

January 7, 2002

# Hard Disk Write Block Tool Specification

Version 1.0 Draft Jan 7

Hard Disk Write Block Tool Specification.....	1
Version 1.0 Draft Jan 7 .....	1
1 INTRODUCTION.....	3
2 PURPOSE .....	3
3 SCOPE .....	3
4 Technical Background.....	3
4.1 Hard Disk Attachment and Access.....	3
4.2 Technical Terminology .....	4
4.3 Terminology Example.....	5
5 REQUIREMENTS .....	5
5.1 Mandatory Requirements .....	5
5.2 Optional Requirements.....	6
6 ASSERTIONS.....	6
7 ABSTRACT TEST CASES .....	6
8 Interrupt 13 BIOS Access.....	7

# 1 INTRODUCTION

Accurate and dependable computer forensics tools are required for a reliable means of investigating crimes that involve computers. In order to insure a measure of reliability and assurance that the results are accurate, the tools used in these investigations should be tested. The Computer Forensics Tool Testing project at the National Institute of Standards and Technology (NIST), an agency of the United States Department of Commerce, provides a measure of confidence in the software tools used in computer forensics investigations. It provides law enforcement personnel with a means of deciding whether the tools in consideration for use should be applied to the purposes required.

The central requirement of a sound forensic examination of digital evidence is that the original evidence must not be modified, i.e., the examination or capture of digital data from the hard disks of a seized computer must be performed so that the disk contents are not changed. The investigator follows a set of procedures designed to prevent the execution of any program that might modify the disk contents. These procedures involve a layered defense against any modifications to the source disk using the following strategies:

- Where possible, set a hardware jumper to make the disk read only.
- Use an operating system and other software that are trusted not to write to the disk unless given explicit instructions.
- Use a hard disk write block tool to intercept any inadvertent disk writes.

## 2 PURPOSE

This document defines requirements for *hard disk write block* tools used in computer forensics investigations and the test methods used to ascertain whether a specific tool meets the requirements.

The requirements are used to derive assertions that will be tested. The assertions are described as general statements of conditions that can be checked after a test is executed. Each assertion generates one or more test cases consisting of a test protocol and the expected test results. The test protocol specifies detailed procedures for setting up the test, executing the test, and measuring the test results.

The requirements and test methods were developed by a focus group of individuals who are expert in the use of hard disk write block tools and have performed investigations that have depended on the results of these tools. As this document evolves through comments from the focus group and others, new versions will be posted to our web site at <http://www.cftt.nist.gov>.

## 3 SCOPE

The scope of this specification is limited to software tools that protect a hard disk attached to a PC from unintended modification. The specifications are general and could be adapted to other write blocking tools. However, actual testing is currently confined to tools that protect disk access through the interrupt 13 BIOS interface of a PC. Not included are tools that protect a hard disk from modification through other mechanisms such as replacing device drives or hardware write blocking tools. Definitions for hard disk drive related terms can be found in NCITS 347:2001 “American National Standard for Information Technology – BIOS Enhanced Disk Drive Services.”

## 4 Technical Background

This section provides technical background for a discussion of write blocking technology. The first subsection presents an overview of how hard disks are physically attached to a computer and then accessed by programs running on the computer. The second subsection defines terminology related to write block tools. The last subsection is an example illustrating how the terminology relates to an actual PC.

### 4.1 Hard Disk Attachment and Access

Before a hard disk drive can be used it must be physically attached to a computer. A hard disk is attached to a computer by one of several available physical interfaces. A disk is usually connected by a cable to an interface controller located

either on the mother board or on a separate adapter card. The most common physical interface is the ATA (AT Attachment or IDE) interface. This includes variants such as EIDE or ATA-2, ATA-3, etc. Some of the other physical interfaces include, but are not limited to SCSI (Small Computer System Interface), IEEE 1394 (aka FireWire or i-Link), and USB (Universal Serial Bus).

All access to a disk is accomplished by commands sent from a computer to a disk through the interface controller. However, since the low level programming required for direct access through the interface controller is difficult and tedious, each operating system usually provides other access interfaces. For example, programs running in the DOS environment can, in addition to direct access via the disk controller, use two other interfaces: DOS service interface (interrupt 21) or BIOS service interface (interrupt 13). The DOS service operates at the logical level of files and records while the BIOS service operates at the physical disk sector level. More sophisticated operating systems, for example Windows NT or a UNIX variant (e.g., Linux), may disallow any low level interface (through the BIOS or controller) and only allow user programs access to a hard disk through a *device driver*, a component of the operating system that manages all access to a device.

## 4.2 Technical Terminology

A hard disk write block tool replaces or monitors a hard disk *access interface* on a general purpose host computer with hard disks attached by a physical interface. A hard disk *access interface* is defined as a method used by a program to access a hard disk. For a program to access a disk, the program issues a high level command to the access interface that is translated by the access interface into the corresponding low level command and is sent to the disk drive through the physical interface controller. For each command issued, the access interface indicates command results (e.g., command completion, error status) by a **return value**. A hard disk write block tool operates by monitoring disk I/O commands sent from the PC through a given access interface. Any commands that could modify a hard disk are intercepted and not passed on to the hard disk. As of the end of 2001, the most common access interfaces are as follows: hard disk device driver, interrupt 13 BIOS (Basic Input Output Services), ATA (AT Attachment) direct controller, ASPI (Advanced SCSI Programming Interface), USB (Universal Serial Bus) and IEEE 1394 (also known as Firewire). Each interface has its own command set and access protocol. The command set for a given interface is partitioned into the following categories:

- **Read:** commands that transfer data from the disk to the computer memory.
- **Write:** commands that transfer data from the computer memory to the disk.
- **Information:** commands that return information about the disk.
- **Control:** commands that request the disk to do a nondestructive operation, for example: *reset* or *seek*.
- **Configuration:** commands that change how the disk is presented to the host computer. These commands often destroy data on the disk or make data inaccessible.
- **Miscellaneous:** commands that do not fit into the other categories.

Appendix A presents two tables: a categorization of the typical interrupt 13 BIOS commands, and a catalog of commands and widely known extensions to the typical command set.

The following terms are defined for convenience in specifying the tool requirements.

- **Covered interface:** a disk access interface that is controlled by the tool.
- **Covered disk:** a disk attached to a **covered interface**.
- **Protected disk:** a disk designated for protection from modification when accessed by a covered interface.
- **Unprotected disk:** a disk that is not protected from modification through a specified access interface.

### 4.3 Terminology Example

Consider a computer with the following four disks attached:

Example Configuration			
Model	Drive	Physical Interface	Access Interfaces
Fujitsu MPF3153AT	0x80	EIDE	ATA, BIOS
WDC WD200BB-00AUA1	0x81	EIDE	ATA, BIOS
QUANTUM ATLAS10K2	0x82	SCSI	ASPI, BIOS
SEAGATE ST318404LC	0x83	SCSI	ASPI, BIOS

Each disk is attached to one physical interface, in this case, either EIDE or SCSI. Each EIDE disk can be accessed in either of two ways. The EIDE disks (attached as drives 0x80 and 0x81) can be accessed directly by the AT-Attachment (ATA) interface to the disk controller, or the disk can be accessed through the BIOS (by interrupt 13). The SCSI disks (attached as drives 0x82 and 0x83) can be accessed through either an ASPI driver or through the BIOS (by interrupt 13).

Consider a write block tool that covers only BIOS disk access through the interrupt 13 interface. All of the disks can be accessed through the BIOS, but suppose that the tool is executed with drives 0x81 and 0x82 specified for protection. The **covered interface** is the interrupt 13 BIOS access. The tool does not cover the ASPI or ATA interfaces. All the disks are **covered disks** when accessed by the BIOS. None of the disks are **covered disks** when accessed by either the ATA or ASPI interfaces. Disks on drives 0x81 and 0x82 are **protected disks** when accessed by the BIOS. All the disks are **unprotected** when accessed by either ATA or ASPI.

Protection Specified for BIOS interrupt 13 access to 0x81 & 0x82		
Disk/access interface	Covered	Protected
0x80/BIOS	yes	no
0x80/ATA	no	no
0x81/BIOS	yes	yes
0x81/ATA	no	no
0x82/BIOS	yes	yes
0x82/ASPI	no	no
0x83/BIOS	yes	no
0x83/ASPI	no	no

Note that even though disks on drives 0x81 and 0x82 are protected from modification by access through the BIOS interface, these disks could still be modified if accessed through the ATA or ASPI interfaces.

## 5 REQUIREMENTS

This section presents mandatory requirements that all write block tools must meet and a list of optional requirements that some write block tools may provide.

### 5.1 Mandatory Requirements

The informal hard disk write block tool requirements are the following:

- The tool shall not allow a protected disk to be changed.
- The tool shall not prevent obtaining any information from or about any disk.
- The tool shall not prevent any changes to a disk that is not protected.

The three informal requirements are the essence of a write blocking tool: protect the evidence from alteration while allowing a complete examination of the evidence. A formal statement of these requirements follows:

- 5.1.1 The tool shall block any commands to a protected disk in the write, configuration, or miscellaneous categories.
- 5.1.2 The tool shall not block any commands to an unprotected disk.
- 5.1.3 The tool shall not block any commands to a protected disk in the read or information categories.

- 5.1.4 The tool shall give an indication to the user that the tool is active.
- 5.1.5 The tool shall report all disks accessible by the covered interfaces.
- 5.1.6 The tool shall report the protection status of all disks.
- 5.1.7 The tool shall, if so configured, adjust the return value of any blocked commands to indicate that the operation was carried out successfully even though the operation was blocked.
- 5.1.8 The tool shall, if so configured, adjust the return value of any blocked commands to indicate that the operation failed.
- 5.1.9 Return values of information commands shall be consistent with return values of any blocked commands. (For example, a command to *return status of last command* after a blocked command shall return the same value as returned by the blocked command.)

## 5.2 Optional Requirements

The following requirements define optional tool features. If a tool provides the capability defined, the tool is tested as if the requirement were mandatory. If the tool does not provide the capability defined, the requirement does not apply.

- 5.2.1. The tool shall alert the user when a command is blocked, either by an audio or a visual signal.
- 5.2.2. The tool shall be able to uninstall itself if requested.
- 5.2.3. The user shall be able to specify a subset of the covered disks for protection.
- 5.2.4. The tool shall log a subset of command executions that have been blocked.

## 6 ASSERTIONS

**[Editor's note: Assertions will be added after the requirements are in final form.]**

Each assertion provides a specific class of conditions that can be tested and the result that is expected.

## 7 ABSTRACT TEST CASES

**[Editor's note: Test cases will be added after the requirements are in final form.]**

Abstract test cases describe the combinations of test parameters required to fully test each assertion. They are abstract in that they do not prescribe the exact environment in which the tests are to be performed. They are written at the next level above the environment. This allows different environments to be substituted under the test cases for testing different products and options.

A set of test parameters is chosen to cover the assertions from various aspects. Not all possible tests will be specified since this number could run into the hundreds or thousands based on the combinations of parameters that could be used. Exhaustive testing, in most cases, is not economically feasible. Instead, a subset of parameters will be used to define the set of test cases needed to evaluate tools against the requirements.

## 8 Interrupt 13 BIOS Access

A typical set of interrupt 13 BIOS disk access commands can be categorized as follows. Other BIOS vendors or software installed on a PC may add or change functionality for the interrupt 13 interface.

Categorization of Interrupt 13 BIOS Commands		
Command	Code	Category
Reset	00h	control
Get last status	01h	information
read sectors	02h	read
Write sectors	03h	Write
Verify sectors	04h	information (or read or control)
Format Cylinder	05h	Configuration
Read Drive Parameters	08h	Information
Initialize Drive Parameters	09h	Configuration
Read Long Sector	0Ah	Read
Write Long Sector	0Bh	Write
Seek Drive	0Ch	Control
Alternate disk reset	0Dh	Control
Test drive ready	10h	Information
Recalibrate drive	11h	Configuration
Controller diagnostic	14h	Configuration
Read drive type	15h	Information
Check extensions present	41h	Information
Extended read	42h	Read
Extended write	43h	Write
Verify sectors	44h	Information
Extended seek	47h	Control
Get drive parameters	48h	Information

The following table is a list of common interrupt 13 BIOS disk access commands and widely known extensions derived from <http://www.ctyme.com/rbrown.htm> (August 23, 2000). NIST does not make any claims for the accuracy of the contents but has included this to illustrate commands that should be used for test cases of interrupt 13 based write blocker tools. Some commands that might modify hard disk contents are highlighted with a gray background.

Common Interrupt 13 BIOS Commands	
Command	Description
AH = 00h	DISK - RESET DISK SYSTEM
AH = 01h	DISK - GET STATUS OF LAST OPERATION
AH = 02h	DISK - READ SECTOR(S) INTO MEMORY
AH = 03h	DISK - WRITE DISK SECTOR(S)
AH = 04h	DISK - VERIFY DISK SECTOR(S)
AH = 05h	FLOPPY - FORMAT TRACK
AH = 05h	FIXED DISK - FORMAT TRACK
AH = 05h	Future Domain SCSI BIOS - SEND SCSI MODE SELECT COMMAND
AX = 057Fh	2M - FORMAT TRACK
AH = 06h	FIXED DISK - FORMAT TRACK AND SET BAD SECTOR FLAGS (XT,PORT)
AH = 06h	Future Domain SCSI BIOS - FORMAT DRIVE WITH BAD SECTOR MAPPING
AH = 06h	Adaptec AHA-154xA/Bustek BT-542 BIOS - IDENTIFY SCSI DEVICES
AH = 06h	V10DISK.SYS - READ DELETED SECTORS
AH = 07h	FIXED DISK - FORMAT DRIVE STARTING AT GIVEN TRACK (XT,PORT)
AH = 07h	Future Domain SCSI BIOS - FORMAT DRIVE
AH = 07h	V10DISK.SYS - WRITE DELETED SECTORS

<b>Common Interrupt 13 BIOS Commands</b>	
<b>Command</b>	<b>Description</b>
AH = 08h	DISK - GET DRIVE PARAMETERS (PC,XT286,CONV,PS,ESDI,SCSI)
AH = 08h	V10DISK.SYS - SET FORMAT
AX = 08000h	SecureDrive - INSTALLATION CHECK
AH = 09h	HARD DISK - INITIALIZE CONTROLLER WITH DRIVE PARAMETERS (AT,PS)
AH = 0Ah	HARD DISK - READ LONG SECTOR(S) (AT and later)
AH = 0Bh	HARD DISK - WRITE LONG SECTOR(S) (AT and later)
AH = 0Ch	HARD DISK - SEEK TO CYLINDER
AH = 0Dh	HARD DISK - RESET HARD DISKS
AH = 0Eh	HARD DISK - READ SECTOR BUFFER (XT only)
AH = 0Fh	HARD DISK - WRITE SECTOR BUFFER (XT only)
AH = 10h	HARD DISK - CHECK IF DRIVE READY
AH = 11h	HARD DISK - RECALIBRATE DRIVE
AH = 12h	HARD DISK - CONTROLLER RAM DIAGNOSTIC (XT,PS)
AH = 12h	Future Domain SCSI CONTROLLER - STOP SCSI DISK
AH = 12h	SyQuest - START/STOP SCSI DISK
AH = 13h	HARD DISK - DRIVE DIAGNOSTIC (XT,PS)
AH = 13h	SyQuest - READ DRIVE PARAMATERS (for DOS 5+)
AH = 14h	HARD DISK - CONTROLLER INTERNAL DIAGNOSTIC
AH = 15h	DISK - GET DISK TYPE (XT 1/10/86 or later,XT286,AT,PS)
AH = 16h	FLOPPY DISK - DETECT DISK CHANGE (XT 1/10/86 or later,XT286,AT,PS)
AH = 17h	FLOPPY DISK - SET DISK TYPE FOR FORMAT (AT,PS)
AX = 1700h	Future Domain SCSI CONTROLLER - GET INQUIRY INFO FROM SCSI DEVICE
AH = 18h	DISK - SET MEDIA TYPE FOR FORMAT (AT model 3x9,XT2,XT286,PS)
AH = 18h	Future Domain SCSI BIOS - GET SCSI CONTROLLER INFORMATION
AH = 18h	PU_1700.COM - INSTALLATION CHECK
AH = 18h	XDF.COM - API
AH = 19h	FIXED DISK - PARK HEADS ON ESDI DRIVE (XT286,PS)
AH = 19h	Future Domain SCSI CONTROLLER - REINITIALIZE DRIVE
AH = 1Ah	ESDI FIXED DISK - FORMAT UNIT (PS)
AH = 1Ah	Future Domain SCSI CONTROLLER - GET SCSI PARTIAL MEDIUM CAPACITY
AH = 1Bh	ESDI FIXED DISK - GET MANUFACTURING HEADER
AH = 1Bh	Future Domain SCSI CONTROLLER - GET POINTER TO SCSI DISK INFO BLOCK
AH = 1Ch	Future Domain SCSI CONTROLLER - GET POINTER TO FREE CONTROLLER RAM
AH = 1Ch	ESDI FIXED DISK - ???
AX = 1C08h	ESDI FIXED DISK - GET COMMAND COMPLETION STATUS
AX = 1C09h	ESDI FIXED DISK - GET DEVICE STATUS
AX = 1C0Ah	ESDI FIXED DISK - GET DEVICE CONFIGURATION
AX = 1C0Bh	ESDI FIXED DISK - GET ADAPTER CONFIGURATION
AX = 1C0Ch	ESDI FIXED DISK - GET POS INFORMATION
AX = 1C0Dh	ESDI FIXED DISK - ???
AX = 1C0Eh	ESDI FIXED DISK - TRANSLATE RBA TO ABA
AX = 1C0Fh	ESDI FIXED DISK - ???
AX = 1C12h	ESDI FIXED DISK - ???
AH = 1Dh	IBMCACHE.SYS - CACHE STATUS
AH = 1Fh	SyQuest - DOOR LATCH/DOOR BUTTON DETECT
AH = 20h	DISK - ??? (Western Digital "Super BIOS")
AH = 20h	Compaq, ATAPI Removable Media Device - GET CURRENT MEDIA FORMAT
AH = 20h	QUICKCACHE II v4.20 - DISMOUNT
AH = 21h	HARD DISK - PS/1 and newer PS/2 - READ MULTIPLE DISK SECTORS
AH = 21h	QUICKCACHE II v4.20 - FLUSH CACHE
AH = 22h	HARD DISK - PS/1 and newer PS/2 - WRITE MULTIPLE DISK SECTORS



<b>Common Interrupt 13 BIOS Commands</b>	
<b>Command</b>	<b>Description</b>
AH = 22h	QUICKCACHE II v4.20 - ENABLE/DISABLE CACHE
AH = 23h	HARD DISK - PS/1 and newer PS/2 - SET CONTROLLER FEATURES REGISTER
AH = 23h	QUICKCACHE II v4.20 - GET ??? ADDRESS
AH = 24h	HARD DISK - PS/1 and newer PS/2 - SET MULTIPLE MODE
AH = 24h	QUICKCACHE II v4.20 - SET SECTORS
AH = 25h	HARD DISK - PS/1 and newer PS/2 - IDENTIFY DRIVE
AH = 25h	QUICKCACHE II v4.20 - SET FLUSH INTERVAL
AH = 26h	QUICKCACHE II v4.20 - UNINSTALL
AH = 27h	QUICKCACHE II v4.20 - INSTALLATION CHECK
AH = 28h	QUICKCACHE II v4.20 - SET AUTOMATIC DISMOUNT
AH = 29h	QUICKCACHE II v4.20 - NOP
AH = 2Ah	QUICKCACHE II v4.20 - SET BUFFER SIZE
AH = 2Bh	QUICKCACHE II v4.20 - DRIVE ACCESS SOUNDS
AH = 2Ch	QUICKCACHE II v4.20 - SET BUFFERED WRITES
AH = 2Dh	QUICKCACHE II v4.20 - SET BUFFERED READ
AH = 2Eh	QUICKCACHE II v4.20 - SET FLUSH COUNT
AH = 2Fh	QUICKCACHE II v4.20 - FORCE IMMEDIATE INCREMENTAL FLUSH
AH = 30h	QUICKCACHE II v4.20 - GET INFO
AH = 31h	QUICKCACHE II v4.20 - RESERVE MEMORY
AH = 32h	QUICKCACHE II v4.20 - ENABLE CACHING FOR SPECIFIC DRIVE
AH = 33h	QUICKCACHE II v4.20 - DISABLE CACHING FOR SPECIFIC DRIVE
AH = 34h	QUICKCACHE II v4.20 - SECTOR LOCKING
AH = 35h	QUICKCACHE II v4.20 - SET LOCK POOL SIZE
AH = 36h	QUICKCACHE II v4.20 - SET TRACE BUFFER SIZE
AH = 37h	QUICKCACHE II v4.20 - SET BUFFERED READS FOR SPECIFIC DRIVE
AH = 38h	QUICKCACHE II v4.20 - SET BUFFERED WRITES FOR SPECIFIC DRIVE
AH = 39h	QUICKCACHE II v4.20 - SET READ BUFFER SIZE FOR SPECIFIC DRIVE
AH = 3Ah	QUICKCACHE II v4.20 - SET WRITE BUFFER SIZE FOR SPECIFIC DRIVE
AH = 3Bh	QUICKCACHE II v4.20 - ENABLE/DISABLE ???
AH = 3Ch	QUICKCACHE II v4.20 - ENABLE/DISABLE ???
AH = 3Dh	QUICKCACHE II v4.20 - ENABLE/DISABLE CYLINDER FLUSH FOR DRIVE
AH = 3Eh	QUICKCACHE II v4.20 - SET SINGLE-SECTOR BONUS
AH = 3Fh	QUICKCACHE II v4.20 - SET BONUS THRESHOLD
AH = 40h	QUICKCACHE II v4.20 - SET "sticky_max"
AH = 41h	IBM/MS INT 13 Extensions - INSTALLATION CHECK
AH = 41h	QUICKCACHE II v4.20 - SAVE/RESTORE ???
AH = 42h	IBM/MS INT 13 Extensions - EXTENDED READ
AX = 4257h ("BW")	Beame&Whiteside BWLPD - INSTALLATION CHECK
AH = 43h	IBM/MS INT 13 Extensions - EXTENDED WRITE
AH = 44h	IBM/MS INT 13 Extensions - VERIFY SECTORS
AH = 45h	IBM/MS INT 13 Extensions - LOCK/UNLOCK DRIVE
AH = 46h	IBM/MS INT 13 Extensions - EJECT MEDIA
AH = 47h	IBM/MS INT 13 Extensions - EXTENDED SEEK
AH = 48h	IBM/MS INT 13 Extensions - GET DRIVE PARAMETERS
AH = 49h	IBM/MS INT 13 Extensions - EXTENDED MEDIA CHANGE
AH = 4Ah	Bootable CD-ROM - INITIATE DISK EMULATION
AX = 4B00h	Bootable CD-ROM - TERMINATE DISK EMULATION
AX = 4B01h	Bootable CD-ROM - GET STATUS
AH = 4Ch	Bootable CD-ROM - INITIATE DISK EMULATION AND BOOT
AX = 4D00h	Bootable CD-ROM - RETURN BOOT CATALOG
AH = 4Eh	IBM/MS INT 13 Extensions v2.1 - SET HARDWARE CONFIGURATION

<b>Common Interrupt 13 BIOS Commands</b>	
<b>Command</b>	<b>Description</b>
AX = 5001h	Enhanced Disk Drive Spec v3.0 - SEND PACKET COMMAND
AX = 5001h	VIRUS - "Andropinis" - INSTALLATION CHECK
AX = 5342h ("SB")	ScanBoot - INSTALLATION CHECK
AX = 5501h	Seagate ST01/ST02 - Inquiry
AX = 5502h	Seagate ST01/ST02 - RESERVED
AX = 5503h	Seagate ST01/ST01 - Set Device Type Qualifier (DTQ)
AX = 5504h	Seagate - ??? - RETURN IDENTIFICATION
AX = 5504h	Seagate ST01/ST02 - RETURN IDENTIFICATION
AX = 5505h	Seagate - ??? - PARK HEADS
AX = 5505h	Seagate ST01/ST02 - PARK HEADS
AX = 5506h	Seagate ST01/ST02 - SCSI Bus Parity
AX = 5507h to 550Dh	Seagate ST01/ST02 - RESERVED FUNCTIONS
AX = 5514h	Seagate - ???
AX = 5515h	Seagate - PARK HEADS???
AH = 59h	SyQuest - Generic SCSI pass through
AH = 70h	Priam EDVR.SYS DISK PARTITIONING SOFTWARE???
AH = 75h	???
AH = 76h	???
AX = 7B00h	NOW! v3.05 - GET INFORMATION
AX = 7B01h	NOW! v3.05 - ???
AX = 7B02h	NOW! v3.05 - SET INFORMATION
AX = 7B03h	NOW! v3.05 - ???
AX = 7B04h	NOW! v3.05 - ???
AX = 7B05h	NOW! v3.05 - GET DISK ACCESSES???
AX = 7B06h	NOW! v3.05 - GET ???
AX = 7B07h	NOW! v3.05 - GET ???
AX = 7B08h	NOW! v3.05 - ???
AH = 80h	FAST! v4.02+ - API
AX = 8001h	FAST! v4.02+ - GET CACHE INFORMATION
AX = 8006h	FAST! v4.02+ - INSTALLATION CHECK
AX = 8007h	FAST! v4.02+ - UNHOOK INTERRUPTS
AH = 81h	Super PC-Kwik v3.20+ - ???
AH = 82h	Super PC-Kwik v3.20+ - ???
AH = 83h	Super PC-Kwik v3.20+ - ???
AH = 84h	Super PC-Kwik v3.20+ - ???
AH = 85h	Super PC-Kwik v3.20+ - ???
AH = 86h	Super PC-Kwik v4.00+ - ???
AH = 87h	Super PC-Kwik v4.00+ - ???
AH = 88h	Super PC-Kwik v4.00+ - ???
AH = 89h	Super PC-Kwik v5.10+ - ???
AH = 8Ah	Super PC-Kwik v5.10+ - ???
AX = 8EEDh	HyperDisk v4.01+ - ???
AX = 8EEEh	HyperDisk v4.01+ - ???
AX = 8EEFh	HyperDisk v4.01+ - ???
AH = 92h	Super PC-Kwik v5.10+ - ???
AH = 93h	Super PC-Kwik v5.10+ - ???
AH = 94h	Super PC-Kwik v5.10+ - ???
AH = 95h	Super PC-Kwik v5.10+ - ???
AH = 96h	Super PC-Kwik v5.10+ - ???
AH = 97h	Super PC-Kwik v5.10+ - ???

<b>Common Interrupt 13 BIOS Commands</b>	
<b>Command</b>	<b>Description</b>
AH = 98h	Super PC-Kwik v5.10+ - ???
AH = 99h	Super PC-Kwik v5.10+ - ???
AH = 9Ah	Super PC-Kwik v5.10+ - ???
AH = 9Bh	Super PC-Kwik v5.10+ - ???
AH = 9Ch	Super PC-Kwik v5.10+ - ???
AH = 9Dh	Super PC-Kwik v5.10+ - ???
AH = A0h	Super PC-Kwik v3.20+ - GET RESIDENT CODE SEGMENT
AH = A1h	Super PC-Kwik v3.20+ - FLUSH CACHE
AH = A2h	Super PC-Kwik v3.20+ - ???
AH = A3h	Super PC-Kwik v5.10+ - DISABLE CACHE
AH = A4h	Super PC-Kwik v5.10+ - ENABLE CACHE
AH = A5h	Super PC-Kwik v5.10+ - PROGRAM TERMINATION NOTIFICATION
AH = A6h	Super PC-Kwik v5.10+ - PROGRAM LOAD NOTIFICATION
AH = A7h	Super PC-Kwik 5.1 - ???
AX = A759h	Novell DOS 7 - SDRes v27.03 - ???
AH = A8h	Super PC-Kwik 5.1 - ???
AH = A9h	Super PC-Kwik 5.1 - EXITCODE RETRIEVAL NOTIFICATION
AH = AAh	Super PC-Kwik v4+ - ???
AH = ABh	Super PC-Kwik v4+ - ???
AH = ACh	Super PC-Kwik v4+ - ???
AH = ADh	Priam HARD DISK CONTROLLER???
AH = ADh	Super PC-Kwik v4+ - ???
AH = AEh	Super PC-Kwik v5.10+ - ???
AH = B0h	Super PC-Kwik v3.20+ - ???
AX = E000h	XBIOS - COMMAND
AX = EC00h	VIRUS - "Tiso" - INSTALLATION CHECK
AH = EEh	SWBIOS - SET 1024-CYLINDER FLAG
AH = EFh	Ontrack Drive Rocket - SET CYLINDER OFFSET
AH = F2h	VIRUS - "Neuroquila" - INSTALLATION CHECK
AH = F9h	SWBIOS - INSTALLATION CHECK
AH = FAh	PC Tools v8+ VSAFE, VWATCH - API
AX = FD50h	VIRUS - "Predator" - INSTALLATION CHECK
AH = FEh	SWBIOS - GET EXTENDED CYLINDER COUNT
AH = FFh	EZ-Drive - INSTALLATION CHECK
AH = FFh	IBM SurePath BIOS - Officially "Private" Function
AX = FFFFh	UNIQUE UX Turbo Utility - SET TURBO MODE