

Testdisk Usage And Installation

Taken From the TestDisk Documentation

(Christophe Grenier, Simone Brandt, and Daniel B. Sed



Introduction

TestDisk is a powerful data recovery utility and a tool to check and undelete partitions. It was primarily designed to help recover lost partitions and/or make non-booting disks bootable again when these symptoms are caused by faulty software, certain types of viruses or human error (such as accidentally erasing your Partition Table). TestDisk can run under DOS (either real or in a Windows 9x DOS-box), Windows 32-bit (NT4, 2000, XP, 2003), Linux OSs, FreeBSD, NetBSD, OpenBSD, SunOS and MacOS; precompiled binary executables are available for DOS, Win32 and Linux. But TestDisk will find lost partitions for all of these file systems:

- BeFS (BeOS)
- BSD disklabel (FreeBSD/OpenBSD/NetBSD)
- CramFS, Compressed File System
- DOS/Windows FAT12, FAT16 and FAT32
- HFS and HFS+, Hierarchical File System
- JFS, IBM's Journaled File System
- Linux Ext2 and Ext3
- Linux Raid
 - RAID 1: mirroring
 - RAID 4: striped array with parity device
 - RAID 5: striped array with distributed parity information
 - RAID 6: striped array with distributed dual redundancy information
- Linux Swap (versions 1 and 2)
- LVM and LVM2, Linux Logical Volume Manager
- Mac partition map
- Netware NSS
- NTFS (Windows NT/2K/XP/2003)
- ReiserFS 3.5 and 3.6
- Sun Solaris i386 disklabel
- UFS and UFS2 (Sun/BSD/...)
- XFS, SGI's Journaled File System

TestDisk queries the BIOS (DOS/Win9x) or the OS (Linux, FreeBSD, Windows) in order to find the Hard Disks and their characteristics (LBA size and CHS geometry). TestDisk does a quick check of your disk's structure and compares it with your Partition Table for entry errors. If the Partition Table has entry errors, TestDisk can repair them. If you have missing partitions or a completely empty Partition Table, TestDisk can search for partitions and create a new Table or even a new MBR if necessary. However, it's up to the user to look over the list of possible partitions found by TestDisk and to select the one(s) which were being used just before the drive failed to boot or the partition(s)

Testdisk Usage And Installation

Taken From the TestDisk Documentation

(Christophe Grenier, Simone Brandt, and Daniel B. Sed



were lost. In some cases, especially after initiating a detailed search for lost partitions, TestDisk may show partition data which is simply from the remnants of a partition that had been deleted and overwritten long ago.

TestDisk has features for both novices and experts. For those who know little or nothing about data recovery techniques, the command line parameters /log and /debug can be used to collect detailed information about a non-booting drive which can then be sent to a tech for further analysis. Those more familiar with such procedures should find TestDisk a handy tool in performing onsite recovery.

To recover lost pictures or files from digital camera or hard disk, run the PhotoRec command.

TestDisk OS

TestDisk can run under:

- DOS (either real or in a Windows 9x DOS-box),
- Windows 32-bit (NT4, 2000, XP, 2003),
- Linux OSs,
- FreeBSD,
- NetBSD,
- OpenBSD,
- SunOS and
- MacOS

On TestDisk web site, precompiled binary executables are available for DOS, Win32 and Linux.

TestDisk Installation

These are generic installation instructions.

The “configure” shell script attempts to guess correct values for various system-dependent variables used during compilation. It uses those values to create a “Makefile” in each directory of the package. It may also create one or more “.h” files containing system-dependent definitions. Finally, it creates a shell script “config.status” that you can run in the future to recreate the current configuration, and a file “config.log” containing compiler output (useful mainly for debugging “configure”).

It can also use an optional file, typically called “config.cache”, and enabled with “--cache-file=config.cache”, or simply “-C”, that saves the results of its tests to speed up reconfiguring. Caching is disabled by default to prevent problems with accidental use of stale cache files.

If you need to do unusual things to compile the package, please try to figure out how “configure” could check whether to do them, and mail diffs or instructions to the address given in the “README” so they

Testdisk Usage And Installation

Taken From the TestDisk Documentation

(Christophe Grenier, Simone Brandt, and Daniel B. Sed



can be considered for the next release. If you are using the cache, and at some point “config.cache” contains results you don't want to keep, you may remove or edit it.

The file “configure.ac”, or “configure.in”, is used to create “configure” by a program called “autoconf”. You only need “configure.ac” if you want to change it or regenerate “configure” using a newer version of “autoconf”.

The simplest way to compile this package is:

1. “cd” to the directory containing the package's source code and type “./configure” to configure the package for your system. If you're using “csh” on an old version of System V, you might need to type “sh ./configure” instead to prevent “csh” from trying to execute “configure” itself. Running “configure” takes awhile. While running, it prints some messages telling which features it is checking for.
2. Type “make” to compile the package.
3. Optionally, type “make check” to run any self-tests that come with the package.
4. Type “make install” to install the programs and any data files and documentation.
5. You can remove the program binaries and object files from the source code directory by typing “make clean”. To also remove the files that “configure” created (so you can compile the package for a different kind of computer), type “make distclean”. There is also a “make maintainer-clean” target, but that is intended mainly for the package's developers. If you use it, you may have to get all sorts of other programs in order to regenerate files that came with the distribution.

Some systems require unusual options for compilation or linking that the “configure” script does not know about. Run “./configure --help” for details on some of the pertinent environment variables.

You can give “configure” initial values for configuration parameters by setting variables in the command line or in the environment. Here is an example: **./configure CC=c89 CFLAGS=-O2 LIBS=-lposix**

You can compile the package for more than one kind of computer at the same time, by placing the object files for each architecture in their own directory. To do this, you must use a version of “make” that supports the “VPATH” variable, such as GNU “make”. “cd” to the directory where you want the object files and executables to go and run the “configure” script. “configure” automatically checks for the source code in the directory that “configure” is in and in “..”.

If you have to use a “make” that does not support the “VPATH” variable, you have to compile the package for one architecture at a time in the source code directory. After you have installed the package for one architecture, use “make distclean” before reconfiguring for another architecture.

Testdisk Usage And Installation

Taken From the TestDisk Documentation

(Christophe Grenier, Simone Brandt, and Daniel B. Sed



By default, “make install” will install the package's files in “/usr/local/bin”, “/usr/local/man”, etc. You can specify an installation prefix other than “/usr/local” by giving “configure” the option “--prefix=PREFIX”. You can specify separate installation prefixes for architecture-specific files and architecture-independent files. If you give “configure” the option “--exec-prefix=PREFIX”, the package will use PREFIX as the prefix for installing programs and libraries. Documentation and other data files will still use the regular prefix.

In addition, if you use an unusual directory layout you can give options like “--bindir=DIR” to specify different values for particular kinds of files. Run “configure --help” for a list of the directories you can set and what kinds of files go in them.

If the package supports it, you can cause programs to be installed with an extra prefix or suffix on their names by giving “configure” the option “--program-prefix=PREFIX” or “--program-suffix=SUFFIX”.

Some packages pay attention to “--enable-FEATURE” options to “configure”, where FEATURE indicates an optional part of the package. They may also pay attention to “--with-PACKAGE” options, where PACKAGE is something like “gnu-as” or “x” (for the X Window System). The “README” should mention any “--enable-” and “--with-” options that the package recognizes.

For packages that use the X Window System, “configure” can usually find the X include and library files automatically, but if it doesn't, you can use the “configure” options “--x-includes=DIR” and “--x-libraries=DIR” to specify their locations.

There may be some features “configure” cannot figure out automatically, but needs to determine by the type of machine the package will run on. Usually, assuming the package is built to be run on the `_same_` architectures, “configure” can figure that out, but if it prints a message saying it cannot guess the machine type, give it the “--build=TYPE” option. TYPE can either be a short name for the system type, such as “sun4”, or a canonical name which has the form: **CPU-COMPANY-SYSTEM**

where SYSTEM can have one of these forms: **OS KERNEL-OS**

See the file “config.sub” for the possible values of each field. If “config.sub” isn't included in this package, then this package doesn't need to know the machine type.

If you are `_building_` compiler tools for cross-compiling, you should use the “--target=TYPE” option to select the type of system they will produce code for.

If you want to `_use_` a cross compiler, that generates code for a platform different from the build platform, you should specify the “host” platform (i.e., that on which the generated programs will eventually be run) with “--host=TYPE”.

Testdisk Usage And Installation

Taken From the TestDisk Documentation

(Christophe Grenier, Simone Brandt, and Daniel B. Sed



If you want to set default values for “configure” scripts to share, you can create a site shell script called “config.site” that gives default values for variables like “CC”, “cache_file”, and “prefix”. “configure” looks for “PREFIX/share/config.site” if it exists, then “PREFIX/etc/config.site” if it exists. Or, you can set the “CONFIG_SITE” environment variable to the location of the site script. A warning: not all “configure” scripts look for a site script.

Variables not defined in a site shell script can be set in the environment passed to “configure”. However, some packages may run configure again during the build, and the customized values of these variables may be lost. In order to avoid this problem, you should set them in the “configure” command line, using “VAR=value”. For example:

```
./configure CC=/usr/local2/bin/gcc
```

will cause the specified gcc to be used as the C compiler, unless it is overridden in the site shell script.

TestDisk & PhotoRec Source Compilation

TestDisk and PhotoRec use an ncurses interface, so ncurses library and development files must be present. To be able to list files from Ext2/ext3, Reiser and NTFS partitions, some additional libraries can be used:

- Ncurses Library (ncurses & ncurses-devel)
- Ext2fs library (e2fsprogs & e2fsprogs-devel)
- Reiserfs library (progsreiserfs)
- NTFS library (e2fsprogs & e2fsprogs-devel)
- For a better JPEG recovery rate, PhotoRec must be compiled with the libjpeg library.

To compile TestDisk and PhotoRec run:

```
./configure PREFIX=/usr/local/testdisk --exec-prefix=/usr/local/testdisk/bin --  
cache-file=config.cache
```

```
make or make static
```

```
make check
```

```
make install
```

You may need to specify parameters to configure, see **configure --help**. Example ./configure:

```
--with-reiserfs-lib=/home/kmaster/perso/testdisk-5.2/progsreiserfs-0.3.1-rc8/libreiserfs/.libs/  
--with-reiserfs-includes=/home/kmaster/perso/testdisk-5.2/progsreiserfs-0.3.1-rc8/include/  
--with-ntfs-lib=/home/kmaster/perso/testdisk-5.2/ntfsprogs-1.8.5/libntfs/.libs/
```

Testdisk Usage And Installation

Taken From the TestDisk Documentation

(Christophe Grenier, Simone Brandt, and Daniel B. Sed



```
--with-ntfs-includes=/home/kmaster/perso/testdisk-5.2/ntfsprogs-1.8.5/include/  
--with-ext2fs-lib=/home/kmaster/perso/testdisk-5.2/e2fsprogs-1.34/lib  
--with-ext2fs-includes=/home/kmaster/perso/testdisk-5.2/e2fsprogs-1.34/lib
```

Static Version

If you need to run the binary on systems that lack one of the previous libraries, create a static binary with **make static**. If you need a small binary for binary distribution, get the UPX packer at <http://upx.sourceforge.net/http://upx.sourceforge.net/> and run

```
make small  
upx src/testdisk src/photorec
```

Note that Linux UPX need decompress program file before run, what request some space on /tmp and running executable must be allowed in this directory (mount -o remount,exec /tmp).

Working with Damaged Hard Disks

You can also use TestDisk to help analyze the sectors copied from a hard drive with physical problems onto a good drive. If there are any bad sectors on your hard disk, you should first copy its data to another hard disk before attempting to recover its data. The new disk must be at least exactly the same size (check the number of LBA sectors) or larger; when larger, it's usually not a problem because the number of heads per cylinder and sectors per head will be the same if both disks use LBA mode.

Under Linux:

- Primary Master **IDE** disk device is **/dev/had**
- Primary Slave IDE device is **/dev/hdb**
- Secondary Master IDE device is **/dev/hdc** and so on.
- **SATA** HDD device filenames usually begin at **/dev/hde** or **/dev/sda**
- **SCSI** device filenames always begin at **/dev/sda**
- **USB** device are often using an SCSI device **/dev/sda**

To list the partitions of a disk, log in as root and run **fdisk -l device**. Once you have verified the device names for your damaged disk and the new one, in a command shell (CLI) or terminal, **not** from within any OS on the damaged disk, run:

```
dd if=/dev/old_disk of=/dev/new_disk conv=noerror,sync
```

or to create an image file:

```
dd if=/dev/old_disk of=image_file conv=noerror
```

Testdisk Usage And Installation

Taken From the TestDisk Documentation

(Christophe Grenier, Simone Brandt, and Daniel B. Sed



to copy the data. To speed up the copy process, you can append **bs=8k**, it will read/write the disk by 16 sectors at a time. If you believe there are many damaged sectors on the drive, you should try using either Kurt Garloff's "dd_rescue" or Antonio Diaz's "ddrescue".

Warning: If you use TestDisk to recover lost partitions on a target hard drive by connecting the drive to a computer other than the one it was originally partitioned and formatted on, you risk the chance of having **incorrect data** about the drive passed along to TestDisk from the BIOS of this other computer. BIOS chips have at least two different ways of translating disk geometry, and using the wrong data may make it impossible for TestDisk to correctly recover a drive's lost partitions; or worse, you may write the wrong data to the drive's MBR, boot up the disk and then incorrectly write data to the disk leading to further corruption and loss of data. (This may not apply to all file systems.) TestDisk tries to detect this problem.

TestDisk can also work with hard disk images. The disk image must be available under its image_filename in the working directory, or you can add its Path and filename to a TestDisk command line. For example:

```
testdisk_win.exe C:\BOCHS\DOS\c.img
```

will include the Path\image_file, c.img, in the list of drives on TestDisk's initial screen. Then open the image file just like any physical drive. Image files are limited in size by the OS and file system:

- 2 GiB (FAT16)
- 4 GiB (FAT32)
- 16 GiB (EXT2/3 with 1kb block)
- 256 GiB (EXT2/3 with 2kb block)
- 2048 GiB (EXT2/EXT3 with 4kb block)

TestDisk Startup

When TestDisk is executed, you may see the phrase "Please wait..." on your screen until it has gathered enough data from the BIOS or OS to list the disk drives on the system.

```
TestDisk 6.2-WIP, Data Recovery Utility, November 2005  
Christophe GRENIER <grenier@cgsecurity.org>  
http://www.cgsecurity.org
```

TestDisk is free software, and comes with ABSOLUTELY NO WARRANTY.

```
Select a media (use Arrow keys, then press ENTER):  
Disk /dev/hdc - 708 MB / 675 MiB  
Disk /dev/sda - 120 GB / 111 GiB
```

Testdisk Usage And Installation

Taken From the TestDisk Documentation

(Christophe Grenier, Simone Brandt, and Daniel B. Sed



```
Disk /dev/sdb - 120 GB / 111 GiB
```

```
[Proceed ] [ Quit ]
```

Note: Disk capacity must be correctly detected for a successful recovery. If a disk listed above has incorrect size, check HD jumper settings, BIOS detection, and install the latest OS patches and disk drivers.

If the reported size doesn't match the hard disk size, i.e., a 120 GB hard disk is recognized as only a 32 GB hard disk, check your BIOS hard disk settings and the jumpers on the disk. On most large hard disks, there are jumpers to limit the size to only 32 or 8 GB. If your HD is detected as 130 GB only, LBA48 support may not be available in your OS, read TestDisk and OS choice for more information.

Next step is to select the partition table type:

```
TestDisk 6.2-WIP, Data Recovery Utility, November 2005
Christophe GRENIER <grenier@cgsecurity.org>
http://www.cgsecurity.org
```

```
Disk /dev/sda - 120 GB / 111 GiB
```

Please select the partition table type, press ENTER when done.

```
[ Intel ] Intel/PC partition
[ Mac   ] Apple partition map
[ None  ] Non partitioned media
[ Sun   ] Sun Solaris partition
[ Xbox  ] Xbox partition
```

Note: Do NOT select 'None' for media with only a single partition. It's very rare for a drive to be 'Non-partitioned'.

TestDisk Menu Items

```
TestDisk 6.2-WIP, Data Recovery Utility, November 2005
Christophe GRENIER <grenier@cgsecurity.org>
http://www.cgsecurity.org
```

```
Disk /dev/sda - 120 GB / 111 GiB - CHS 14593 255 63
```

```
[ Analyse ] Analyse current partition structure and search for lost
partitions
[ Advanced ] Filesystem Utils
[ Geometry ] Change disk geometry
[ Options  ] Modify options
```

Testdisk Usage And Installation

Taken From the TestDisk Documentation

(Christophe Grenier, Simone Brandt, and Daniel B. Sed



```
[ MBR Code ] Write TestDisk MBR code to first sector
[ Delete   ] Delete all data in the partition table
[ Quit     ] Return to disk selection
```

Note: Correct disk geometry is required for a successful recovery. 'Analyse' process may give some warnings if it thinks the logical geometry is mismatched.

If you don't understand how to use TestDisk:

- run "**testdisk_win**" or "**testdisk /log /debug**" (make sure to hit the space bar once before each forward slash),
- select the faulty hard disk using arrow keys then press the ENTER key,
- after TestDisk is finished (Note: you may need to press the ENTER key a couple more times during its processing), choose **Search!** to restart the analysis.
- just send the file which TestDisk creates, **testdisk.log**, to <mailto:grenier@cgsecurity.org> and a brief explanation about the problem and your previous partitions (size, label, file system type).

Note: TestDisk **appends** new information to **testdisk.log**; it does not overwrite an existing file.

After writing a new partition structure, you have to reboot for the change to take effect. Check the file systems and repair them if necessary.:

- To undelete files on Linux, you can try <http://e2undel.sourceforge.net/e2undel>.
- If the superblock of a ReiserFS partition is missing, it can be rebuild with **reiserfsck --rebuild-sb device**.
- To check a NTFS partition from Windows, run **chkdsk c:**.
- For Linux or FreeBSD, you may have to update your **/etc/fstab** to reflect the new partition order.
- In DOS or Win9x, if your OS doesn't boot, you can reinstall the system files with "sys c:".
- If your Windows 2000/XP/2003 machine doesn't boot, try **fixmbr** from the Windows Recovery Console and **fixboot** to repair NTFS boot sector.

Testdisk Usage And Installation

Taken From the TestDisk Documentation

(Christophe Grenier, Simone Brandt, and Daniel B. Sed



- You may have to update your multiboot configuration and reinstall the multiboot in the Master Boot Record.
 - Lilo: `/etc/lilo.conf`, lilo to re-install
 - Grub: `/boot/grub/grub.conf`, grub-install device
 - Windows (NT/2000/XP/2003/...): `c:\boot.ini`, run fixmbr from the Recovery Console (i.e. fixmbr \Device\HardDisk0)

Examples

Here are some complex examples of data recovery with TestDisk:

Recovery of a Dell computer

On Dell computer, there is a special partition called DellUtility. It's a FAT16 partition that is not visible from Windows because its partition type is DE.

```
Disk 80 - CHS 4865 255 63 - 38162 MB (Enh BIOS mode)
* FAT16 >32M          0  1  1      3 254 63      64197 [DellUtility]
P HPFS - NTFS         4  0  1  4864 254 63      78091965
```

Select the DellUtility partition, use 'T' to change the partition type to DE.

Use the arrow key to boot on the NTFS partition.

```
Disk 80 - CHS 4865 255 63 - 38162 MB (Enh BIOS mode)
P Dell Utility       0  1  1      3 254 63      64197 [DellUtility]
* HPFS - NTFS       4  0  1  4864 254 63      78091965
```

When all Partitions are Deleted

In this case, all partitions have been deleted. TestDisk show no partition! The user has run testdisk /debug /log. Extract of testdisk.log

```
Analyse Disk /dev/hdb - CHS 5169 240 63 - 38161 MB
No partition is bootable
search_part()
Disk /dev/hdb - CHS 5169 240 63 - 38161 MB
FAT32 at 0/1/1
heads/cylinder 255 (FAT) != 240 (HD)
```

A FAT32 partition has been found but it has been rejected because of a bad hard disk geometry. The BIOS has selected another hard disk geometry because of the content of the MBR (empty in this case). Under TestDisk, choose geometry, set the number of heads to 255 instead of 240 and run again Analyse. TestDisk still displays no partition!

Testdisk Usage And Installation

Taken From the TestDisk Documentation

(Christophe Grenier, Simone Brandt, and Daniel B. Sed



In the log, the FAT partition is visible.

```
Disk /dev/hdb - CHS 4864 255 63 - 38154 MB
FAT32 at 0/1/1
D empty                0  1  1      0  0 63          0
FAT1 : 32-9576
FAT2 : 9577-19121
start_rootdir : 19122 root cluster : 2
Data : 19122-78156145
sectors : 78156162
cluster_size : 64
no_of_cluster : 1220891 (2 - 1220892) => FAT32
fat_length 9545 calculated 9539
D FAT32 LBA            0  1  1  4864 254 63    78156162
```

This is another problem, the partition ends at cylinder 4864, but the hard disk has only 4864 cylinders (first cylinder is 0). In Options, choose "Allow partition last cylinder" and after set the geometry. This time, the partition is visible and can be recovered.

Recovery of a Damaged FAT Boot Sector

```
Analyse Disk 80 - CHS 3737 255 63 - 29313 MB (Enh BIOS mode)
1 * FAT32                0  1  1   382 254 63    6152832 [LOKAL DISK]
2 E extended LBA        383  0  1  3736 254 63    53882010
Partition sector doesn't have the endmark 0xAA55
5 L FAT32               383  1  1  3736 254 63    53881947
5 L FAT32               383  1  1  3736 254 63    53881947
```

The boot sector of the logical FAT32 partition is damaged. (Windows error message is usually "The type of the file system is RAW." or "The disk in drive D is not formatted. Do you want to format it now ?") In Advanced, select this partition:

```
Interface Advanced
1 * FAT32                0  1  1   382 254 63    6152832 [LOKAL DISK]
2 E extended LBA        383  0  1  3736 254 63    53882010
5 L FAT32               383  1  1  3736 254 63    53881947
Boot sector
test_FAT :
Partition sector doesn't have the endmark 0xAA55
Backup boot sector
OK
```

First sectors (Boot code and partition information) are not identical.
Second sectors (cluster information) are not identical.

Testdisk Usage And Installation

Taken From the TestDisk Documentation

(Christophe Grenier, Simone Brandt, and Daniel B. Sed



Third sectors (Second part of boot code) are not identical.

The backup boot sector is valid, choose "Backup BS" to copy backup boot sector over boot sector.

Recovery of a lost and Damaged NTFS

Analyse Disk 81 - CHS 2434 255 63 - 19092 MB (Enh BIOS mode)

No partition is bootable

No partition is available.

Analyse Disk 81 - CHS 2434 255 63 - 19092 MB (Enh BIOS mode)

L FAT32 1275 1 1 2433 254 63 18619272 [NO NAME]

Only the second NTFS partition is found. Select Search! to try to find more partitions.

Analyse Disk 81 - CHS 2434 255 63 - 19092 MB (Enh BIOS mode)

* HPFS - NTFS 0 1 1 1274 254 63 20482812

L FAT32 1275 1 1 2433 254 63 18619272 [NO NAME]

Both partitions have been found, but the first NTFS has been found using backup boot sector, we need to restore the boot sector. Choose Write and next choose "Backup BS" to copy backup boot sector over boot sector.

Two FAT32 Partitions To Recover

There were two FAT32 on the hard disk but they have been deleted. After running Analyze and Search!, TestDisk has found only the second one

Disk 81 - CHS 525 255 63 - 4118 MB

L FAT32 384 1 1 524 254 63 2265102

Using A to add what we think is the missing partition, a new partition table have been written with two FAT32:

1 P FAT32 0 1 1 383 254 63 6168897

2 E extended 384 0 1 524 254 63 2265165

5 L FAT32 384 1 1 524 254 63 2265102

Using Advanced, Boot, RebuildBS, we have try to rebuild the boot sector of the first FAT32. Using List, it is possible to see a listing of files from the root directory but there is also a lot of garbage... Nothing has been written. In the log file, we can see that 4 copies of FAT32 have been found:

FAT32 at 32(0/1/33), nbr=123

FAT32 at 8221(0/131/32), nbr=123

FAT32 at 16097(1/1/33), nbr=1234

FAT32 at 22100(1/96/51), nbr=1234

Testdisk Usage And Installation

Taken From the TestDisk Documentation

(Christophe Grenier, Simone Brandt, and Daniel B. Sed



Normally only two copies of FAT should be found. There are remaining data from two different FAT32 partitions: one beginning at 0/1/1, the second at 1/1/1. We have done a mistake. This time, we add the correct partition and choose to write

```
1 E extended          1 0 1 524 254 63 8418060
5 L FAT32             1 1 1 383 254 63 6152832
6 L FAT32             384 1 1 524 254 63 2265102
```

With RebuildBS (Advanced/Boot), we have been able to successfully rebuild the boot sector.

Lost Partition After Defrag

After running defrag against the first partition, the second FAT32 drive vanishes. When checking the partition table, TestDisk detects a problem with the first partition and no logical partition.

Disk 81 - CHS 1245 255 63 - 9766 MB

check_FAT: Incorrect size of partition

```
1 * FAT32 LBA          0 1 1 746 254 63 12000492
1 * FAT32 LBA          0 1 1 746 254 63 12000492
2 E extended          747 0 1 1244 254 63 8000370
```

TestDisk has been able to find the first partition but this partition is one sector bigger than the actual partition entry. Defrag has overwrite the beginning of the extended partition, clearing the logical partition entry.

```
* FAT32                0 1 1 747 254 63 12016557 [D DRIVE]
D Linux                1023 1 1 1244 254 63 3566367
```

The user has choose to only recover the FAT32 partition. After a backup of its data, he has shrink the FAT32 file system to use one sector less. It's now time to recover the second partition. The user has manually add the second partition entry.

```
1 * FAT32                0 1 1 746 254 63 12000492 [D DRIVE]
2 E extended LBA       747 0 1 1244 254 63 8000370
5 L FAT32              747 1 1 1244 254 63 8000307
```

In Advanced, select the second FAT32 and RebuildBS. After a reboot and a little file system check, data were again available.

Testdisk Usage And Installation

Taken From the TestDisk Documentation

(Christophe Grenier, Simone Brandt, and Daniel B. Sed



An Example With Pictures

```
TestDisk 3.9, Data Recovery Utility, September 30 2001
Christophe GRENIER <grenier@nef.esiea.fr>
http://www.esiea.fr/public_html/Christophe.GRENIER/
Linux version

Disk /dev/hda - CHS 2501 255 63 - 19618 MB
Disk /dev/hdb - CHS 1232 255 63 - 9664 MB

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

Analyze current partition structure and recover lost partitions
```

```
TestDisk 3.9, Data Recovery Utility, September 30 2001
Christophe GRENIER <grenier@nef.esiea.fr>
http://www.esiea.fr/public_html/Christophe.GRENIER/

Disk /dev/hda - CHS 2495 255 63 - 19571 MB
Use arrows to change partition type
Partition      Start      End      Size
D HPFS - NTFS  0 1 1 248 254 63 4000122 [MYNTFS]
D HPFS - NTFS  0 1 1 259 254 63 4176837 [MYNTFS]
L FAT16 >32M   260 1 1 520 254 63 4192902 [WIN98]
D Linux ext2fs 521 0 1 526 254 63 96390 [MP3]
D FreeBSD     521 0 1 1020 254 63 8032500
D FAT16 >32M   785 0 1 1020 254 63 3791340
L Linux ext2fs 1021 1 1 1022 254 63 32067 [BOOT]
L Linux ext2fs 1023 1 1 1048 254 63 417627 [RACINE]
L Linux Swap  1049 1 1 1080 254 63 514017
L Linux ext2fs 1081 1 1 1233 254 63 2457882 [ARCH2]
L Linux ext2fs 1234 1 1 1259 254 63 417627 [TMP]
L Linux ext2fs 1260 1 1 1272 254 63 208782 [ROOT]
L Linux ext2fs 1273 1 1 1403 254 63 2104452 [HOME]
L Linux ext2fs 1404 1 1 1920 254 63 8305542 [arch1]
L Linux ext2fs 1921 1 1 2182 254 63 4208967 [USR]
Structure: 0k Use 'T' to toggle partition type
*=Primary bootable P=Primary L=Logical E=Extended D=Deleted
NTFS found using backup sector!
```

Testdisk Usage And Installation

Taken From the TestDisk Documentation

(Christophe Grenier, Simone Brandt, and Daniel B. Sed



```
TestDisk 3.9, Data Recovery Utility, September 30 2001
Christophe GRENIER <grenier@nef.esiea.fr>
http://www.esiea.fr/public_html/Christophe.GRENIER/

Disk /dev/hda - CHS 2495 255 63 - 19571 MB
Check current partition structure
  Partition      Start      End      Size
  1 P hid. HPFS/NTFS      0  1  1  259 254 63  4176837 [MYNTFS]
  2 P hid. FAT16 >32M    260  1  1  520 254 63  4192902 [WIN98]
  3 P FreeBSD          521  0  1  1020 254 63  8032500
  4 E extended LBA     1021  0  1  2494 254 63  23679810
There are hidden partitions, but no OS2 MultiBoot.
No partition is bootable
  5 L Linux ext2fs     1021  1  1  1022 254 63   32067 [B00T]
  X extended         1023  0  1  1048 254 63   417690
  6 L Linux ext2fs     1023  1  1  1048 254 63  417627 [RACINE]
  X extended         1049  0  1  1080 254 63   514080
  7 L Linux Swap       1049  1  1  1080 254 63   514017
  X extended         1081  0  1  1233 254 63  2457945
  8 L Linux ext2fs     1081  1  1  1233 254 63  2457882 [ARCH2]
  X extended         1234  0  1  1259 254 63   417690
Previous  Next  Quit 

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

```
TestDisk 3.9, Data Recovery Utility, September 30 2001
Christophe GRENIER <grenier@nef.esiea.fr>
http://www.esiea.fr/public_html/Christophe.GRENIER/

Disk /dev/hda - CHS 2495 255 63 - 19571 MB
Analyse cylinder 555/2494

  D HPFS - NTFS      0  1  1  259 254 63  4176837 [MYNTFS]
  D HPFS - NTFS      0  1  1  248 254 63  4000122 [MYNTFS]
  D HPFS - NTFS      0  1  1  259 254 63  4176837 [MYNTFS]
  D FAT16 >32M      260  1  1  520 254 63  4192902 [WIN98]
  D Linux ext2fs     521  0  1  526 25 20   81920 [MP3]
  D Linux ext2fs     521  0  1  526 25 20   81920 [MP3]
  D FreeBSD         521  0  1  1020 254 62  8032499

 Stop
```

Testdisk Usage And Installation

Taken From the TestDisk Documentation

(Christophe Grenier, Simone Brandt, and Daniel B. Sed



```
TestDisk 3.9, Data Recovery Utility, September 30 2001
Christophe GRENIER <grenier@nef.esiea.fr>
http://www.esiea.fr/public_html/Christophe.GRENIER/

Disk /dev/hda - CHS 2495 255 63 - 19571 MB

    Partition          Start          End          Size
  1 * HPFS - NTFS      0             1 1 248 254 63 4000122 [MYNTFS]
  2 P FAT16 >32M      260           1 1 520 254 63 4192902 [WIN98]
  3 P FreeBSD         521           0 1 1020 254 63 8032500
  4 E extended LBA    1021           0 1 2494 254 63 23679810
  5 L Linux ext2fs     1021           1 1 1022 254 63 32067 [BOOT]
  6 L Linux ext2fs     1023           1 1 1048 254 63 417627 [RACINE]
  7 L Linux Swap       1049           1 1 1080 254 63 514017
  8 L Linux ext2fs     1081           1 1 1233 254 63 2457882 [ARCH2]
  9 L Linux ext2fs     1234           1 1 1259 254 63 417627 [TMP]
 10 L Linux ext2fs     1260           1 1 1272 254 63 208782 [ROOT]
 11 L Linux ext2fs     1273           1 1 1403 254 63 2104452 [HOME]
 12 L Linux ext2fs     1404           1 1 1920 254 63 8305542 [arch1]
 13 L Linux ext2fs     1921           1 1 2182 254 63 4208967 [USR]
 14 L FAT32           2183           1 1 2494 254 63 5012217 [WIN98 DJ]
Quit Search! Write

Write partition structure to disk
```

Recovery of CD-ROM Session

With multi-session CD-ROM, it is possible to delete files of previous session. Because the files are not really deleted, it is possible to recover them. To read files from the first session, run under Linux `mount /dev/cdrom /mnt/cdrom -t iso9660 -o session=0`

Remnants Of Partitions

TestDisk may show partition data, which is simply from the *remnants* of a partition that had been deleted and overwritten long ago. Because of this possibility, one should always give each new partition a unique label name. Whenever you decide to *delete all existing partitions* on a drive in order to *'start all over again,'* you should 'zero-out' every byte on the disk first. That procedure will remove all remnants of previously existing partitions (HDD manufacturers often have free utility programs to 'zero-out' a drive in the downloads section of their web sites). If one isn't available, or if you're **not** removing *all* of the partitions from a drive, you could at the very least 'defrag' the drive, or use a 'wipe free space' utility *after* you installed a new OS or *resized* any of the drive's partitions.

As we mentioned above, TestDisk could easily find *remnants* of partitions you purposely deleted long ago. In order to make the recovery of lost partitions (that you do want to recover!) less confusing, each time you alter the structure of your hard drive (by deleting, creating or resizing a partition) you should run a utility program that shows you exactly where all of your existing partitions are located then save that data in a safe place (not on the HDD, since you might not be able to access it when needed!). There are three versions of *Partition Info* by PQ (now Symantec). PartInNT.zip is for NT/2k/XP etc., PartIn9x.zip for Win 9x/Me and partinfo.zip for DOS; which could be useful. For Linux

Testdisk Usage And Installation

Taken From the TestDisk Documentation

(Christophe Grenier, Simone Brandt, and Daniel B. Sed



users, it's quite easy to run **fdisk -l** and see a listing of all your existing partitions. You can also make a copy of your MBR sector, or the whole first track of any HDD, using the 'dd' program like this:

```
dd if=/dev/hda of=HDA1stTrack.bin bs=512 count=63
```

which will save the first 63 sectors of your first IDE drive to the file *HDA1stTrack.bin*. To save only the MBR sector, use a count of just 1. Copy the files to a couple floppy diskettes, don't forget to print out a listing of your partitions for safekeeping.

Also note that TestDisk itself can be run at any time (just be sure to **Quit** rather than save its MBR to disk!) and set to save a log file (/log switch) of all the partitions it finds. Once you identify your existing partitions in its log file, this info could be very useful if you ever need to recover a partition in the future.

SMART

It is possible to control and monitor storage systems using the Self-Monitoring, Analysis and Reporting Technology System (SMART) built into most modern ATA and SCSI hard disks. Linux users can use <http://smartmontools.sourceforge.net/Smartmontools>

Norton GoBack && TestDisk

Norton <http://en.wikipedia.org/wiki/GoBackGoBack> (previously known as Wildfile GoBack, Adaptec GoBack, Roxio GoBack) is a Microsoft Windows based disk utility. If you run Windows version of TestDisk, you need to disable or uninstall GoBack first to be able to write a new partition table.

Microsoft Fdisk

When run, Microsoft FDISK destroys some data on your drive. While "checking disk integrity", FDISK writes hexadecimal F6h into the first and 7th sectors on each track. If you have accidentally deleted a volume, do not run any Microsoft FDISK program!

How Do DOS Versions Of Testdisk Get Disk Geometry ?

The DOS version of TestDisk uses three BIOS calls to get a disk's geometry. It gets the number of cylinders, heads and sectors (CHS) from function 8/int 0x13, but this function is limited to small hard disks (the maximum values are 1024 cylinders, 255 heads, 63 sectors: 8.4 GB at best). If enhanced BIOS disk calls are available (check by calling function 0x41/int 0x13), it gets the disk size from function 0x48/int 0x13 and calculates the number of cylinders.

How Is The Partition Table Written ?

CHS (Cylinder, Head, Sector) values are limited by a set number of bits for each value in the 16-byte partition table entries to: 1023,254,63. So LBA and CHS values can't be equal for HD bigger than 8GB. There are two ways to store the CHS value:

Testdisk Usage And Installation

Taken From the TestDisk Documentation

(Christophe Grenier, Simone Brandt, and Daniel B. Sed



First way:

- convert LBA to CHS, store (cylinder & 0x3FF, head & 0xFF, sector & 0x3F)
- It's what Partition Magic does (prior to version 8.0?).

Second way:

- convert LBA to CHS if cylinder <= 1023,
- store (cylinder & 0x3FF, head & 0xFF, sector & 0x3F)
- else
- store (1023, max_head & 0xFF, max_sector & 0x3F)

This is what Linux fdisk and TestDisk do. When TestDisk checks the partition table, it considers both ways may be correct. But the second way is better because start CHS is always lower or equal to end CHS.

Example: A hard disk's logical geometry is 255 heads per cylinder and 63 sectors per head. A partition begins at LBA=46781280 or CHS=2912,0,1. This partition ends at 3072,254,63.

First way start: 864, 0, 1 end: 0,254,63

Second way start: 1023,254,63 end: 1023,254,63

NB: $1023 = 0x3FF$ $(1023*255+254)*63+63-1=16450559$ $(2912*255+ 0)*63+ 1-1=46781280$

Partition Magic (before version 8.0?) considered the second way as invalid; even though it's an agreed upon standard. TestDisk handles both without complaining.

Limitations

TestDisk is now using a 64 bits internal data offset representation, IDE LBA28 addressing is limited to 128GB HD IDE LBA48 addressing is "limited" to 131072 TB PC/Intel MBR partition table is limited to 2TB. GUID Partition Table (GPT) can be up to 18 exabytes (Not yet supported by TestDisk) Most common limitation is to have an OS that doesn't handle IDE LBA48 addressing mode.