

# Using the TestDisk Recovery Utility to Recover a FAT32 'Lost' Partition

Copyright©2003-2005 by Daniel B. Sedory

**TestDisk** is a GNU Public Licensed OpenSource Program.

Get the latest \*.zip version available here:

 <http://www.cgsecurity.org/testdisk.html>

**TestDisk** will soon be at version **5.8**

This page discusses the use of **TestDisk** to recover a partition that's been accidentally **deleted** (using a utility such as [DELpart](#), FDISK or any other program that has overwritten only a drive's **MBR** sector, EBR sectors or the whole first track -- as [ZAP63](#) does). If any area of both copies of a partition's **FAT** have *already* been overwritten by a [Format](#) program (or by [MS-FDISK](#)), chances are that TestDisk will **not** be able to return the partition to its former state; you'd need to use a data recovery program on subdirectories instead. If any of the drive's **Boot Records** have been overwritten (as in the case of CIH virus attacks), it would require someone skilled in the use of a disk editor or other tools to *manually* rebuild deleted FAT32 Boot Records and use a 2nd copy of the **FAT** to restore the original, *before* TestDisk could be successfully employed.

Although TestDisk can be run from a hard drive (under **Linux** or a **Windows 9x** OS), it is recommended that you do so only if the target drive (the one having lost partitions) was formatted on the same computer. If your computer has only one bootable partition (the 'lost' partition), then boot the computer with a DOS/Win9x boot disk and run TestDisk from a floppy diskette (the file **cwstdpmi.exe** must also be on the same floppy disk to give TestDisk the necessary 32-bit capabilities under DOS). This method is not only safer (as far as Geometry and the BIOS is concerned), but saves you all the hassles of trying to hook up the target drive as a slave on some other computer! [Experts understand there are various methods used to translate "drive geometry" by different BIOS manufacturers. TestDisk has a "Geometry" function for them.]

When TestDisk is first executed, it will output the phrase: **Please wait...** until it has gathered enough data about the target drive to display something similar to this:

```
TestDisk 4.4, Data Recovery Utility, March 26 2003
Christophe GRENIER <grenier@cgsecurity.org>
http://www.cgsecurity.org
Dos version

Disk 80 - CHS 812 32 63 - 799 MB (Mxd BIOS mode)

[Analyse ] [ Delete ] [ MBR Code ] [ Geometry ] [ Options ] [ Advanced ]
[ Quit ]

Analyse current partition structure and search for lost partitions_
```

**Fig.1: TestDisk's Main Screen (not to scale).**

If more than one drive is attached, you would first use the up and down arrow keys to select the target drive, then press the ENTER key while "[ **Analyse** ]" is highlighted to get this:

```
TestDisk 4.4, Data Recovery Utility, March 26 2003
Christophe GRENIER <grenier@cgsecurity.org>
http://www.cgsecurity.org

Disk 80 - CHS 812 32 63 - 799 MB (Mxd BIOS mode)
Check current partition structure
      Partition                Start                End                Size
No partition is bootable

Quit

*=Primary bootable  P=Primary  L=Logical  E=Extended  D=Deleted
Quit this section_
```

**Fig.2: "Check Current Partition Structure" (not to scale).**

This shows that the MBR currently has **no** Partition Table entries in it; which is what we expected having deleted the only entry before running TestDisk.

**If** you see a display like this instead:

```
TestDisk 4.4, Data Recovery Utility, March 26 2003
Christophe GRENIER <grenier@cgsecurity.org>
http://www.cgsecurity.org

Disk 80 - CHS 812 32 63 - 799 MB (Mxd BIOS mode)
Check current partition structure

get_MBR_data:
Partition sector doesn't have the endmark 0xAA55
Run "MBR code" in TestDisk to correct it

Quit

*=Primary bootable  P=Primary  L=Logical  E=Extended  D=Deleted
Quit this section_
```

**Fig.3: TestDisk Warning there's no "endmark" in MBR (not to scale).**

Then *most likely* your entire MBR sector was filled with zero bytes by some zap or wipe utility! TestDisk checks *only* the **64 bytes** that make up the Partition Table and the **last two bytes** of the **MBR** to see if they are the Hex Word "**AA55**" (i.e., the bytes 55 AA in that order on the disk). If the last two bytes have a problem, you'll see the display above. [Note: If you suspect that any of the MBR code has been damaged, you should first run FDISK /MBR from a Win98 Boot Disk *or* use TestDisk's "MBR Code" Option; since TestDisk *obviously* can't test for every type of MBR code in existence!]

Once you see the screen above, you must press the ENTER key a couple more times to get back to the Main display (hopefully that will be changed in a future version of TestDisk) where you can finally write a copy of the Standard MBR code:

```

TestDisk 4.4, Data Recovery Utility, March 26 2003
Christophe GRENIER <grenier@cgsecurity.org>
http://www.cgsecurity.org
Dos version

Disk 80 - CHS 812 32 63 - 799 MB (Mxd BIOS mode)

[Analyse ] [ Delete ] [ MBR Code ] [Geometry] [Options ] [Advanced]
[ Quit ]

Write the Classic MBR code to first sector_

```

**Fig.4: "MBR Code" (not to scale).**

Once you've highlighted "[ MBR Code ]" and pressed the ENTER key, TestDisk will respond with: "Write a new copy of MBR code to first sector? (Y/N)\_" ; press the Y key to do so, then highlight "[ Analyse ]" and press the ENTER key again.

The next display will show TestDisk counting through each of the cylinders as it further *analyzes* the drive:

```

Disk 80 - CHS 812 32 63 - 799 MB (Mxd BIOS mode)
Analyse cylinder      0/811_

Stop

```

**Fig.5: TestDisk's "Analyse Cylinders" (not to scale).**

*Be careful not* to press the ENTER key at this time, or you'll 'Stop' the analysis! Once the Analysis is complete, TestDisk will present a display similar to this:

```

TestDisk 4.4, Data Recovery Utility, March 26 2003
Christophe GRENIER <grenier@cgsecurity.org>
http://www.cgsecurity.org

Disk 80 - CHS 812 32 63 - 799 MB (Mxd BIOS mode)
Use arrow keys to change partition characteristics (see below)
Press the ENTER key to continue

   Partition                Start              End              Size
* FAT32                    0  1  1    811  31  63    1636929 [WIN98]

Structure: Ok  Press 'T' key to change partition type, 'P' to list files_
*=Primary bootable  P=Primary  L=Logical  E=Extended  D=Deleted

```

**Fig.6: TestDisk's "Current Structure" display (not to scale).**

For this simple case, TestDisk has already set the only partition it could find to be the "Primary bootable" partition. However, if your drive has many partitions (especially if you had *previously deleted* partitions -- let's call them *ghosts*; which overlap *or are overlapped by* newer partitions created after them), then it will be up to you to decide which of these are the most recent partitions that need to be placed into the Partition Table. Pressing the '**P**' key to **list** and verify the **files** that TestDisk finds in one of these partitions can help a great deal in deciding what to leave marked as **D**=Deleted or change to the \*=bootable Primary, **P**=Primary or **L**=Logical partition. (Use the Right and Left Arrow keys to make these changes.)

**[Note:** To recover and set the partitions correctly for a large drive with many different partitions may require you to learn a bit about how **Primary**, **Extended** and **Logical partitions** are used on such a drive!]

After pressing the '**P**' key, you can use the Right Arrow key when a directory <DIR> is 'highlighted' to see files inside that folder as well. Here's what the simple listing of some files in my test drive's DOS folder looked like:

```

TestDisk 4.4, Data Recovery Utility, March 26 2003
Christophe GRENIER <grenier@cgsecurity.org>
http://www.cgsecurity.org

* FAT32          0  1  1  811  31  63    1636929 [WIN98]
Use arrow keys to change directory, enter to quit

<DIR> 28-Oct-2002 18:13 .
<DIR> 28-Oct-2002 18:13 ..
 1063 23-Apr-1999 22:22 AUTOEXEC.BAT
 1005 23-Apr-1999 22:22 CONFIG.SYS
 1416 23-Apr-1999 22:22 SETRAMD.BAT
14764 23-Apr-1999 22:22 README.TXT
 6855 23-Apr-1999 22:22 FINDRAMD.EXE
12663 23-Apr-1999 22:22 RAMDRIVE.SYS
14386 23-Apr-1999 22:22 ASPI4DOS.SYS
21971 23-Apr-1999 22:22 BTCDDROM.SYS
29620 23-Apr-1999 22:22 ASPICD.SYS
30955 23-Apr-1999 22:22 BTDOSM.SYS
35330 23-Apr-1999 22:22 ASPI2DOS.SYS
37564 23-Apr-1999 22:22 ASPI8DOS.SYS

```

**Fig.7: TestDisk can list all the files in a partition! (not to scale).**

Pressing the Right Arrow key while a ".." <DIR> is selected, will take you back the parent folder. With the release of **TD** version **4.4**, you can now use the PageUp or PageDown key as well. The point *of being able to move through every file listing* is to make sure that this is the partition you want to recover. (Press the ENTER key to quit the 'File Listing' View.)

After pressing the ENTER key again, we arrive at the following display where you decide if you want to write the changes you made in memory to the hard drive's MBR (and EBR) sectors:

```

TestDisk 4.4, Data Recovery Utility, March 26 2003
Christophe GRENIER <grenier@cgsecurity.org>
http://www.cgsecurity.org

Disk 80 - CHS 812 32 63 - 799 MB (Mxd BIOS mode)

   Partition          Start          End          Size
 1 * FAT32            0  1  1      811  31 63    1636929 [WIN98]

Quit  Search!  Write

                               Quit this section_

```

**Fig.8: TestDisk's "Quit, Search! or Write" Screen (not to scale).**

**IF** you were not satisfied by TestDisk's results in the previous displays, you can always try using the "**Search!**" Option which will do a much more detailed search of the drive!

Once you have decided the partitions are correct, move the highlight over the "**Write**" selection like this:

```

Quit  Search!  Write

                               Write partition structure to disk_

```

**Fig.9: "Write partition structure to disk."**

and press the ENTER key, whereupon TestDisk will ask you: "**Confirm ? (Y/N)**" so press the **Y** key to complete the write to disk. Finally, you will be returned to the Main display where you can use the Left or Right Arrow keys to move the highlight over the "Quit program" selection.

I suggest that you try booting up your computer with the Win98 Boot Disk **again**, to check your partitions under **DOS** rather than trying to jump

right into Windows (just in case there's a big mistake in the partition size or something)!

## Appendix -- The FAT32 Boot Record under TestDisk:

Lastly, for those who'd like to check out the "[ **Advanced** ]" Option of TestDisk (for FAT partitions only), here are a few displays of a FAT32 Boot Record being examined under TestDisk:

```

Disk 80 - CHS 812 32 63 - 799 MB (Mxd BIOS mode)

  Partition      Start          End          Size
  1 * FAT32      0  1  1  811  31  63  1636929 [WIN98]

[ Quit ] [ Boot ]

                          Boot sector recovery_

```

**Fig.10: TestDisk's "Boot sector recovery" (not to scale).**

There are only a few functions that TestDisk can carry out on Boot Records... mostly *synchronizing* the bytes of the first copy to the second, or vice versa. But you can safely view every byte in both copies first:

```

Disk 80 - CHS 812 32 63 - 799 MB (Mxd BIOS mode)
 1 * FAT32          0  1  1  811  31 63      1636929 [WIN98]

Boot sector
OK
Backup boot sector
OK
Sectors are identical.

Quit  Rebuild BS  Dump

Quit this section_

```

**Fig.11: TestDisk's "Boot Sector functions" (not to scale).**

In this case the Boot and Backup sectors are identical; if they weren't, you'd be offered a choice of copying one to the other. Read the following NOTE! It gives you an idea why these functions are labeled ADVANCED in the Main menu!

**NOTE:** If any of your FAT32 Boot Records and their backup sectors are *not* identical, you should never 'sync' them until you know exactly **where** and **WHY** there's a difference! The "**Dump**" command (see pic below) shows the two records *side-by-side* so you can look for the differences. TestDisk already *knows* that the **2nd sectors** of FAT32 Boot Records have a valid reason for being different (and will *ignore* this particular type of difference): It's because the data which identifies how many free clusters are left in the volume (and where to look for them) is never updated in the backup sector! As a matter of fact, it's a good thing to begin a rebuilt Boot Record with the *default* value of "FF FF FF FF" hex (probably the reason they're never updated in the backups); see my page: [asm/mbr/MSWIN41.htm#FSINFO](http://asm/mbr/MSWIN41.htm#FSINFO) for more on the **FS Info Sector's** default cluster values. There may also be differences in the **1st** sector due to some utility trying to identify which is the 'backup' vs. the original Boot Record; again, don't make a change *unless you know why it is necessary!* The **3rd** sector should **never** have any differences, since this is always filled with the same code.

The "**Rebuild BS**" selection -- Rebuild Boot Sectors -- should be used **only** if you know that both copies are corrupted *and preferably* if you have a reference explaining what Hex values are supposed to be at which offsets (the tables on this page: [asm/mbr/MSWIN41.htm#BPB](http://asm/mbr/MSWIN41.htm#BPB) should be helpful in doing so). A disk editor is essential for making any necessary changes!

If you highlight the " **Dump** " selection and press ENTER, you'll see something similar to this:

```

Disk 80 - CHS 812 32 63 - 799 MB (Mxd BIOS mode)
 1 * FAT32          0 1 1 811 31 63      1636929 [WIN98]

Boot sector          Backup boot sector
0000 eb58904d 5357494e  .X.MSWIN  eb58904d 5357494e  .X.MSWIN
0008 342e3100 02082000  4.1...   342e3100 02082000  4.1...
0010 02000000 00f80000  .....   02000000 00f80000  .....
0018 3f002000 3f000000  ?. ?...  3f002000 3f000000  ?. ?...
0020 41fa1800 3d060000  A...=... 41fa1800 3d060000  A...=...
0028 00000000 02000000  .....   00000000 02000000  .....
0030 01000600 00000000  .....   01000600 00000000  .....
0038 00000000 00000000  .....   00000000 00000000  .....
0040 800029f7 18750d57  ..).u.W  800029f7 18750d57  ..).u.W
0048 494e3938 20202020  IN98     494e3938 20202020  IN98
0050 20204641 54333220  FAT32    20204641 54333220  FAT32
0058 2020fa33 c98ed1bc  .3....   2020fa33 c98ed1bc  .3....
0060 f87b8ec1 bd7800c5  .f...x.. f87b8ec1 bd7800c5  .f...x..
0068 76001e56 1655bf22  v..U.U." 76001e56 1655bf22  v..U.U."

Previous      Next      Quit
Quit dump section_
    
```

**Figs.12: TestDisk's "Boot sector Dump" Screens (not to scale).**

```

Boot sector                               Backup boot sector
0198 0d0a4469 736b2049  ..Disk I  0d0a4469 736b2049  ..Disk I
01A0 2f4f2065 72726f72  /O error  2f4f2065 72726f72  /O error
01A8 ff0d0a52 65706c61  ...Repla  ff0d0a52 65706c61  ...Repla
01B0 63652074 68652064  ce the d  63652074 68652064  ce the d
01B8 69736b2c 20616e64  isk, and  69736b2c 20616e64  isk, and
01C0 20746865 6e207072  then pr  20746865 6e207072  then pr
01C8 65737320 616e7920  ess any  65737320 616e7920  ess any
01D0 6b65790d 0a000000  key....  6b65790d 0a000000  key....
01D8 494f2020 20202020  IO      494f2020 20202020  IO
01E0 5359534d 53444f53  SYSMSDOS 5359534d 53444f53  SYSMSDOS
01E8 20202053 59537e01  SYS~.   20202053 59537e01  SYS~.
01F0 0057494e 424f4f54  .WINBOOT 0057494e 424f4f54  .WINBOOT
01F8 20535953 000055aa  SYS..U. 20535953 000055aa  SYS..U.
0200 52526141 00000000  RRaA... 52526141 00000000  RRaA...

Previous  Next  Quit

```

**NOTE:** In TestDisk v. 4.4, Christophe has added the use of the **PageUp** and **PageDown** keys rather than just the Arrow keys, **so scrolling through all three sectors** is much easier now!

Right Arrow over to the "Quit" and press ENTER, then either save your changes or just Quit the program like normal.

*Last Update: May 4, 2005. (04.05.2005)*

You can write to me using this: [online reply form](#). (It opens in a new window.)

 [The Starman's Realm Index Page](#)