit error |
| 0xA9 | Error: Writing EC flash memory fail |
| 0xAA | Error: Exit EC programming mode fail |
| 0xAB | Error: ROM Chip ID mismatch |
| 0xAC | Error: Invalid EC Header Table |
| 0xAD | Error: EC does not permit BIOS update |
| 0xAE | Error: BIOS doesn't support OEMCMD function |
| 0xAF | Error: Store DMI Data error |
| 0xB0 | Error: Restore DMI Data error |
| 0xB1 | Error: Invalid Activation Key file. |
| 0xB2 | Error: File Size is greater than image activation key length. |
| 0xB3 | Error: Image activation key larger than BIOS activation key. |
| 0xB4 | Error: Activation Key checksum error. |
| 0xB5 | Error: No Support Activation Key error. |
| 0xB6 | Error: OA key is not empty at all. And OA Key is no the same as Bin File in system. |
| 0xB7 | Error: OA key is empty at all already. |
| 0xB8 | Error: OA key region incorrect. |
| 0xB9 | Error: BIOS doesn't support Clear event log function. |
| 0xBA | Error: Clear event log error. |
| 0xBB | Error: Rom image layout detected RomHole is redesigned. |
| 0xBC | Error: BIOS have more than one RomHole's GUID is the same. |
| 0xBD | Error: Requested Rom Hole not available in ROM file. |
| 0xBE | Error: RomHoles in ROM image file do not match those in the system. |
| 0xBF | Error: OA key is not empty at all. And OA Key is the same as Bin File in system. |
| 0xC0 | Error: BIOS doesn't support process ME information |
| 0xC1 | Error: BIOS return error, when trying to re-flash ME Firmware data. |
| 0xC2 | Error: Region is write-protected |
| 0xC6 | Error: No EC blocks found in system ROM. |
| 0xC7 | Error: BIOS doesn't support all ROM flashing function. |
| 0xD0 | Error: OA key data is invalid. |
| 0xD1 | Error: BIOS has already updated OA. |
| 0xD2 | Error: BIOS does not allow updating OA. |
| 0xD3 | Error: BIOS doesn't support updating OA. |
| 0xD4 | Error: The DMI data size of system is greater than File's DMI data length. |

| CODE | Definition |
|-------------|--|
| 0xD5 | Error: BIOS doesn't support EC Battery Check function. |