March 6, 2003

Test Environment and Procedures for Testing SafeBack 2.18

Version 1.0

NIST
National Institute of Standards and Technology
Technology Administration, U.S. Department of Commerce

Abstract[†]

This document describes the testing of **SafeBack** 2.18. The test cases that were applied are described in *Disk Imaging Tool Specification, Version 3.1.6*.

The tests were run on test systems in the Computer Forensics Tool Testing Lab at the National Institute of Standards and Technology. A variety of hard drives were used for the tests. The source disks (the ones that are copied from) were setup with FAT16, FAT32, NTFS or Linux EXT2 type partitions to represent the most common partition types.

The main objective of this document is to provide enough information about the testing process for either an independent evaluation of the process or independent replication of the results. The intended audience for this document should be familiar with the DOS operating system, computer operation, computer hardware components such as hard drives, hard drive interfaces (e.g., IDE or SCSI) and computer forensics.

[†] **Certain trade names and company products are mentioned in the text or identified. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the products are necessarily the best available for the purpose.**

Table of Contents

List of Tables

List of Figures

1 Introduction

The objective of the Computer Forensics Tool Testing (CFTT) project is to provide a measure of assurance that the tools used in computer forensics investigations produce accurate results. This is accomplished by developing specifications and test methods for computer forensics tools and then testing specific tools. The test results provide the information necessary for toolmakers to improve tools, for users to make informed choices about acquiring and using computer forensics tools, and for the legal community and others to understand the tool capabilities. Our approach for testing computer forensic tools is based on well-recognized methodologies for conformance testing and quality testing.

The CFTT is a joint project of the National Institute of Justice, the National Institute of Standards and Technology (NIST), and other agencies, such as the Technical Support Working Group. The entire computer forensics community helps develop the specifications and test methods by commenting on drafts as they are published on the NIST website http://www.cftt.nist.gov/.

This document describes the procedures used for testing **SafeBack** 2.18. The test cases that were applied are described in *Disk Imaging Tool Specification, Version 3.1.6*. The main objective of this document is to provide enough information about the testing process for either an independent evaluation of the process or independent replication of the results. An independent replication of the testing would require sufficient hardware and software resources to execute the test cases, this document, the **SafeBack** test report, *Disk Imaging Tool Specification, Version 3.1.6*, the FS-TST 1.0 software plus revised error simulation software . Since it is unlikely that the exact hardware used by NIST is present, adjustments and substitutions must be made to run the test cases. Section 6 gives suggestions for adapting to a test environment different from the environment at NIST.

The intended audience for this document should be familiar with the DOS operating system, computer operation, computer hardware components such as hard drives, hard drive interfaces (e.g., IDE or SCSI) and computer forensics.

1.1 Testing Overview

Several items were assembled and prepared before the testing began including: computers to execute the tests, hard disk drives, removable media, support software (FS-TST Version 1.0) and scripts to control the testing process. The support software, *FS-TST: Forensic Software Testing Support Tools*, revised error simulation software, scripts and the documentation for the software is available from the web site: http://www.cftt.nist.gov.

A subset of the hard drives was selected for initial setup as source drives for the test cases. The source drives were setup once and then used for multiple test cases. After all the

components were prepared, the test cases were run. All the test cases followed a similar execution plan of three steps.

1. Prepare the destination drive. Specified values were written to each sector of the destination drive. If a partition was required, it was created and formatted. This step was executed in a DOS environment.
2. Execute **SafeBack**. This step was executed in a DOS environment.
3. Measure the results. The accuracy and completeness of the copy was checked by a sector-by-sector comparison. The source drive was checked for any change by comparing a SHA-1 taken before the execution of **SafeBack** with a SHA-1 taken after **SafeBack** was executed. This step was executed in a DOS environment.

For each test case, the commands that need to be executed were contained in a set of script files. Except for partition creation and formatting, the programs required to setup each test case and to measure the results are contained in the FS-TST package.

1.2 Test Case Selection

Not all of the 168 the test cases specified in *Disk Imaging Tool Specification, version 3.1.6* applied to SafeBack. Of the 168 test cases specified, 122 cases were applied to SafeBack and 56 cases were not run.

The primary criterion for selecting a test case was for there to be a tool feature that was covered by the objective of the test case as defined by the test case summary from *Disk Imaging Tool Specification, version 3.1.6*. For example, test case DI-001 calls for the following setup: *Copy a BIOS-IDE source disk to a BIOS-IDE destination disk where the source disk is smaller than the destination.* Since every parameter specified in the setup could be applied to SafeBack, test case DI-001 was used. However, for test case DI-013, logical copy of a Linux (i.e., ext2 or ext3) partition, the test case was not used since SafeBack does not allow selection of a logical drive formatted as a Linux partition for the **copy** operation.

The 56 cases that were not run were eliminated for one or more of the following reasons (see Table 1-1 for details):

- SafeBack does not support partition (logical drive) operations (**copy restore**, or **backup**) on NTFS or Linux Ext2 partitions.
- Some test cases are going to be deleted from the test specification. For example, cases involving deleted file recovery are being deleted from the specification since deleted file recovery tools will be tested separately.
- Some test cases require support software or other tools that are not available, e.g., some test cases specify I/O error simulation beyond the scope of the current tools.

Table 1-1 Test Cases Not Applied to SafeBack

Case	Reason case not used
DI-011	Unsupported operation
DI-012	Unsupported partition type

Case	Reason case not used
DI-013	Case deleted, Unsupported partition type
DI-026	Case deleted, Unsupported partition type
DI-028	Unsupported partition type
DI-029	Unsupported partition type
DI-037	Unsupported partition type
DI-038	Unsupported partition type
DI-039	Case deleted
DI-042	Unsupported partition type
DI-043	Unsupported partition type
DI-052	Case deleted
DI-053	Case deleted
DI-073	Unsupported partition type
DI-074	Unsupported partition type
DI-075	Case deleted
DI-076	Case deleted
DI-077	Case deleted, Unsupported partition type
DI-078	Case deleted, Unsupported partition type
DI-079	Unsupported partition type
DI-080	Unsupported partition type
DI-084	Unsupported partition type
DI-085	Unsupported partition type
DI-086	Beyond scope of err simulator
DI-087	Unsupported partition type
DI-088	Unsupported partition type
DI-095	Beyond scope of err simulator
DI-096	Beyond scope of err simulator
DI-102	Unsupported partition type
DI-103	Unsupported partition type
DI-104	Case deleted, Unsupported partition type
DI-105	Case deleted, Unsupported partition type
DI-106	Case deleted
DI-107	Case deleted
DI-110	Unsupported partition type
DI-111	Unsupported partition type
DI-112	Unsupported partition type
DI-113	Unsupported partition type
DI-114	Beyond scope of err simulator
DI-115	Beyond scope of err simulator
DI-116	Unsupported partition type
DI-117	Unsupported partition type
DI-124	Beyond scope of err simulator
DI-125	Beyond scope of err simulator
DI-131	Unsupported partition type
DI-132	Unsupported partition type
DI-133	Case deleted
DI-134	Case deleted
DI-135	Case deleted, Unsupported partition type
DI-136	Case deleted, Unsupported partition type
DI-143	Beyond scope of err simulator
DI-144	Beyond scope of err simulator
DI-165	Case deleted
DI-166	Case deleted
DI-167	Case deleted

Case	Reason case not used
DI-168	Case deleted

1.3 Document Overview

Section 2 describes the test hardware, hard disk drives used and system configurations for running the tests. The procedures for creation or setup of source disks, DOS boot floppies, Linux boot drive and Jaz disk are described in Section 3. The script files for each step are described in Section 4. Section 5 describes the execution procedures. Guidelines for executing the tests in a different environment are presented in the last section.

2 Test Environment

The tests were run using eleven host computers: **Beta1**, **Beta3, Beta4, Beta6, Beta7, Delta1, Marple, Cadfael, Rumpole, Wimsey,** and **JudgeDee**. More than 40 hard drives (14 different models, 6 different brands) were used for the tests (Table 2-2). The tests were run with the hard drives arranged in one of 12 possible configurations (Table 2-3) as required by the test parameters.

2.1 Extended BIOS Host Computers: Cadfael, Rumpole, Wimsey, and JudgeDee

Four host computers (**Cadfael, Rumpole, Wimsey,** and **JudgeDee)** had the following hardware components in common:

Table 2-1 Extended BIOS Host Computer Hardware Components

ASUS CUSL2 Motherboard
BIOS: Award Medallion v6.0
Intel Pentium III (Coppermine) 933Mhz
512672k Memory
Adaptec 29160N SCSI Adapter card
Plextor CR-RW PX-W124TS Rev: 1.06
Iomega 2GB Jaz drive Rev: E.17
LS-120 Super floppy
Two slots for removable IDE hard disk drives
Two slots for removable SCSI hard disk drive

The computer **Rumpole** also had a 30GB OnStream SC30 tape drive (not used in the test procedures). The computer **JudgeDee** had a third slot for a removable IDE hard disk drive.

2.2 Legacy BIOS Host computers: Beta1, Beta3, Beta4, Beta6 and Beta7

Beta1, Beta3, Beta4, Beta6 and **Beta7** were Nexar 166MHz computers with 256MB RAM , two hard disk drive bays, both of which take hard drives mounted in removable carriages, a CD-ROM drive, a 1.44MB floppy drive, and a 17" color monitor. The usual operating system environment was PC-DOS 6.3, booted from the floppy drive. The motherboard was a HCL Hewlett-Packard Integrated ISA/PCI P54C with an Award v4.51PG BIOS. **Beta7** also had an Adaptec 29160N SCSI Adapter card with an Iomega 2GB Jaz drive Rev: E.17 attached.

2.3 Special SCSI System: Marple

Marple was a Nexar 166MHz computer with 256MB RAM , two SCSI hard disks (C0 and C1), a CD-ROM drive, a 1.44MB floppy drive, and a 17" color monitor. The SCSI adapter was the Adaptec AHA-2940UW Pro with SCSI BIOS V2.11.0. The usual operating system environment was PC-DOS 6.3, booted from the floppy drive. The BIOS was Award v4.51PG.

Marple was used on some SCSI test cases.

2.4 Fast SHA-1 for Nexar Tests: Delta1

Delta1 was a Dell Computer Corporation system with 256MB RAM, one hard disk drive bay, one installed 15.37 GB hard disk, a CD-ROM drive, a 1.44MB floppy drive, a 250 MB ZIP drive, and a 17" color monitor. The usual operating system environment was PC-DOS 6.3, booted from the floppy drive. The BIOS was PhoenixBios 4.0 Release 6.0.

Delta1 (888Mhz) computes SHA-1 values much faster than the Nexar (166 Mhz) systems and was used to compute SHA-1 values for tests run on Nexar systems as needed.

2.5 Hard Disk Drives

The hard disk drives that were used are listed in Table 2-2. These hard drives were mounted in removable DataPort storage modules. Any combination of two IDE hard drives and two SCSI hard drives were installed in **Cadfael, Rumpole, Wimsey,** or **JudgeDee** as required for a test. The legacy BIOS computers could only have two IDE drives mounted at a time.

The IDE disks used in the legacy BIOS computers had jumpers set manually to drive 0 for source drives, drive 1 for destination drives and the media drive is set to either 0 or 1 depending on the available drive slot available after either the source or destination drive was installed. The IDE disks used in **Cadfael, Rumpole, Wimsey,** and **JudgeDee** had jumpers set for *cable select.*

The SCSI ID for the SCSI disk was set to either 0 or 1 as required by the test case. Except as noted, a source disk was set to ID 0 and a destination disk was set to ID 1.

Table 2-2 Hard Drives Used in Testing

Label	Model	Interface	Usable sectors	Capacity (GB)
1E	QUANTUM ATLAS10K3_18_SCA	SCSI	35916547	18.38
63	WDCWD64AA	IDE	12594960	6.45
64	WDCWD64AA	IDE	12594960	6.45
65	WDCWD64AA	IDE	12594960	6.45
A1	QUANTUM SIROOCO1700A	IDE	3335472	1.70
A3	QUANTUM SIROOCO1700A	IDE	3335472	1.70
A4	QUANTUM SIROOCO1700A	IDE	3335472	1.70
A5	WDC WD200BB-00AUA1	IDE	39102336	20.02
A6	WDC WD200BB-00AUA1	IDE	39102336	20.02
A7	WDC WD200BB-00AUA1	IDE	39102336	20.02
A8	WDC WD200BB-00AUA1	IDE	39102336	20.02
AB	MAXTOR53073H4	IDE	60030432	30.73
B5	WDC AC21600H	IDE	3173184	1.62
B6	WDC AC21600H	IDE	3173184	1.62
B7	WDC AC21600H	IDE	3173184	1.62
B8	WDC AC21600H	IDE	3173184	1.62
B9	WDC AC21600H	IDE	3173184	1.62
BA	QUANTUM FIREBALL ST3.2A	IDE	6306048	3.22
BB	QUANTUM FIREBALL ST3.2A	IDE	6306048	3.22
BC	QUANTUM FIREBALL ST3.2A	IDE	6306048	3.22
BD	QUANTUM FIREBALL ST3.2A	IDE	6306048	3.22
C0	SEAGATE ST39204LC	SCSI	17921835	9.17
C1	SEAGATE ST39204LC	SCSI	17921835	9.17
CB	SEAGATE ST336705LC	SCSI	71687370	36.70
CC	SEAGATE ST336705LC	SCSI	71687370	36.70
D3	FUJITSU MPE3064AT	IDE	12672450	6.48
D7	QUANTUM SIROOCO1700A	IDE	3335472	1.70
DA	FUJITSU MPE3064AT	IDE	12672450	6.48
DB	FUJITSU MPE3064AT	IDE	12672450	6.48
E1	QUANTUM ATLAS10K2-TY092J	SCSI	17938985	9.18
E2	QUANTUM ATLAS10K2-TY092J	SCSI	17938985	9.18
E3	QUANTUM ATLAS10K2-TY092J	SCSI	17938985	9.18
E4	QUANTUM ATLAS10K2-TY092J	SCSI	17938985	9.18
E6	SEAGATE ST318404LC	SCSI	35843670	18.35
E7	SEAGATE ST318404LC	SCSI	35843670	18.35
E8	SEAGATE ST318404LC	SCSI	35843670	18.35
EA	SEAGATE ST39204LC	SCSI	17921835	9.17
EB	SEAGATE ST39204LC	SCSI	17921835	9.17
EC	SEAGATE ST39204LC	SCSI	17921835	9.17
F1	QUANTUM SIROOCO1700A	IDE	3335472	1.70
F5	IBM-DTLA-307020	IDE	40188960	20.57
F6	IBM-DTLA-307020	IDE	40188960	20.57

2.6 Test Configurations

The host computer and hard drive setup were determined by the test case parameters. Either two or three disks were required for each test case. A *source* and *destination* were

required for all test cases. A *media* disk was required for *image* operations. The source disk provided something to copy. The destination disk provided a place to put the copy. The media disk provided a place to put the image file for test cases that require the creation of an image file. One of two DOS Boot floppies was selected and then used to create the run-time environment for the test case and contained control scripts, and log files. Either a PC DOS 6.3 or Windows 98 DOS environment was selected. The Windows 98 DOS environment was required for test cases with an operation on a FAT32 partition. A CD-ROM contained the support software, a copy of SafeBack and utility software. The support software provided for setup of test data, measurement of test results, and control of the test process.

The type of BIOS required for the test case determined the selection of the host computer. If an extended BIOS was required then one of **Cadfael, Rumpole, Wimsey,** or **JudgeDee** was selected. If a legacy BIOS was required then one of the Nexar computers was selected. If a copy operation on equal sized SCSI disks was required then **Marple** was selected.

The factors determining the source disk selection were the source disk interface and type of partition to copy. A disk was selected with the matching interface and a partition of the type required for the test case. The factors for the selection of the destination drive were the destination interface and the relative size parameters. A drive was selected with the specified interface and, for whole disk copies, size relative to the source. For partition copies, the actual size of the destination drive did not matter since it was the size of the partition on the destination that was relevant. After the source and destination drives were selected, the media disk was selected for one of the two available drive slots.

The 12 system hard drive configurations used for the tests are presented in Table 2-3. The *Source* column indicates where the source drive was mounted. Only the primary IDE channel was used. The drive was usually positioned as *drive 0.* SCSI source drives were set to SCSI ID 0. The *Destination* column indicates the positioning of the destination drive. The *Media* column indicates the positioning of the media drive.

Configurations 1-10 were used for the entire duration of a test case. Configurations 11 and 12 were used when only two disk drive slots were available on the host computer. The media disk was swapped with either the source or destination disk as required for the step of the test case execution. If an image file was to be created then only the source and media disk were installed. If the image was to be restored to the destination then the source drive was replaced by the media drive. If the source was to be compared to the destination then the media drive was not installed.

Table 2-3 System Configurations

ID	Source	Destination	Media
1	IDE primary 0	IDE primary 1	SCSI ID 0
2	SCSI ID 0	SCSI ID 1	IDE primary 0
3	IDE primary 0	IDE primary 1	Jaz
4	SCSI ID 0	IDE primary 1	IDE primary 0
5	IDE primary 0	SCSI ID 0	IDE primary 1
6	SCSI ID 0	SCSI ID 1	Jaz

ID	Source	Destination	Media
7	IDE primary 0	IDE primary 1	none
8	IDE primary 0	SCSI ID 1	none
9	SCSI ID 0	IDE primary 1	none
10	SCSI ID 0	SCSI ID 1	none
11	IDE primary 0	IDE primary 1	IDE primary 1, IDE primary 0
12	SCSI ID 0	SCSI ID 1	SCSI ID 1, SCSI ID 0

2.7 Support Software

Support software, FS-TST Release 1.0, was developed to support the testing of disk imaging tools. FS-TST Release 1.0 can be obtained from the http://www.cftt.nist.gov web site. The support software serves five main functions; initialization of a disk to a known value [DISKWIPE], comparison of a source with a destination [DISKCMP, PARTCMP, ADJCMP, SECCMP], detection of changes to a disk [DISKHASH, SECHASH], corruption of an image file [CORRUPT] and simulation of a faulty disk [BADDISK and BADX13]. All programs were written in ANSI C (except for BADDISK and BADX13) and compiled with the Borland C++ compiler version 4.5. BADDISK and BADX13 were written in assembler language and compiled with Borland Turbo Assembler version 5.0.

For these test cases, version 3.2 of BADDISK was used, not the version 3.1 included in FS-TST 1.0. Version 3.2 can be obtained from the CFTT website as an update to FS-TST Release 1.0. In addition to the FS-TST software, one of two DOS boot floppies, either a PC DOS 6.3 or Windows 98 DOS, was used to create the run-time environment for the test case. The Windows 98 DOS boot disk was required for test cases with an operation on a FAT 32 partition.

3 Media Setup

The test cases required several media components to be created before the test cases could be executed. The following items were created.

1. Source hard disk drives for the test cases.
2. A DOS boot floppy that contains control scripts, log files and creates the run-time environment for the test case setup and measurement.
3. A Windows 98 DOS boot floppy alternate environment.
4. ZIP disk with **partition magic** and space for removable media tests.
5. A CD that contains support software, and utility software.

In addition to the components that were setup once, a destination hard drive was setup for each test case.

3.1 Source Disks

There were too many possible disk layouts for all to be used in the tests. Five configurations were selected that covered the most common partition types. The first configuration was a dual boot Red Hat Linux 7.1 and Windows Me. This configuration

also included FAT16, Linux EXT2, hidden partitions and deleted partitions (Table 3-1). The second configuration was a Windows 2000 system with both FAT32 and NTFS file systems (Table 3-2). The third configuration did not contain a valid partition table. The fourth configuration was a single FAT12 partition and the fifth configuration was a single FAT32 partition. All partitions were created with **partition magic Pro 6.0**. Table 3-6 documents the setup assignments for each source drive.

Table 3-1 Windows Me/Linux Source Drive Layout

Type	Size (MB)	Comment
FAT16	600	Windows Me C drive
none	500	Unallocated Space
Extended	3500	Extended partition containing the next four partitions
EXT2	100	Linux EXT2 partition
FAT16	70	D drive for Windows
FAT16	2000	A FAT16 partition that has been deleted
FAT16	90	A FAT16 partition marked as *hidden*
EXT2	3000	Linux EXT2 partition with Red Hat 7.1
none	variable	Unallocated space up to the next partition
SWAP	200	Linux Swap partition

Table 3-2 Windows 2000 Source Drive Layout

Type	Size (MB)	Comment
FAT32	3000	Windows 2000 C drive
none	1000	Unallocated space
Extended	variable	Remainder of disk space
NTFS	1000	Deleted NTFS partition
NTFS	600	D drive
FAT32	1000	Deleted FAT32 partition
NTFS	800	Hidden NTFS partition
none	variable	Unallocated space up to next partition
FAT32	600	Hidden FAT32 partition

The setup procedure for a source disk was as follows:

1. Selected type of setup: Windows Me/Linux, Windows 2000 or none. Disks E3 and F5 were given the Windows Me/Linux layout; disks E4 and F6 were given the Windows 2000 Layout; disk CC was setup without a partition table.
2. Selected a hard drive
3. Selected computer, installed drive, booted into PC DOS 6.3 from a boot floppy.
4. Ran **LOGSETUP** to make a record of the setup.
5. Ran **DISKWIPE** to initialize the drive contents.
6. If the setup type uses an operating system the following two steps were completed

7. Ran **partition magic** to partition the drive. For Windows Me/Linux source disk the script in Table 3-3 was used and for Windows 2000 source disk the script in Table 3-4 was used.
8. The installation instructions for each operating system were followed For a Windows Me/Linux configuration, Windows Me was first installed then Linux. For a Windows 2000 configuration, Windows 2000 was installed.
9. Deleted files were created using a script (DOS batch file) that created a directory (X:\UDT, where X is a drive letter) with deleted files and a deleted subdirectory (Table 3-5).
10. Ran **DISKHASH** to create a reference SHA-1 hash for the source disk.
11. (optional) A backup to another disk was created for some of the source disks that could be used to restore the disk if it were modified. If a disk needed to be restored, the hash value could be recomputed to verify that the backup and restore were successful.

Table 3-3 Partition Magic script for Windows Me/Linux Source (`FAT-SRC.TXT`)

```
Select Drive 1
Select Unallocated First
Create /FS=FAT /Size=600 /Label="P1FAT"
Select Unallocated First
Create /FS=Extended /Size=4000
Select Partition Extended
Resize Left Boundary Smaller 500
Select Unallocated 2
Create /FS=LINUXEXT2 /Size=100 /Label="X1Unix"
Select Unallocated 2
Create /FS=FAT /Size=70 /label="X1Fat"
Select Unallocated 2
Create /FS=FAT /Size=2000 /label="GONE"
Select Unallocated 2
Create /FS=FAT /Size=90 /label="GHOST"
Select Unallocated 3
Create /FS=LINUXEXT2 /Size=3000 /Label="Unix"
Select Unallocated 3
Create /FS=LINUXswap /Size=200  /Position=END
Select Partition "GHOST"
Hide
Select Partition "GONE"
Delete "GONE"
Select Partition "P1FAT"
Set Active
```

Table 3-4 Partition Magic script for Windows 2000 Source (`NT-SRC.TXT`)

```
Select Drive 1
Select Unallocated First
Create /FS=FAT32 /Size=3000 /Label="FAT3GB"
Select Unallocated First
Create /FS=Extended
Select Partition Extended
Resize Left Boundary Smaller 1000
Select Unallocated 2
```

```
Create /FS=NTFS /Size=1000 /label="GONE1"
Select Unallocated 2
Create /FS=FAT32 /Size=600 /Label="GHOST32" /position=end
Select Unallocated 2
Create /FS=NTFS /Size=600 /label="X1NT"
Select Unallocated 2
Create /FS=FAT32 /Size=1000 /label="GONE2"
Select Unallocated 2
Create /FS=NTFS /Size=800 /label="GHOST4NT"
Select Partition "GHOST4NT"
Hide
Select Partition "GHOST32"
Hide
Select Partition "GONE2"
Delete "GONE2"
Select Partition "GONE1"
Delete "GONE1"
Select Partition "FAT3GB"
Set Active
```

Table 3-5 Script to Create Deleted Files (`UDT-SET.BAT`)

```
echo undelete test setup
Rem Setup a directory with some deleted files and a deleted
subdirectory
date
time
:L1
Rem are we done?
      if "%1"=="" goto L1X
      echo "Set up drive %1:"
rem   create a directory for the deleted files
      mkdir %1:\udt
rem   create two files
      copy a:readme.txt %1:\udt
      copy a:back.txt %1:\udt
rem   delete one file
      del %1:\udt\back.txt
rem   undelete %1:\udt
rem   create a subdirectory
      mkdir %1:\udt\sub
Rem   create some files in the subdirectory
      copy a:missing.txt %1:\udt\sub
      copy a:gone.txt %1:\udt
rem   delete one file
      del %1:\udt\sub\missing.txt
rem   delete the directory
      rmdir %1:\udt\sub
rem   delete another file
      del %1:\udt\gone.txt
rem   shift cmd line, look for another drive
      shift
      goto L1
:L1X
echo Setup finished
```

Table 3-6 Source Drive Setup Assignments

Soruce Setup	Assigned Drives
Win Me/Linux FAT16/EXT2	E3 F5 1E A1 A3 B5 B6 EC B8 F1
Win 2000 FAT32/NTFS	E4 F6
No OS	C0
Single FAT12	A4
Single FAT32	A5

3.2 DOS Boot Floppy

Two DOS boot floppies were used. This section describes the setup of a PC DOS 6.3 boot floppy. A second boot floppy created in the Windows 98 environment was setup in a similar fashion and also used. The DOS floppy disk provided the execution environment for the support software. The commands to setup the DOS floppy are presented in Table 3-7. The DOS boot floppy was used for source drive setup, destination drive setup and for measuring the results of a test run.

Table 3-7 DOS Boot Floppy Setup Procedure

From a PC DOS 6.3 System, insert a blank floppy disk

```
FORMAT A: /S
MKDIR A:\ASPI
MKDIR A:\GUEST
MKDIR A:\MISC
COPY HIMEM.SYS A:\MISC
COPY MSCDEX.EXE A:\MISC
COPY MOUSE.COM A:\MISC
COPY MOUSE.INI A:\MISC
COPY SMARTDRV.EXE A:\MISC
COPY GUEST.EXE A:\GUEST
COPY GUEST.INI A:\GUEST
COPY GUESTHLP.TXT A:\GUEST
COPY ASPI8U2.SYS A:\ASPI
COPY ASPICD.SYS A:\ASPI
```

setup AUTOEXEC.BAT and CONFIG.SYS

The AUTOEXEC.BAT file used in testing is presented in Table 3-8 and is a simplified version of a typical forensic boot floppy based on the recommendations in the SafeBack 2.0 manual.

Table 3-8 DOS AUTOEXEC.BAT

```
@ECHO OFF
PROMPT $p$g
A:\misc\MOUSE
A:\misc\smartdrv
a:\misc\mscdex.exe /d:aspicd0 /L:Z /m:12
a:\guest\guest letter=x
```

The contents of the CONFIG.SYS used in testing is presented in Table 3-9.

Table 3-9 DOS CONFIG.SYS

```
device=A:\misc\himem.sys
dos=high,umb
lastdrive=z
FILES = 30
BUFFERS = 8
device=a:\aspi\aspi8u2.sys /D /PD800 /Q9
device=a:\aspi\aspicd.sys /d:aspicd0
```

3.3 Windows 98 Boot Floppy

The Windows 98 boot floppy disk provided an alternate execution environment for the support software and **SafeBack** execution. The commands used to setup the floppy are presented in Table 3-12. The **io.sys** file on the boot floppy was scrubbed of any references to the C: drive and the following programs: **DBLSPACE.BIN, DRVSPACE.BIN**, and **STACKER**.

Table 3-10 Windows 98 Boot Floppy AUTOEXEC.BAT

```
@ECHO OFF
A:\drivers\mscdex.exe /D:mscd001 /L:Z
a:\guest\guest letter=x
```

The **CONFIG.SYS** file loads drivers for ASPI SCSI devices, CDROM drive and sets the last drive letter.

Table 3-11 Windows 98 Boot Floppy CONFIG.SYS

```
ACCDATE=C- D- E- F- G- H- I- J- K- L- M- N- O- P- Q-
device=A:\drivers\himem.sys /testmem:off
device=A:\drivers\oakcdrom.sys /D:mscd001
device=A:\drivers\aspi8u2.sys /D /PD800 /Q9
device=A:\drivers\aspi8dos.sys
device=a:\drivers\aspicd.sys /D:mscd001
files=30
buffers=10
dos=high,umb
stacks=9,256
lastdrive=z
```

Table 3-12 Windows 98 Boot Floppy Setup

From a Windows 98 System, insert a blank floppy disk and open a DOS command window

```
FORMAT A: /S
MKDIR A:\GUEST
MKDIR A:\DRIVERS
COPY HIMEM.SYS A:\DRIVERS
COPY MSCDEX.EXE A:\DRIVERS
COPY MOUSE.COM A:\DRIVERS
```

```
COPY MOUSE.INI A:\DRIVERS
COPY SMARTDRV.EXE A:\DRIVERS
COPY GUEST.EXE A:\GUEST
COPY GUEST.INI A:\GUEST
COPY GUESTHLP.TXT A:\GUEST
COPY ASPI2DOS.SYS A:\DRIVERS
COPY ASPI4DOS.SYS A:\DRIVERS
COPY ASPI8DOS.SYS A:\DRIVERS
COPY ASPI8U2.SYS A:\DRIVERS
COPY ASPICD.SYS A:\DRIVERS
COPY BTCDROM.SYS A:\DRIVERS
COPY BTDOSM.SYS A:\DRIVERS
COPY FLASHPT.SYS A:\DRIVERS
COPY OAKCDROM.SYS A:\DRIVERS
setup AUTOEXEC.BAT and CONFIG.SYS
```

The **Scrub Log** is a list of changes made to the **IO.SYS** file to ensure no references to the C: drive during boot.

Figure 3-1 Windows 98 Boot Floopy Scrub Log

```
A:\SCRUB.EXE compiled at 23:47:10 on Nov  7 2001
C:\ found at 17268, replaced by A:\
C:\ found at 17664, replaced by A:\
C:\ found at 20484, replaced by A:\
C:\ found at 20499, replaced by A:\
C:\ found at 42632, replaced by A:\
C:\ found at 42667, replaced by A:\
DBLSPACE.BIN found at 42720, replaced by NOOSPACE.BIN
DRVSPACE.BIN found at 42734, replaced by NOOSPACE.BIN
C:\ found at 60980, replaced by A:\
C:\ found at 60996, replaced by A:\
C:\ found at 66795, replaced by A:\
C:\ found at 66805, replaced by A:\
C:\ found at 69755, replaced by A:\
DBLSPACE.BIN found at 60999, replaced by NOOSPACE.BIN
DRVSPACE.BIN found at 60967, replaced by NOOSPACE.BIN
STACKER found at 61013, replaced by SLACKER
Starting Windows 98 found at 68272, replaced by Starting NIST Fboot
C:\ found at 220112, replaced by A:\
222390 bytes read from io.sys
```

3.4 Jaz Disk

The Jaz disk served a number of functions. The utility programs (e.g., **partition magic**) were located on the Jaz disk and the Jaz disk served as removable media for tests that required removable media. The disk was setup by placing a copy of **pqmagic.exe** on the disk.

3.5 CD-ROM

The CD-ROM was created using **Easy CD Creator 5** and was divided into the following directories used in testing **SafeBack**:

- SB: Contained a copy of **SafeBack MASTER** program.

- SS: Contained a copy of FS-TST support programs.

In addition, the CD contained the following additional directories not used in testing:
- PM: Contained a copy of **Partition Magic**.
- NU: Contained some commercial utility programs.
- Tools: Contained other DOS utility programs.

4 Test Execution Scripts

Each test case had a unique set of DOS batch files (i.e., scripts) that guide the test operator through each phase of the test. The scripts for each test case were similar with minor differences for the specific requirements of a given test case. In general each test case was executed as a sequence of scripts. The test case was started from an initial script that records basic information and generates a script to call the next script in the sequence. As each script is executed a new script is generated to perform the next step. The scripts are described in Table 4-1.

Table 4-1 Test Scripts

Script Name	Description and Comments
ST-nnn	Start test case nnn. Run **logcase** to record basic information such as drive assignments about the test case. Generate scripts DST (to call DST-nnn), IMG (to create a media drive if needed) and NEXT to execute the next script (CP-nnn).
DST-nnn	Create the destination drive by running **diskwipe**. If the test requires a partition on the destination, create the partition.
IMG-nnn	Run **diskwipe** and create a large partition to contain an image. The script is only used for a few cases to create a media drive as needed. After a media drive is created it can be used for several cases.
CP-nnn	If the test case requires an I/O error run **baddisk** or **badx13** to setup the simulated I/O error. Run **SafeBack**. Run **partab** to log the source drive partition table. Create NEXT to run the next script. If the test is a copy case the next step is CMP-nnn, otherwise the next step is RS-nnn.
RS-nnn	Run **partab** to log the destination drive partition table. If the test case requires an I/O error run **baddisk** or **badx13** to setup the simulated I/O error. Run **SafeBack**. Create NEXT to run the CMP-nnn script.
CMP-nnn	Compare the source to the destination. Use either **adjcmp**, **diskcmp** or **partcmp** as needed for the test. Create the HASH script.
HASH	Run **diskhash** to check the source drive for changes. Create a directory for log files and copy the log files to the directory.
NEXT	Created dynamically as a link to the next step.

For example, consider test case DI-108. The test case is to create an image of a FAT32 partition from an IDE hard drive accessed via an extended BIOS with a simulated read error during image creation, and then to restore the image to a partition of the same size on an IDE hard drive.

The first script executed for case 108 was ST-108 in Figure 4-1. The script takes five command line parameters as follows:

Parameter	Value
%1	Host computer name
%2	Operator identification
%3	Source drive label
%4	Destination drive label
%5	Media drive label

The function of each step is as follows:

1. Turn off command echo.
2. Comment.
3. Comment.
4. Set the current directory to the floppy disk.
5. Delete all text files to remove any log files from other test cases.
6. Record information about the case.
7. Record the DOS version in the case log file.
8. Record the script version in the case log file.
9. Create a batch file to setup the destination drive (calls DST-108).
10. Create a batch file to execute the next step (CP-108).
11. Create a batch file to setup a media drive.

Figure 4-1 Script ST-108 Start the Case

```
1.  @ECHO OFF
2.  REM Host Operator Src Dst Boot/Media
3.  REM Script Version Nov  6 2001 10:10:33
4.  cd A:\
5.  del a:\*.txt
6.  Z:\ss\logcase DI-108 %1 %2 80:%3 81:%4 82:%5
7.  ver >> A:\CASE.TXT
8.  echo Script Version: Nov  6 2001 10:10:33 >> A:\CASE.TXT
9.  echo A:\s\dst-108 %1 %2 %3 %4 %5 >A:\dst.bat
10. echo A:\s\cp-108 %1 %2 %3 %4 %5 >A:\next.bat
11. echo A:\s\img-108 %1 %2 %3 %4 %5 >A:\img.bat
```

The second script to execute is DST-108 (Figure 4-2). The second script is executed from the DST batch file.

1. Comment.
2. Turn off command echo.
3. Run the **diskwipe** program to write a fill pattern to the entire disk.
4. Run **partition magic** to create a partition on the drive.

Figure 4-2 Script DST-108 Setup Destination Drive

```
1.  REM Script Version Nov  6 2001 10:10:33
2.  @ECHO OFF
3.  Z:\ss\diskwipe DI-108 %1 81 %4 /noask /dst /new_log /comment "%2"
```

```
4. X:\PM\PQMAGIC /cmd=X:\PM\D32X-ES.txt
```

The script in Figure 4-3 is used by DST-108 to create a FAT32 partition of 3000 MB.

Figure 4-3 Partition Magic Script to Create a FAT32 Partition

```
1. Select Drive 1
2. Select Unallocated First
3. Create /FS=FAT32 /Size=3000 /Label="FAT32_EQ"
```

If a media drive is needed, the script in Figure 4-4 is used to setup a media drive.

1. Comment.
2. Turn off command echo.
3. Run the **diskwipe** program to write a fill pattern to the entire disk.
4. Run **partition magic** to create a partition on the drive.

Figure 4-4 IMG-108 Script to Setup a Media Drive

```
1. REM Script Version Nov  6 2001 10:10:33
2. @echo off
3. Z:\ss\diskwipe DI-108 %1 80 %4 /noask /media /new_log /comment "%2"
4. X:\PM\PQMAGIC /cmd=X:\PM\img-X.txt
```

The script in Figure 4-5 is run from the batch file NEXT.BAT created by ST-108.

1. Comment.
2. Turn off command echo.
3. Log the contents of the source drive partition table.
4. Create NEXT.BAT. The next step is to restore the image file to the destination drive.
5. Setup the error simulation.
6. Prompt to operator.
7. Prompt to operator.
8. Wait for operator.
9. Run **SafeBack**.
10. Prompt to operator.
11. Prompt to operator.

Figure 4-5 CP-108 Create an Image File

```
1. REM Script Version Nov  6 2001 10:10:33
2. @ECHO OFF
3. Z:\ss\partab DI-108 %1 80 /all /new_log /comment %2(%4)
4. echo A:\s\rs-108 %1 %2 %3 %4 %5 >A:\next.bat
5. call A:\err\err-108
6. echo ready to run safeback for case 108
7. echo "image XBIOS-IDE FAT32 src = dst"
8. pause
9. Z:\sb\master
10. echo Shutdown and reboot
11. echo after reboot: A:\next
```

The script in Figure 4-6 is run from the batch file NEXT.BAT created by CP-108.

1. Comment.
2. Turn off command echo.
3. Log the destination drive partition table.
4. Setup NEXT.BAT to compare source to destination.
5. Prompt to operator.
6. Wait for operator.
7. Run **SafeBack**.
8. Prompt to operator.
9. Prompt to operator.

Figure 4-6 RS-108 Restore the Image to a Destination

```
1. REM Script Version Nov  6 2001 10:10:33
2. @echo off
3. Z:\ss\partab DI-108 %1 81 /all /new_log /comment %2(%4)
4. echo A:\s\cmp-108 %1 %2 %3 %4 %5 >A:\next.bat
5. echo Safeback restore case 108
6. pause
7. Z:\sb\master
8. echo Shutdown and reboot
9. echo after reboot: A:\next
```

The script in Figure 4-7 is run from the batch file NEXT.BAT created by RS-108.

1. Comment.
2. Turn off command echo.
3. Compare the source partition to destination partition.
4. Create HASH.BAT to hash the source drive.
5. Add to HASH.BAT, create a log directory for test case DI-108
6. Add to HASH.BAT, copy log file to case directory.
7. Add to HASH.BAT, prompt for operator.
8. Prompt for operator.

Figure 4-7 CMP-108 Compare the Source to the Destination

```
1. REM Script Version Nov  6 2001 10:10:33
2. @ECHO OFF
3. Z:\ss\partcmp DI-108 %1 80 %3 81 %4 /new_log /comment "%2" /select 1 1
4. echo Z:\ss\diskhash DI-108 %%1 80 /comment %2(%3) /new_log /after >A:\hash.bat
5. echo mkdir a:\DI-108 >> A:\hash.bat
6. echo copy A:\*.TXT a:\DI-108 >> A:\hash.bat
7. echo echo Case: DI-108 finished >> A:\hash.bat
8. echo ready to hash for case DI-108
```

5 Test Case Execution

This section presents the procedures used for running the test cases. It is assumed that the reader is familiar with basic computer operation.

5.1 Execution Procedure

The procedure to execute a test case was as follows:

1. Select the test case to run.
2. Collect removable media: DOS Boot floppy, Jaz disk and CD-ROM.
3. Select a source disk based on the test case parameters. The *source interface* parameter determines if the disk is IDE or SCSI. If a partition type is specified then a source disk with the specified setup is selected.
4. Select a destination disk for the test based on the test case parameters. The *destination interface* determines if the disk is IDE or SCSI. The *relative size* parameter determines acceptable selections for test cases that operate on an entire disk. For partition operations, any size disk can be used.
5. Select a system configuration (Table 2-3) based on the *source interface* and *destination interface* test case parameters.
6. Select a host computer to run the test. Ensure that the BIOS boot order is set as required by the selected system configuration.
7. Select a media disk based on the selected system configuration.
8. Ensure that the host computer is off. Install DOS boot disk, Jaz disk, media disk and destination disk.
9. Turn on the host computer to boot from forensic DOS floppy.
10. Run the ST-nnn script (nnn is the test case number).
11. Setup the destination drive with the DST script.
12. If the media drive needs to be setup, run the IMG script.
13. Turn off the system.
14. If this is a copy test case, install the source and destination drives, boot the system and run NEXT to setup any I/O error and to execute **SafeBack**. After **SafeBack** is finished, shutdown the system.
15. If this is an image test case, install the source and media drives, boot the system and run NEXT to setup any I/O error and to execute **SafeBack** (backup option). After the backup is complete, shutdown the system. Remove the source drive and install the destination drive, run NEXT to setup any I/O error, to corrupt the image file (if required), and to execute **SafeBack** (restore option). After **SafeBack** is finished, shutdown the system.
16. Install source and destination drives. Insert the DOS boot floppy.
17. Turn on the host computer to boot into DOS from the boot floppy.
18. Run the post-execution script by running NEXT to measure the results. The script compares the source to the destination and computes a SHA-1 for the source disk.
19. After the measurement script finishes, the log files should be copied to a permanent location.

5.2 Guide to examination of Log Files

After a test case is finished the results are contained in a set of log files that are located in a directory named **DI-xx** (**xx** is the test case number). Each of the support programs executed in the setup and measurement steps produces a log file that can be examined. For source drive setup log files, there is a directory named **setup** with a subdirectory for each source disk. Within each subdirectory are log files from the setup of the corresponding source disk. There are log files from the execution of **logsetup**, **diskwipe** and **diskhash**.

The remainder of this section discusses the relevant content of the log files produced by each support program used in testing. The FS-TST documentation contains detailed descriptions of the complete log file content.

5.2.1 LOGSETUP: Setup a Source Drive

Administrative details about the setup of a source disk drive are recorded in the log file, **SETUP.TXT**, from **logsetup**. The disk drive label, host computer used, operator, operating system loaded (if any) and date are recorded.

5.2.2 LOGCASE: Start a Test Case

Administrative details about the execution of a test case are recorded in the log file, **CASE.TXT**, from **logcase**. The labels of the disk drives used, the role assigned each disk, the BIOS drive number for each disk, host computer used, operator, and date are recorded.

5.2.3 BADDISK and BADX13: Simulate I/O Errors

These programs intercept interrupt 13 to simulate disk I/O errors. BADDISK is a *terminate and stay resident* (TSR) program that monitors the interrupt 13 disk interface for a given disk address and command. BADDISK usually passes an I/O request on to the BIOS interface, however if the I/O request is for the given address and command a specified error result code is returned instead. BADX13 is a version for use on large disks that require the interrupt 13 extensions. The log file repeats the command line parameters.

5.2.4 CORRUPT: Modify an Image File

CORRUPT is used to corrupt an image file by changing a single selected byte in an image file. CORRUPT logs the original content of the selected byte, the replacement value, the index of the byte within the file and the file name.

5.2.5 PARTAB: Document partition tables

The **partab** program documents the partition tables of the source and destination disk drives. The log file for the source disk should show that the drive has one of the initial setups from Section 3.1. The log file for the destination drive should show that for an operation on an entire disk drive there is no partition table, but that for an operation on a partition there is a partition of the type required by the test case on the destination drive.

5.2.6 ADJCMP, DISKCMP and PARTCMP: Check Accuracy of Duplicate

The comparison programs, **adjcmp**, **diskcmp** and **partcmp**, have two functions: measure the accuracy of the duplication of the source to the destination and for destinations larger than the source, and determine if **Safeback** has changed any of the excess sectors.
To measure the accuracy of the duplication, two values from the log file are relevant. The value labeled **Sectors compared** indicates the number of sectors checked and the value labeled **Sectors differ** indicates the number of sectors that are not as expected. If there is a small number of sectors that do not match, the LBA addresses of the non-matching sectors is reported under **Diffs range**. The non-matching sectors can be examined in detail with the **seccmp** program. For **diskcmp** the LBA addresses are relative to the beginning of the disk, for **partcmp** the LBA addresses are relative to the beginning of the partition.

To determine if the tool has changed the content of the excess sectors the comparison programs categorize the excess sectors of the destination. The evaluation of the categorization of the excess destination sectors is simple in the case of a FAT partition, but has a complication for NTFS and Linux EXT2 partitions. In the case of a FAT partition, all the excess sectors should be categorized as *destination fill*. The number of destination sectors is the value labeled **fewer sectors**. This value should match the value labeled **Dst Byte fill**.

The **adjcmp** program was used for test cases that adjusted restored partitions to cylinder boundaries. The **adjcmp** program compares each pair of partitions in similar fashion to **partcmp** and then does an excess sector analysis on any areas of unallocated space. A summary of disk space usage and partition comparison appears at the end of the log file.

5.2.7 SECCMP: Investigate Anomaly

The **seccmp** program is not part of the usual test procedures. However, it is used in some test cases where either **diskcmp** or **partcmp** indicated that some sector of a source or destination did not match the corresponding sector from the copy operation. The **seccmp** produces a list of differences between two specified sectors.

5.2.8 DISKHASH: Verify no Change to Source

The **diskhash** program is used to verify that a source disk has not been changed by the tool. The **diskhash** log files contains a SHA-1 hash value. The verification is accomplished by comparing the hash value from the test case log file, **HASHALOG.TXT**, to the hash value from the source disk setup, **HASHBLOG.TXT**. If the value agree, then the tool has not changed the source disk.

5.3 Results Evaluation Procedure

After a test case was run, the results were examined to determine if the results should be accepted or if some further actions were required to complete the test case. The evaluation of results determines if the apparent results are an accurate reflection of the

tool under test. Either a successful or unsuccessful test outcome was reviewed to ensure that an error in the testing process had not occurred.

The first issue was if a test appeared to be successful should we accept the result that the tool has produced the expected result for the particular test case. There are several ways that the test could appear to produce expected results without actually doing so. This would usually involve entire steps not running and the measurement of disks that are left in the final state of an earlier successful test. This was mitigated by always ensuring that the destination disk was wiped at the beginning of each test. The **diskwipe** log file was reviewed to verify that the correct number of sectors were wiped for the given destination disk.

The second issue was if a test produced an anomaly and appeared to fail, has the tool failed or is something else wrong. Each anomalous test run was reviewed to characterize the anomaly and then a course of action was selected.

1. If a hardware or procedural problem was found, e.g., disk drive has failed, or improper configuration for the test, then the test was rerun with appropriate adjustments.
2. If no hardware or procedural problem was found and the anomaly matched a known anomaly then we accepted the anomaly as genuine.
3. If the anomaly was unique then a decision was deferred until more test cases were been run. These test cases are referred to as *defer until more*.
4. If the anomaly matched an anomaly in the *defer until more* category then both results were examined for common factors. If sufficient common factors were found, a new *known anomaly* would have been established, or the test cases remained *defer until more*.

After all test cases were run any test cases remaining in the *defer until more* category would be resolved by either accepting the anomaly as genuine or reclassified based on additional investigation.

6 Adapting to a Different Test Environment

The tests were conducted in the CFTT lab at NIST. An attempt to reproduce the test results in another lab may require significant adjustments to the test scripts and procedures. This section gives guidelines for adapting the support components of the test cases for other lab environments. Hardware, source hard drives, execution environments, and test scripts are discussed in turn.

6.1 Hardware

The available hardware determines the strategy for organizing the test process. At a minimum, six disk drives are required. Three should be IDE drives such that two of the drives are different sizes and the third drive is the same size as one of the other two drives. The other three disk drives should be SCSI drives with the same size relationship. In addition, one of the SCSI drives should be larger than at least one of the IDE drives and one of the IDE drives should be larger than at least one of the SCSI drives. The drives could be mounted (not removable) in one computer or as in the CFTT lab at NIST, each hard drive can be removed from one computer and placed in another.

Independent replication is accomplished by an agency or lab other than NIST repeating each test case in their own lab environment. Since it is unlikely that the exact hardware used by NIST is present, adjustments and substitutions must be made to run the test cases. For example, the NIST environment used Iomega Jaz drives to contain some software tools accessed as drive X. Another lab that does not have Jaz drives available might put the tools on floppy disks, LS-110 drives (SuperDisk) or CD-ROM. Even more significant are the actual hard drives used in the tests. It is not always clear what would be an equivalent substitution for a hard drive used.

There is an important issue about replication here. What should be done if an anomaly actually depended on some condition that was not one of the actual test parameters. Replication of the anomaly would depend on this triggering condition. There is no easy answer to this issue. The *ad hoc* answer is to require any substitution to conform to any conditions determined after the test is run that are required to replicate the result. Information learned during the test process should be applied to any attempted replication. It may be the case that a hidden test parameter is discovered during testing. Any attempt to replicate the tests must use this hidden parameter.

6.2 Source Hard Drives

The more hard drives available the easier it is to organize and setup source drives in advance. If there are only a few drives available then source drives may need to be repeatedly setup as the drive is moved into different roles. This can take a significant amount of time. This can be mitigated by a careful selection of test case order such that once a source drive is setup all the test cases that require that drive are run before assigning the drive to another role.

6.3 Execution Environments

There are two execution environments used for the test cases. The support software that does test setup and results measurement runs in a DOS environment.

The DOS environment is established from a DOS boot disk. The boot disk should be similar to the one described in section 3.2, except for changes to reflect the actual hardware present. For example, if no Jaz drive is present all files in the **guest** directory and references to **guest** in the **autoexec.bat** file can be deleted.