



Whitepaper

AMIBIOS8 Flash Update & BIOS Recovery Methods

Version 1.2, Updated June 2, 2005

Copyright (c) 2005 American Megatrends, Inc.
All Rights Reserved.

American Megatrends, Inc.
6145-F, Northbelt Parkway
Norcross, GA - 30071, USA

This publication contains proprietary information which is protected by copyright. No part of this publication 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 BIOS Sales Department at 1-800-828-9264 for additional information.

Limitations of Liability

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

Limited Warranty

No warranties are made, either 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.

Trademark and Copyright Acknowledgments

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

Revision Information

Date	Rev	Description of Changes	Editor
11 June 2003	0.1	Draft Version	B. Richardson
12 June 2003	0.2	List AMIFLASH as 'unsupported' flash tool (Section 2.3). Add Section 3.4 for recovery from ATAPI Removable Media. Add text to Section 3 describing boot block recovery methods.	B. Richardson
13 June 2003	0.3	Add description of WINFLASH with screenshots. Add FLASH827 to Section 2.3. Reformatted Section 2.1. Added footnote describing <CTRL><PGUP>/<PGDN>. Added description of flash via BMC to Section 3.	B. Richardson
16 June 2003	0.4	Update Section 1.1 & 1.4. Added note for 'CR/LF' configuration for Serial Flash (Section 3.5)	B. Richardson
17 June 2003	0.5	Rearrange Section 1. Add footnotes regarding AMI customers & end-users to Section 1. Add better description of 'BIOS Rom image' to Section 1.3.	B. Richardson U. Vezzani
19 June 2003	1.0	First public release. Corrected AFU title & added DOS to list of supported operating systems (Section 2.1). Moved revision information to last page	B. Richardson
2 June 2005	1.2	Reformatted template	U. Vezzani

Table of Contents

1	INTRODUCTION	4
1.1	Why update the BIOS?	4
1.2	Why do I need a specific BIOS for my system? Isn't all BIOS the same?	4
1.3	How is a BIOS upgraded?	5
1.4	Why is it called "flashing"?	5
2	AMIBIOS8 FLASH UPGRADE METHODS	6
2.1	AMI Firmware Update (AFU)	6
2.2	WinFLASH	7
2.3	Existing/Unsupported BIOS Update Tools	8
3	AMIBIOS8 BOOT BLOCK RECOVERY METHODS.....	9
3.1	Boot Block Recovery from Floppy Disk	9
3.2	Boot Block Recovery from IDE CD-ROM	10
3.3	Boot Block Recovery from USB Storage.....	10
3.4	Boot Block Recovery from ATAPI Removable Media.....	11
3.5	Boot Block Recovery via Serial Port ("Serial Flash")	11
3.6	Boot Block Recovery via IPMI Baseboard Management Controller (BMC).....	13

1 Introduction

1.1 Why update the BIOS?

The Basic Input/Output System (BIOS) is software associated with the system motherboard hardware, also referred to as “firmware”. The BIOS is responsible for configuring the system motherboard and related hardware before the operating system loads. BIOS updates are often used by system vendors to correct platform issues discovered after hardware has been shipped to a customer. BIOS can be upgraded to correct any number of problems, including:

- Processor configuration (clock speed, microcode, etc.)
- Compatibility with operating systems
- Support for new IDE/ATAPI or USB storage devices
- Compatibility with new memory modules or add-on devices
- Workarounds for hardware bugs

1.2 Why do I need a specific BIOS for my system? Isn't all BIOS the same?

The system motherboard BIOS is designed specifically for that particular system motherboard. Even if two motherboards use the same types of processors and chipsets, they will most likely require different BIOS code based on their configuration. AMI provides customers¹ with AMIBIOS8 as a development platform to create a customized BIOS solution for their platform. While all BIOS images based on AMIBIOS have some commonality, they are essentially unique to a particular system ... designed as a part of that system.

Flashing the wrong BIOS on your motherboard can void your warranty and result in serious damage to your computer hardware². Each AMIBIOS image is made specifically for a particular system motherboard. If you cannot identify the motherboard manufacturer³, try using the [AMI Motherboard Identification Utility](#) available at [AMI.COM](#).

¹ AMI provides BIOS code to motherboard and system manufacturers. Roughly half of the desktops sold worldwide feature AMIBIOS or a BIOS derived from AMIBIOS.

² You might think we're kidding about the wrong BIOS damaging your system, but we're not. The BIOS controls your hardware configuration, including clock speeds and voltages. The wrong BIOS can improperly set these values, causing the processor to overheat and fail ... but that's an extreme example. In most cases, flashing the wrong BIOS will just turn your computer into a giant beige paperweight ... which is still a bad thing.

³ AMI will directly provide BIOS upgrades for motherboards manufactured by AMI. The BIOS of motherboards from third parties is always serviced by the original motherboard or equipment manufacturer (ASUS will provide BIOS upgrades for ASUS motherboards, NEC will provide BIOS upgrades for NEC notebooks, etc.).

1.3 How is a BIOS upgraded?

There are two AMIBIOS update mechanisms: “flash update” and “BIOS recovery”. Both program a BIOS image, typically delivered as a file with the .ROM extension, into the motherboard’s flash memory.

- The standard “flash update” method uses a software utility, which is separate from the BIOS image. AMI provides a variety of programs for updating the BIOS from the operating system. These utilities feature help text, simple user interfaces, the ability to backup the current BIOS image and BIOS image verification to help prevent the wrong BIOS from being flashed onto the system.
- The “BIOS recovery” scenario is recommended for situations when the traditional flash update fails and the user can no longer boot back to an operating system to restore the system to a functional BIOS image. The code that handles BIOS recovery resides in a section of the flash referred to as “boot block”, which is not reprogrammed during a standard flash update⁴.

Section 2 of this document describes the BIOS update mechanisms available for AMIBIOS8, the latest revision of the AMIBIOS product. Section 3 describes BIOS Recovery mechanisms used by AMIBIOS8. For more information on AMIBIOS8, please refer to the “[Introduction to AMIBIOS8](#)” whitepaper available from [AMI.COM](#). Also check AMI.COM for more information on AMI products.

1.4 Why is it called “flashing”?

The term “flashing” applies to the re-programming of the flash memory used to store the BIOS. Flash memory, also referred to as “flash RAM”, is a type of non-volatile memory that can be erased and reprogrammed without being removed from the computer. Flash memory gets its name because it is organized so that a memory block is erased in a single action or “flash.” This is an improvement over older types of read-only memory (ROM) used to store the BIOS that could only be reprogrammed using a special programmer.

When BIOS needs to be changed, the flash memory can be written to in block sizes, making the update faster than other types of memory that are written to in single byte increments. Since flash memory is non-volatile, the BIOS is retained when the computer power is switched off, unlike system RAM which loses data when power is removed.

⁴ The AMIBIOS boot block can be rewritten during a flash update using advanced features of our software utilities. This is not recommended, since improperly programming the boot block will most likely erase the code used to perform BIOS recovery. Please see Section 3 for a more detailed description of the boot block and AMIBIOS Boot Block Recovery methods.

2 AMIBIOS8 Flash Upgrade Methods

2.1 AMI Firmware Update (AFU)

AMI Firmware Update (AFU) is a package of utilities used to update the system BIOS under various operating systems. Note that AFU only works for AMIBIOS with "SMI FLASH" support, which is a standard AMIBIOS8 feature. AFU is supported under a number of operating systems⁵ ... DOS, Windows 95/98/ME/NT4/2000/XP, Linux & FreeBSD.

```
afudos /i<ROM filename> [/o<save ROM filename>] [/n] [/p[b][n][c]]  
[/r<registry_path>] [/s] [/k[N]] [/q] [/h]
```

```
/n      - don't check ROM ID  
/pbnc  -  
        b - Program Boot Block  
        n - Program NVRAM  
        c - Destroy System CMOS  
/r      - registry path to store result of operation  
        (only for Windows version)  
/k      - Program all non-critical block only  
/kN     - Program N'th non-critical block only  
        (Example: k0/k1/k3 upto k7)  
/c      - Skip non-critical blocks  
/s      - leave signature in BIOS  
/q      - silent execution  
/h      - print help  
/t      - Display ROM ID string  
/c      - Program Main Bios and all Non-critical blocks  
/cN     - Program Main Bios and N'th Non-critical block  
        (from c0 up to c7)  
/d      - Compare ROM file (skips flashing)  
/u<filename> -Display ROM ID
```

2.1.1 How to run the utility under NT 4.0 / 2000 / XP

Copy the file AMIFLDRV.SYS and AFUWIN.EXE to a directory. Run AFUWIN.EXE

2.1.2 How to run the utility under 95 / 98 / ME

Copy the file AMIFLDRV.VXD and AFUWIN.EXE to a directory. Run AFUWIN.EXE

⁵ Note that while AMI generically supports multiple operating systems and flash methods, the system manufacturer may not make these methods available to the end user. In most cases, this is because the system motherboard was not tested & validated with all variations of the AMIBIOS flash utilities.

2.1.3 How to run the utility under DOS⁶

Run the file AFUDOS.EXE. No drivers are required.

2.1.4 How to run the utility under LINUX (kernel version 1.2 and above)⁷

To execute the command from the current directory type `./AFULNX`

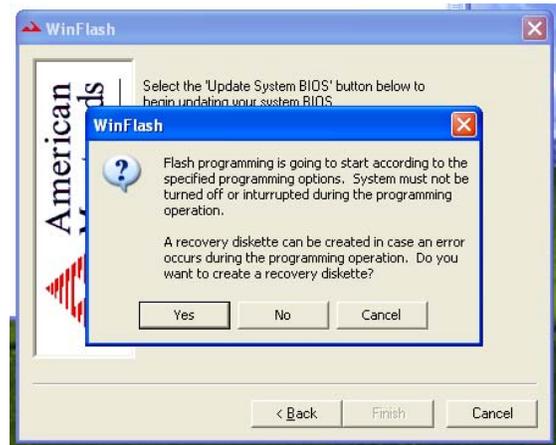
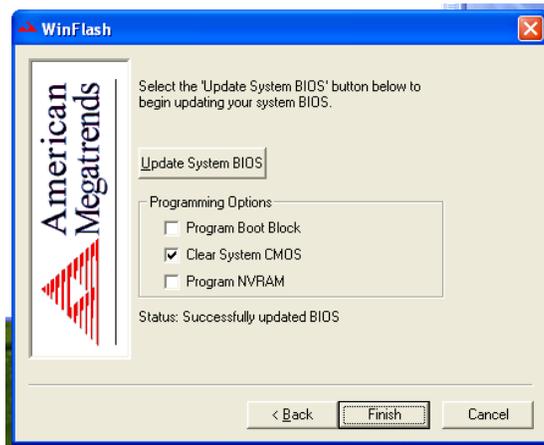
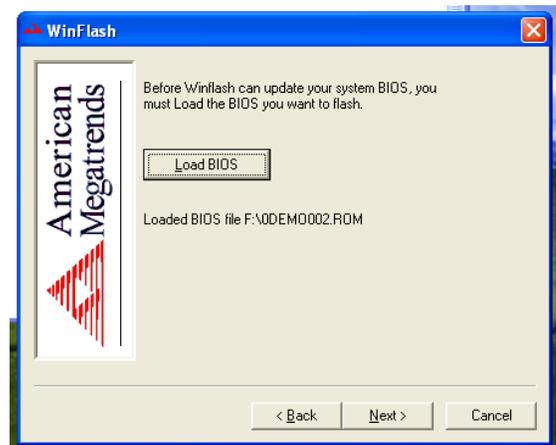
2.1.5 How to run the utility under FreeBSD (version 3.0 and above)

To execute the command from the current directory type `./AFUBSD`

2.2 WinFLASH

WinFLASH has the same capabilities as AFU for Windows (AFUWIN), but features a user-friendly graphical interface. This allows the end-user to easily perform a BIOS update. WinFLASH uses standard flash interface support available in AMIBIOS8.

WinFLASH can be customized based on system manufacturer requirements, including custom vendor logo display.



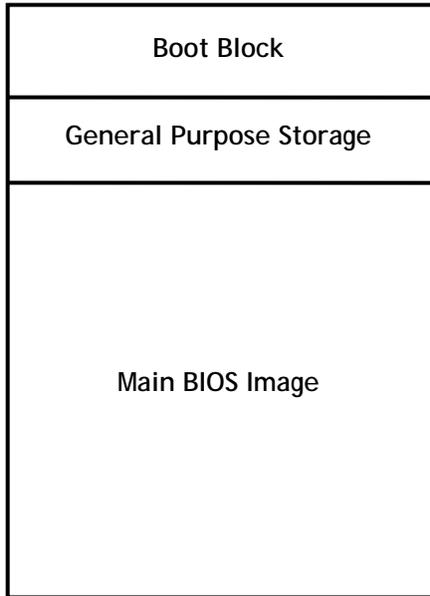
⁶ AFUDOS works only in a "pure DOS" environment, such as booting from a floppy disk. AFUDOS will not work in a Microsoft Windows Command Prompt (a.k.a. "DOS box").

⁷ The Linux kernel has a problem that affects the execution of this utility in system with more than 1GB of memory or for kernel built with 4GB flag. In this case it is necessary to use the utility AFULNX2.

2.3 Existing/Unsupported BIOS Update Tools

There are several existing utilities used to update AMIBIOS from a DOS environment (AMIFLASH & FLASH827). This utility will still update older AMIBIOS products and will work on some AMIBIOS8-based systems, but is not the preferred method for flash updating products based on AMIBIOS8.

3 AMIBIOS8 Boot Block Recovery Methods



Flash chips used to store BIOS on system motherboards are divided into multiple segments. Some segments are for general data storage, while others have special purposes.

The boot block segment of a flash chip contains critical BIOS code, including memory detection and “recovery” code used to flash a new BIOS image in case the main BIOS image is corrupted.

The BIOS boot block code executes first when the system is powered on. Once the boot block code has completed execution, the main BIOS code completed system initialization.

Boot block segments vary in size from 16KB to 128KB. Most systems shipping today use 16KB, 32KB or 64KB boot blocks.

Boot Block Recovery in AMIBIOS8 is improved over previous AMIBIOS products by adding the following features:

- Enabling video for user interaction during recovery.
 - Older boot block recovery methods do not enable video, leaving the user with no prompts or status updates.
 - AMIBIOS8 provides clear user notification of boot block update progress.
- Larger number of methods available to retrieve BIOS image for recovery.
 - The traditional BIOS recovery media has been the 3.5” floppy disk. The floppy disk is no longer a popular method of distributing data, and floppy drives are not always available on newer computers.
 - AMIBIOS8 allows for a variety of methods to perform BIOS recovery.

In most cases, AMIBIOS8 Boot Block Recovery does not rewrite the flash boot block. The AMIBIOS boot block can be rewritten during a flash update using advanced features of our software utilities. This is not recommended, since improperly programming the boot block will most likely erase the code used to perform BIOS recovery.

3.1 Boot Block Recovery from Floppy Disk

Boot Block Recovery from Floppy Disk is a standard feature of the AMIBIOS8 core, and is enabled by default. This allows the user to recover a BIOS image using a standard floppy disk without the need for additional utilities.

Instructions for manually initiating Boot Block Recovery from Floppy Disk:

1. Insert the floppy disk with the new BIOS file in the root directory into drive A:. In most cases the file must be named AMIBOOT.ROM for the recovery process to load the file. However, this filename may differ from one product to another. Please consult the system motherboard documentation for more information.
2. Press and hold the <Ctrl> and <Home>⁸ keys down while turning the power on. Continue to hold the <Ctrl> and <Home> keys down until the access light on the floppy drive comes on. It may take a few seconds or more before this light turns on.
3. Release the <Ctrl> and <Home> keys. AMIBIOS issues a series of beep codes⁹ that indicate that the system BIOS ROM file is being updated. There will also be a progress dialog displayed on the screen after the file has been loaded.
4. When the flash ROM has successfully been programmed, the computer will reboot. Please do not interrupt the BIOS flash process until it has fully completed.

One function of the AMIBIOS Boot Block code is to test the integrity of the BIOS image in flash memory. If AMIBIOS8 Boot Block code detects a problem with the BIOS image, Boot Block recovery will automatically be initiated. This condition will be indicated using a series of beep codes⁹.

3.2 Boot Block Recovery from IDE CD-ROM

Boot Block Recovery from IDE CD-ROM is nearly identical to Boot Block Recovery from Floppy Disk, except that the BIOS image is loaded from CD-ROM. The BIOS image may be loaded from CD-R & CD-RW drives, even if the drive was burned as a multi-session disk. In most cases the file must be named AMIBOOT.ROM for the recovery process to load the file.

Boot Block Recovery from IDE CD-ROM is a standard feature of the AMIBIOS8 core, but is not enabled by default. System motherboard manufacturers have the option of enabling this feature on their systems. Please consult your system motherboard documentation to see if this feature is enabled.

3.3 Boot Block Recovery from USB Storage

Boot Block Recovery from USB Storage is nearly identical to Boot Block Recovery from Floppy Disk, except that the BIOS image is loaded from a USB Storage Device. This includes a variety of USB devices that meet the USB Mass Storage Class definition.

⁸ <CTRL><HOME> is the standard keystroke to initiate BIOS recovery, which also clears CMOS after programming. <CTRL><PGDN> will initiate BIOS recovery without clearing CMOS. <CTRL><PGUP> will initiate BIOS recovery while clearing CMOS & NVRAM.

⁹ Please refer to the "AMIBIOS Checkpoint and Beep Codes" document available at AMI.COM for more information on the meaning of AMIBIOS beep codes.

- USB Floppy
- USB Flash Drive
- USB CDROM & CDRW
- USB ZIP & LS-120/LS-240 "Superdisk"

Boot Block Recovery from USB Storage is a standard feature of the AMIBIOS8 USB eModule, but is not enabled by default. System motherboard manufacturers have the option of enabling this feature on their systems. Please consult your system motherboard documentation to see if this feature is enabled.

3.4 Boot Block Recovery from ATAPI Removable Media

Boot Block Recovery from ATAPI Removable Media, such as Iomega Zip drives & LS-120/LS-240 disks, is nearly identical to Boot Block Recovery from Floppy Disk. In most cases the file must be named AMIBOOT.ROM for the recovery process to load the file.

Boot Block Recovery from ATAPI Removable Media is a standard feature of the AMIBIOS8 core, but is not enabled by default. System motherboard manufacturers have the option of enabling this feature on their systems. Please consult your system motherboard documentation to see if this feature is enabled.

3.5 Boot Block Recovery via Serial Port ("Serial Flash")

The Serial Flash method supported by AMIBIOS8 allows for boot block recovery to load a BIOS image via a serial port. This is used by many embedded systems which rely on a serial port as a debug & utility console port.

Serial Flash Requirements:

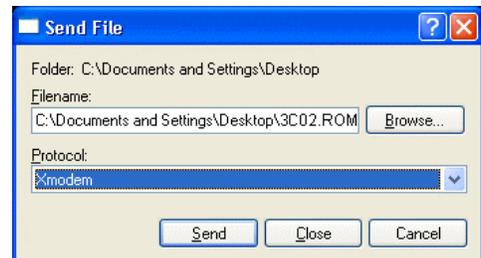
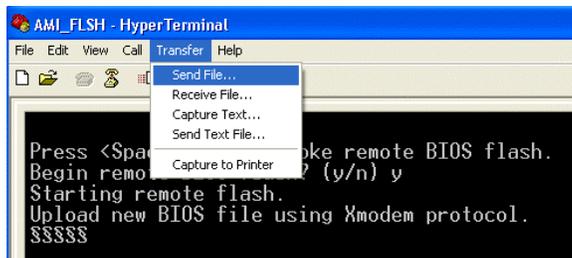
- The system being updated must have a serial port & "Serial Flash" support compiled into the BIOS image
- "Host" system with serial port running terminal programs that support XMODEM transfer protocol (HyperTerminal for Microsoft Windows, minicom for Linux/FreeBSD, etc.)
- Null-modem cable

The following section describes how to use Serial Flash with AMIBIOS8:

1. Attach a null modem cable to the serial port of the system that requires an update ("target"). Attach the other end of the null modem cable to a system running the terminal program ("host").
2. Make sure the new BIOS image file is accessible from the host system.
3. Start the terminal program on the host and open a new session. The session should use the following communication parameters:
 - Bits per second: 115200
 - Data bits: 8
 - Parity: None
 - Stop bits: 1

4. Start the target system. The terminal on the host should display the following message: *Press <SpaceBar> to invoke remote BIOS flash¹⁰*. Immediately press the SpaceBar on the host to confirm. If the SpaceBar is not pressed within a few seconds, the system will skip the flash update and perform a normal boot procedure.
5. A second string will appear on the host terminal: *Begin remote BIOS flash? [y/n]*. Press the 'Y' key on the host to continue. If the 'N' is pressed, the system will skip the flash update and perform a normal boot procedure.
6. You will be prompted to upload the new BIOS file using the XMODEM protocol. Use the host terminal program to select the proper BIOS image and transfer it to the target.
7. If the transfer from host to target is successful, the target will update the BIOS and indicate success. The system will then reboot using the new AMIBIOS image.

HyperTerminal¹¹ for Microsoft Windows is the most common terminal program available today. XMODEM transfers can be initiated using the 'Send File' dialog under the 'Transfer' menu.



AMIBIOS8 Serial Flash will work with any terminal communications program that supports VT-100 and XMODEM protocols. This includes products designed for GNU/LINUX & BSD operating systems, such as minicom. It is recommended that the terminal program be configured to use the 'CR/LF' style of line termination.

Serial Flash is an optional component in AMIBIOS8. This feature may not be supported on every system with AMIBIOS. Please consult your system motherboard documentation to see if this feature is enabled.

¹⁰ If the target system has multiple serial ports, only one will be enabled for Serial Flash. COM1 is the default port, but this can be changed by the system manufacturer. A system with AMIBIOS that does not display the '*Press <SpaceBar> to invoke remote BIOS flash*' string over the serial port does not have support for the Serial Flash feature.

¹¹ There is a known issue with AMIBIOS8 Serial Flash, AMIBIOS8 Serial Console Redirection and the version of HyperTerminal that ships with some installations of Windows 2000. Please make sure you are using the most updated version of HyperTerminal to avoid problems.

3.6 Boot Block Recovery via IPMI Baseboard Management Controller (BMC)

System motherboards designed to support the Intelligent Platform Management Interface (IPMI) make use of a Baseboard Management Controller (BMC). The BMC is a service processor that can remotely manage system hardware, even when the main system processor has halted due to error.

AMI's support for IPMI includes two major system management components:

1. AMIBIOS8 IPMI eModule - Allows AMIBIOS POST to establish communication between the BIOS & BMC and enable full IPMI functionality.
2. Unified Management Server (UMS) - Java-based server application that remotely manages IPMI-enabled systems, including communication with each system's BMC.

The interaction between BIOS & the BMC is managed by the AMIBIOS8 IPMI eModule. A set of IPMI commands has been defined to transfer a new BIOS image from UMS to the BMC and initiate the Boot Block Recovery procedure. This recovery method is initiated by UMS using the IPMI SetBootOption command.

For more information on AMIBIOS8 & IPMI, please refer to the "[American Megatrends System Management Solutions](#)" whitepaper available at [AMI.COM](#).

Revision Information

Date	Rev	Description of Changes	Editor
11 June 2003	0.1	Draft Version	B. Richardson
12 June 2003	0.2	List AMIFLASH as 'unsupported' flash tool (Section 2.3). Add Section 3.4 for recovery from ATAPI Removable Media. Add text to Section 3 describing boot block recovery methods.	B. Richardson
13 June 2003	0.3	Add description of WINFLASH with screenshots. Add FLASH827 to Section 2.3. Reformatted Section 2.1. Added footnote describing <CTRL><PGUP>/<PGDN>. Added description of flash via BMC to Section 3.	B. Richardson
16 June 2003	0.4	Update Section 1.1 & 1.4. Added note for 'CR/LF' configuration for Serial Flash (Section 3.5)	B. Richardson
17 June 2003	0.5	Rearrange Section 1. Add footnotes regarding AMI customers & end-users to Section 1. Add better description of 'BIOS Rom image' to Section 1.3.	B. Richardson U. Vezzani
19 June 2003	1.0	First public release. Corrected AFU title & added DOS to list of supported operating systems (Section 2.1). Moved revision information to last page	B. Richardson
2 June 2005	1.2	Reformatted template	U. Vezzani