TestDisk Documentation

Release 7.1

Christophe GRENIER

Oct 16, 2018

Contents

CHAPTER 1

Presentation

TestDisk & PhotoRec are free and open-source data recovery utilities. TestDisk has been created in 1998 and PhotoRec in April 2002 by Christophe GRENIER, they can be downloaded from https://www.cgsecurity.org/. They are distributed under the GNU General Public License v2 or later, you can

- run the program as you wish, for any purpose,
- study how the program works, and change it so it does your computing as you wish (You have access to the source code.),
- redistribute copies so you can help your neighbor,
- distribute copies of your modified versions to others under the same license. By doing this you can give the whole community a chance to benefit from your changes.

Archives with ready-to-use binaries are available for

- DOS (32-bit x86)
- Microsoft Windows (32-bit x86 or 64-bit x64)
- Linux (32-bit x86 or 64-bit x64)
- Mac OS X (PowerPC or Intel) / OS X / macOS
- Marvell 88F628x Linux

TestDisk & PhotoRec can also be compiled for other platforms, notably

- FreeBSD/OpenBSD/NetBSD, Unix-like computer operating system descended from Berkeley Software Distribution (BSD), a Research Unix derivative developed at the University of California, Berkeley.
- Haiku, a free and open-source operating system compatible with the now discontinued BeOS.
- SunOS/Solaris, a Unix-branded operating system developed by Sun Microsystems for their workstation and server computer systems,

1.1 TestDisk - Partition recovery

TestDisk recognizes the following disk partitioning:

- Apple partition map
- GUID Partition Table
- Humax
- PC/Intel Partition Table (master boot record)
- Sun Solaris slice
- Xbox fixed partitioning scheme

It also handles non-partitioned media.

TestDisk can

- recover deleted partition
- rebuild partition table
- rewrite the Master boot record (MBR)

TestDisk does a quick check of the disk's structure and compares it with the partition table for entry errors. Next, it searches for lost partitions of these file systems:

- Be File System (BeOS)
- BSD disklabel (FreeBSD/OpenBSD/NetBSD)
- Cramfs, Compressed File System
- DOS/Windows FAT12, FAT16, and FAT32
- Windows exFAT
- HFS, HFS+ and HFSX, Hierarchical File System
- IBM Journaled File System 2 (JFS2)
- Linux ext2, ext3 and ext4
- 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
- Novell Storage Services (NSS)
- Windows New Technology File System (NTFS)
- ReiserFS 3.5, 3.6 and 4
- Sun Solaris i386 disklabel
- Unix File System UFS and UFS2 (Sun/BSD/...)
- XFS, SGI's Journaled File System

1.2 TestDisk - Filesystem repair

TestDisk can deal with some specific logical filesystem corruption:

- File Allocation Table, FAT12 and FAT16
  - Find filesystem parameters to rewrite a valid boot sector
  - Use the two copies of the FAT to rewrite a coherent version
- File Allocation Table, FAT32
  - Find filesystem parameters to rewrite a valid boot sector
  - Restore the boot sector using its backup
  - Use the two copies of the FAT to rewrite a coherent version
- exFAT
  - Restore the boot sector using its backup
- NTFS (New Technology File System) boot sector and MFT repair
  - Find filesystem parameters to rewrite a valid boot sector
  - Restore the boot sector using its backup
  - Restore the Master File Table (MFT) from its backup
- Extended file systems, ext2, ext3 and ext4
  - Find backup superblock location to assist fsck
- HFS+
  - Restore the boot sector using its backup

1.3 TestDisk - File recovery

When a file is deleted, the list of disk clusters occupied by the file is erased, marking those sectors available for use by other files created or modified thereafter. If the file wasn't fragmented and the clusters haven't been reused, TestDisk can recover the deleted file for various filesystem:

- FAT
- NTFS
- exFAT
- ext2

1.4 PhotoRec - File recovery

PhotoRec is a file carver data recovery software tool. It doesn't recover the original filenames but it can recover delete files even from corrupted filesystem. PhotoRec recognizes and recovers numerous file formats including ZIP, Office, PDF, HTML, JPEG and various graphics file formats. The whole list of file formats recovered by PhotoRec contains more than 480 file extensions (about 300 file families). It's possible to create custom signature to recover file format unknown to PhotoRec.

1.5 QPhotoRec - File recovery

QPhotoRec is a file carver data recovery software tool with a graphical user interface. Like PhotoRec, it doesn't recover the original filenames but it can recover delete files even from corrupted filesystem.

CHAPTER 2

Installation

2.1 Linux: Installation of distribution package

2.1.1 CentOS

As root,

```
yum install testdisk qphotorec
```

2.1.2 Fedora

As root,

```
dnf install testdisk qphotorec
```

2.1.3 Fedora Copr

Copr is an automatic build system. It provide the latest development version. As root,

```
dnf copr enable grenier/testdisk
dnf install testdisk qphotorec
```

2.1.4 Debian / Ubuntu

As root,

```
apt install testdisk
```

2.2 Official binaries

2.2.1 Official binaries: stable or WIP ?

Using the development version (WIP=Work In Progress) is usually recommended as fixes are not backported. The WIP archive may be modified several times per week but keep the same name. If this version doesn't start, you can always use the stable version and warn the developer of the problem with the beta version.

2.2.2 Installation of official binaries for Windows

- Download the archive (32-bit x86 or 64-bit x64) from https://www.cgsecurity.org/wiki/TestDisk_Download
- Extract all the files including the subdirectories

2.2.3 Installation of official binaries for Mac OS X

- **Download the archive from https://www.cgsecurity.org/wiki/TestDisk_Download**
  - Mac OS X Intel / OS X / macOS
  - Mac OS X PowerPC for very old Mac (Mac OS X <= 10.5)
- Extract all the files including the subdirectories

2.2.4 Installation of official binaries for Linux

Download the archive from https://www.cgsecurity.org/wiki/TestDisk_Download Currently we have

- https://www.cgsecurity.org/testdisk-7.0.linux26-x86_64.tar.bz2 for the last stable version
- https://www.cgsecurity.org/testdisk-7.1-WIP.linux26-x86_64.tar.bz2 for the development version

The archives contains static binaries for Intel (x86_64 or i686) platforms. They should work as-is on any recent Linux distribution.

Decompress the archive, no need to be root

```
tar xjf testdisk-7.1-WIP.linux26-x86_64.tar.bz2
```

List your files, a directory named testdisk-7.1-WIP should has been created.

CHAPTER 3

Building from source

Building the source code is usually reserved to

- developers wanting to add new features
- packagers wanting to create archive/package to distribute
- users who are using a platform for which no ready to use binaries are available

3.1 Compilation environment

testdisk uses several libraries if available:

- libncurses - Required, TestDisk and PhotoRec use a text user interface, Ncurses library and development files must be available.
- Ext2fs library - Optional, used by TestDisk to list files from ext2/ext3/ext4 partition and by PhotoRec to be able to carve the free space from an ext2/ext3 partition instead of the whole partition
- EWF library - Optional, TestDisk and PhotoRec use it to access Expert Witness Compression Format files (e.g. Encase files)
- Iconv - Optional, used to handle Unicode filenames
- Jpeg library - Optional, used by PhotoRec to improved JPEG recovery rate
- NTFS library - Optional, used by TestDisk to list files from NTFS partition
- Reiserfs library - Optional, used by TestDisk to list files from reiserfs partition
- zlib library - Optional, used by PhotoRec to decompress gzipped content
- Qt5 library - Optional, required for QPhotoRec and to update the configure script.

3.1.1 Linux

- Debian/Ubuntu: `apt-get install -y build-essential e2fslibs-dev libewf-dev libncurses5-dev libncursesw5-dev ntfs-3g-dev libjpeg-dev uuid-dev zlib1g-dev qtbase5-dev qttools5-dev-tools pkg-config dh-autoreconf git`
- RHEL/CentOS 5: `yum install -y buildsys-build e2fsprogs-devel libjpeg-devel ncurses-devel ntfs-3g-devel zlib-devel git`
- RHEL/CentOS 6 or later: `yum install -y @buildsys-build desktop-file-utils e2fsprogs-devel libewf-devel libjpeg-devel libuuid-devel ncurses-devel ntfs-3g-devel qt-devel qt5-qtbase-devel zlib-devel git`
- Fedora: `dnf install -y @buildsys-build desktop-file-utils e2fsprogs-devel libewf-devel libjpeg-devel libuuid-devel ncurses-devel ntfs-3g-devel qt-devel qt5-qtbase-devel zlib-devel git`

3.1.2 macOS

Install Xcode

3.1.3 Windows

cygwin

Cygwin https://cygwin.com/ is a large collection of GNU and Open Source tools which provide functionality similar to a Linux distribution on Windows, it includes the gcc compiler. A DLL (cygwin1.dll) provides substantial POSIX API functionality, such functions may be required by some libraries that TestDisk or PhotoRec can use.

MinGW-w64

MinGW-w64 https://mingw-w64.org/ is a free and open source software development environment for creating Microsoft Windows applications. It provides GCC for Windows 64 & 32 bits.

3.2 Cross Compilation environment

Using Linux, it's possible to generate binaries for Windows. Two cross-compiler toolchains are available under Fedora and CentOS 7 to create binaries for Windows 32 and 64 bits. All packages needed are available at

- Windows cygwin target: https://copr.fedorainfracloud.org/coprs/grenier/cygwin-testdisk/
- Windows mingw target: https://copr.fedorainfracloud.org/coprs/grenier/mingw-testdisk/

testdisk, photorec and fidentify official binaries are generated using cygwin, qphotorec using mingw.

3.3 Compilation

3.3.1 Compilation from source archive

Once you have downloaded the source archive from https://www.cgsecurity.org/wiki/TestDisk_Download, run

```
tar xjf testdisk-7.1-WIP.tar.bz2
cd testdisk-7.1-WIP
./configure && make
```

3.3.2 Compilation from git repository

```
git clone https://git.cgsecurity.org/testdisk.git
```

If you have already cloned the project, to update your local copy, run `git pull` from the testdisk directory.

```
cd testdisk
mkdir config
autoreconf --install -W all -I config
./configure
make
```

3.3.3 Compiling a static version

Once you have been able to build a "normal" version, you can try to build a static version.

```
make static
```

A static build is a compiled version of a program which has been statically linked against libraries. A static binary does not depend on library availability of the computer it's running on, usually you can copy this binary on another computer and it will work. It is still architecture specific (i.e. CPU) and may be kernel (OS version) dependent, so static binaries may be used for portable applications. For the build to be successful, you may have to install static version of libraries.

CHAPTER 4

Creating a live USB

If you need to repair a computer that isn't booting correctly, you can move its harddisk to a working computer or start your computer from an USB key or a DVD. It's this later solution that will be presented here.

You need an USB flash drive also known as USB stick, thumb drive, pen drive, or jump drive that you can erase. Note it's also possible to use a blank DVD.

Download Fedora "Image Live" from https://getfedora.org/fr/workstation/download/

4.1 Windows

- Download and run SUSE Studio ImageWriter or Rawrite32
- Choose the Fedora image as the **Image** (SUSE Studio) or **Filesystem image** (Rawrite32) - if the image file is not shown, you may have to change the file selector options or change the image's extension
- Choose the USB stick in the drop-down box by the **Copy** button (SUSE Studio) or as the **Target** (Rawrite32)
- Double-check you're really, really sure you don't need any of the data on the USB stick!
- Click **Copy** (SUSE Studio) or **Write to disk…** (Rawrite32)
- Wait for the operation to complete,

4.2 Linux (command line)

- Identify the name of the USB drive partition
- unmount all mounted partition from that device (Replace /run/media/user/mountpoint by the correct mountpoint)
- use *dd* to create do the copy (Adapt the source and destination)

```
lsblk
umount /run/media/user/mountpoint
sudo dd if=/path/to/image.iso of=/dev/sdX bs=8M status=progress oflag=direct
```

Wait until the command completes. If you receive `dd: invalid status flag: 'progress' error`, your dd version doesn't support `status=progress` option and you'll need to remove it (and you won't see writing progress).

4.3 Linux (GNOME)

This method is for people running Linux with GNOME, Nautilus and the GNOME Disk Utility installed. A standard installation of Fedora, or a standard GNOME installation of many other distributions, should be able to use this method. On Fedora, ensure the packages nautilus and gnome-disk-utility are installed. Similar graphical direct-write tools may be available for other desktops.

- Download a Fedora image, choose a USB stick that does not contain any data you need, and connect it
- Run Nautilus (Files) - for instance, open the Overview by pressing the Start/Super key, and type *Files*, then hit enter
- Find the downloaded image, right-click on it, go to **Open With**, and click **Disk Image Writer**
- Double-check you're really, really sure you don't need any of the data on the USB stick!
- Select your USB stick as the **Destination**, and click **Start Restoring...**
- Wait for the operation to complete, then reboot your computer, and do whatever you need to do to boot from a USB stick - often this will involve pressing or holding down **F12**, **F2** or **Del**.

4.4 OS X

- Open a terminal
- Run `diskutil list`. This will list all disks connected to the system, as `/dev/rdisk1`, `/dev/rdisk2` and so on. Identify - very carefully! - which one corresponds to the USB stick you wish to use. Hereafter, we'll assume it was `/dev/rdisk2` - modify the commands as appropriate for your stick.
- Run `diskutil unmountDisk /dev/rdisk2`
- Type `dd if=`, then drag and drop the Fedora image file to the terminal window - this should result in its filesystem location being appended to the command. Now complete the command with `of=/dev/rdisk2 bs=1m`, but *don't hit Enter yet*. You should wind up with something like `sudo dd if=/Volumes/Images/Fedora-Live-Desktop-x86_64-20-1.iso of=/dev/rdisk2 bs=1m`
- Double-check you have the correct disk number and you're really, really sure you don't need any of the data on the USB stick!
- Hit Enter

4.5 Starting from the USB stick

Plug the USB key on the damaged computer and boot this computer, and do whatever you need to do to boot from a USB stick - often this will involve pressing or holding down **F12**, **F2** or **Del**. If you are using a Mac computer, hold down the left Alt/Option key to access the boot menu - you should see a Fedora logo. Click this to boot.

Original source of this page: https://fedoraproject.org/wiki/How_to_create_and_use_Live_USB

CHAPTER 5

Storage: can I repair it or recover data from it ?

There are 3 kinds of storage:

- Direct Attached Storage (DAS) or local storage for hard disks connected via * IDE/PATA * SATA/eSATA * SAS * firewire * devices connected via USB (external disk, digital camera, thumb drive, phone...) in USB mass storage mode
- Storage Area Networks (SAN) * Fibre Channel Protocol (FCP) * Fibre Channel over Ethernet (FCoE) * iSCSI, mapping of SCSI over TCP/IP
- Network Attached Storage (NAS) * Windows share (CIFS/SMB) * Network File System (NFS) * Phone or digital camera in Media Transfer Protocol (MTP) mode (even if connected via USB)

TestDisk & PhotoRec can recover data from DAS and SAN storage. For NAS server (QNAP, Synology...), they need to run on the server itself or the disks need to be moved to a computer running Linux (sometimes FreeBSD). TestDisk & PhotoRec can store recovered data on any storage available from your computer. When recovering deleted files, be careful to avoid writing new data to the same partition the files were stored on.

CHAPTER 6

Starting the tools

6.1 Disk image

TestDisk and PhotoRec can be used on disk image:

- raw files (.dd)
- Encase (.E01)
- splitted Encase files (.E01, E02...)

Splitted raw files are not supported. No administrator rights are needed to run testdisk or photorec on disk image.

Examples:

- `photorec image.dd` to carve a raw disk image
- `photorec image.E01` to recover files from an Encase EWF image
- `photorec 'image.???'` if the Encase image is split into several files.

6.2 Running TestDisk, PhotoRec or QPhotoRec under Windows

Double-click on the executable (`testdisk_win.exe`, `photorec_win.exe` or `qphotorec_win.exe`) from an account in the Administrator Group. Administrator rights are necessary to get a low-level access to all medias (hard disk, USB key, Smart Card, etc.). Windows UAC (Vista and later) will ask you to confirm that you want to run the executable with administrator rights.

Note: Windows users, if you see `cygwin1.dll not found, c\\cygwin is missing`, extract all the files from the archive before running TestDisk or PhotoRec.

6.3 Running TestDisk, PhotoRec under Linux

You need to be root to run TestDisk.

```
cd testdisk-7.0
sudo ./testdisk_static
```

```
cd testdisk-7.0
sudo ./photorec_static
```

Note: If your Raid device (ie. Intel raid) is missing, run "sudo dmraid -ay" to activate it.

6.4 Running QPhotoRec under Linux X.org X11

QPhotoRec is a Qt5 application, it isn't shipped with the official Linux binaries from www.cgsecurity.org. But it is available on most Linux distribution or can be compiled from source. To run it in a Terminal,

```
sudo qphotorec
```

6.5 Running QPhotoRec under Linux Wayland

To run QPhotoRec in a Terminal,

```
xhost +local:
sudo qphotorec
```

6.6 Running TestDisk, PhotoRec under macOS

If you are not root, TestDisk (i.e. `testdisk-7.0/testdisk`) or PhotoRec will restart itself using `sudo` after confirmation from your part.

If your administrator account has no password (a blank password), you must give that user a password before using the sudo command:

- Choose Apple menu > System Preferences and click Accounts.
- Click Change Password.

Terminal doesn't show the password as you type. If you enter the wrong password or a blank password, the command isn't executed and Terminal asks you to try again.

6.7 Running fidentify under Windows

fidentify checks all the files from a directory with the same signatures than photorec. It's useful to check if PhotoRec is able to recover some file extensions/some file formats. Run `cmd`, Windows Command Prompt. `cd` is the command to change directory.

```
cd testdisk-7.0
fidentify_win.exe d:\directory
```

6.8 Running fidentify under Linux or macOS

Start a terminal, go in testdisk directory and use fidentify to check if the files present in a directory are recognized. This identification is identical in PhotoRec.

```
cd testdisk-7.0
./fidentify_static /home/user/
```

CHAPTER 7

Repairing filesystem

Repairing a filesystem may be a risky business as sometimes the problem is "fixed" by removing all invalid files. So if you have access to some of your files but not all, it's recommended to backup what it's possible to access before trying to repair the filesystem.

7.1 Repairing filesystems from Windows

Windows can read and write files from FAT, exFAT and NTFS filesystem. The *chkdsk* command is used to check and repair filesystems. Run *cmd* (Right-click Run As Administrator)

```
chkdsk /f d:
```

7.2 Repairing filesystems from Linux

Linux can read and write from a large variety of filesystems. The *fsck* generic command is used to run a filesystem check. To check and repair automaticaly the filesystem on /dev/sda, run as root

```
fsck -y /dev/sda1
```

fsck starts a filesystem specific command, in example for ext4, it run *fsck.ext4*. If you need a fine grained repair, you should read the man page of the command related to the filesystem you want to repair, i.e. *man fsck.ext4*. If some files or directories are missing, remember to check the *lost+found* directory at the root of this filesystem.

7.3 Repairing filesystems from macOS

To check an external drive,

```
sudo diskutil list
sudo fsck /dev/disk1s1
```

You may have to repeat the fsck command several times until no remaining error is reported.

If you get Invalid b-tree node size, you can try

```
sudo fsck_hfs -r -d /dev/disk1s1
```

7.4 Repairing FAT32, exFAT and NTFS boot sector using TestDisk

The boot sector is a sector containing information required to access any files from a FAT, exFAT or NTFS filesystem. FAT32 and NTFS filesystems have a main boot sector and a backup. If the main boot sector is damaged, the filesystem is listed as raw or unreadable. TestDisk is able to use the backup boot sector to repair the main boot sector:

- start testdisk
- select the device containing the partition (avoid drive letter like D:)
- confirm the partition table type
- go in the Advanced menu
- select the partition
- choose Boot

If the boot sector is damaged, *Boot sector: Bad* will be shown. If the backup is OK, *Backup boot sector: Ok* will also be listed.

- choose BackupBS
- confirm
- Quit
- restart the computer

7.5 TestDisk: Repairing FAT boot sector

The first sector of a FAT filesystem is named boot sector. It contains the main filesystem properties and some small code necessary only to start the computer from this partition. If the boot sector is damaged, it's impossible to access your data. Windows *chkdsk* or Linux *fsck* can not repair a filesystem without a valid boot sector, they return error message like *Chkdsk is not available for RAW drives*. Fortunately TestDisk can find all the parameters that need to be recorded in the boot sector and rewrite this sector, so further repair operations or normal access can be conducted.

- start testdisk
- select the device containing the partition (avoid drive letter like D:)
- confirm the partition table type
- go in the Advanced menu
- select the FAT partition
- choose Boot
- select RebuildBS

- choose List

If testdisk is able to list your files, choose

- quit the file listing
- choose Write
- confirm
- Quit
- restart the computer

7.6 TestDisk: Repairing NTFS boot sector

The first sector of a NTFS filesystem is named boot sector. It contains the main filesystem properties and some small code necessary only to start the computer from this partition. If the boot sector is damaged, it's impossible to access your data. Windows *chkdsk* or Linux *fsck* can not repair a filesystem without a valid boot sector, they return error message like *Chkdsk is not available for RAW drives*. Fortunately TestDisk can find all the parameters that need to be recorded in the boot sector and rewrite this sector, so further repair operations or normal access can be conducted.

- start testdisk
- select the device containing the partition (avoid drive letter like D:)
- confirm the partition table type
- go in the Advanced menu
- select the NTFS partition
- choose Boot
- select RebuildBS
- choose List

If testdisk is able to list your files, choose

- quit the file listing
- choose Write
- confirm
- Quit

7.7 TestDisk: repairing ext2/3/4 filesystem superblock

1024 bytes after the beginning of the ext2/3/4 filesystem sits the superblock. It contains the main filesystem properties. With a damaged main superblock, it's not possible to mount and access the files normally. Fortunately copies are the main superblock are spread over the filesystem. To be precise, they are not exact copy of the main superblock, each copy contains its own location to prevent confusion between copies and the original. TestDisk can search for alternate superblocks.

- start testdisk
- select the device containing the partition
- confirm the partition table type

- go in the Advanced menu
- select the Linux partition
- choose SuperBlock

```
TestDisk 7.1-WIP, Data Recovery Utility, August 2016
Christophe GRENIER <grenier@cgsecurity.org>
http://www.cgsecurity.org

Disk /dev/sda - 2000 GB / 1863 GiB - CHS 243201 255 63

     Partition                  Start        End    Size in sectors

  MS Data                     2048 3907020799 3907018752 [/home2]
superblock 0, blocksize=4096 [/home2]
superblock 32768, blocksize=4096 [/home2]
superblock 98304, blocksize=4096 [/home2]
superblock 163840, blocksize=4096 [/home2]
superblock 229376, blocksize=4096 [/home2]
superblock 294912, blocksize=4096 [/home2]
superblock 819200, blocksize=4096 [/home2]
superblock 884736, blocksize=4096 [/home2]
superblock 1605632, blocksize=4096 [/home2]
superblock 2654208, blocksize=4096 [/home2]

To repair the filesystem using alternate superblock, run
fsck.ext4 -p -b superblock -B blocksize device

>[  Quit  ]
                            Return to Advanced menu
```

If superblock 0 is listed, it means the main superblock is correct. If it's damaged, this line will be missing, use next superblock and block size information to run fsck.

```
fsck.ext4 -p -b 32768 -B 4096 /dev/sda1
```

7.8 Repairing HFS/HFS+ volume header using TestDisk

The volume header is locate 1024 bytes after the beginning of the HFS/HFS+ filesystem. If it is damaged, it is not possible to access files normally. TestDisk is able to use the backup volume header to repair the main volume header:

- start testdisk
- select the device containing the partition
- confirm the partition table type
- go in the Advanced menu
- select the partition
- choose SuperBlock

If the main superblock is damaged, *Volume header: Bad* will be shown. If the backup is OK, *Backup volume header: HFS+ Ok* (or HFS Ok) will also be listed. In this case,

- choose BackupBS
- confirm

- Quit
- restart the computer

7.9 Repairing Bitlocker volume

Repair-bde can reconstruct critical parts of the drive and salvage recoverable data as long as a valid recovery password or recovery key is used to decrypt the data. See https://technet.microsoft.com/en-us/library/ff829851(v=ws.11).aspx

CHAPTER 8

Recovering deleted files using TestDisk

When a file is deleted, the data remains on the disk. Unless new data has overwritten your lost file, TestDisk can usually recover it. It's possible for

- FAT12/16/32
- exFAT
- NTFS
- ext2

For other filesystems or if sought-after lost files are still missing, give PhotoRec a try. PhotoRec is a signature based file recovery utility and may be able to recover your data where other methods failed.

- Do not further use the media (HDD, USB key, ...) on which the data stored have been deleted until data recovery process is completed.
- It is highly recommended that TestDisk or PhotoRec recovers files on another destination media, at minimum on another filesystem.

For maximum security, TestDisk doesn't try to unerase files but lets you copy the deleted files onto another partition or disk. Remember, you must avoid writing anything on the filesystem that was holding the data. If you do, deleted files may be overwritten by new ones.

8.1 TestDisk: undelete file for FAT, exFAT, ext2

FAT is mainly used on memory cards from digital cameras and on USB keys. When a file is deleted, the filename is marked as deleted and the data area as unallocated/free, but TestDisk can read the deleted directory entry and find where the file began. If the data area hasn't been overwritten by a new file, the file is recoverable.

exFAT can be found on large memory card, large USB keys and hard disk.

ext2 is a Linux filesystem. It has been superseded by ext3 and ext4, so it's not found often now. With ext3 and ext4, it's possible to find the names of the deleted files but the location of the deleted data isn't available anymore, so even if ext3/ext4 is similar to ext2, it's not possible to recover lost files using TestDisk.

8.1.1 Start testdisk

- *Running TestDisk, PhotoRec or QPhotoRec under Windows*
- *Running TestDisk, PhotoRec under Linux*
- *Running TestDisk, PhotoRec under macOS*

8.1.2 Log creation

- Choose Create unless you have a reason to append data to the log or if you execute TestDisk from read only media and can't create it elsewhere.
- Press Enter to proceed.

8.1.3 Disk selection

All hard drives should be detected and listed with the correct size by TestDisk.

- Use up/down arrow keys to select your hard drive with the lost partition/s.
- Press Enter to Proceed.

macOS If available, use raw device `/dev/rdisk*` instead of `/dev/disk*` for faster data transfer.

8.1.4 Partition table type selection

TestDisk displays the partition table types.

- Select the partition table type - usually the default value is the correct one as TestDisk auto-detects the partition table type.
- Press Enter to Proceed.

8.1.5 Start the undelete process

- Select **Advanced**
- Select the partition that was holding the lost files and choose **Undelete**

8.1.6 File undelete

Navigate to the folder where your files were. Deleted files and directories are displayed in red.

- To undelete a file, select the file to recover and press 'c' to copy the file.
- To recover a deleted directory, select the directory and press 'c' to undelete the directory and its content.

8.1.7 Select where recovered files should be written

Select the destination

8.1.8 File recovery is completed

When you get your files back, use Quit to exit.

If testdisk has been unable to find your lost data, try PhotoRec instead.

8.2 TestDisk: undelete file for NTFS

8.2.1 Start testdisk

- *Running TestDisk, PhotoRec or QPhotoRec under Windows*
- *Running TestDisk, PhotoRec under Linux*
- *Running TestDisk, PhotoRec under macOS*

8.2.2 Log creation

- Choose Create unless you have a reason to append data to the log or if you execute TestDisk from read only media and can't create it elsewhere.
- Press Enter to proceed.

8.2.3 Disk selection

All hard drives should be detected and listed with the correct size by TestDisk.

- Use up/down arrow keys to select your hard drive with the lost partition/s.
- Press Enter to Proceed.

macOS If available, use raw device `/dev/rdisk*` instead of `/dev/disk*` for faster data transfer.

8.2.4 Partition table type selection

TestDisk displays the partition table types.

- Select the partition table type - usually the default value is the correct one as TestDisk auto-detects the partition table type.
- Press Enter to Proceed.

8.2.5 Start the undelete process

- Select **Advanced**
- Select the partition that was holding the lost files and choose **Undelete**

8.2.6 NTFS file undelete

TestDisk scans MFT entries for deleted files. A list of NTFS deleted files found by TestDisk is displayed

- To recover a single file, highlight the file and press 'c' (lowercase) to copy it.
- To recover a several files, move the first file you want to recover, press ':' to select it, repeat the process for the others files, press 'C' (uppercase) to copy them

It's not visible in interface but it's possible to filter the results, press 'f' to add a filter. Several filters can be added. To cancel all the filters, press 'r' (reset).

8.2.7 Select where recovered files should be written

Select the destination

8.2.8 File recovery is completed

When the NTFS file recovery is finished, choose Quit to exit.

If testdisk has been unable to find your lost data, try PhotoRec instead.

CHAPTER 9

Recovering deleted partition using TestDisk

When a partition is deleted or if the partition table is corrupted, the filesystems remain on the disk but their location is unknown and no data can be accessed. TestDisk can search partitions and rewrite the partition table with the partitions selected by the user.

9.1 Start testdisk

- *Running TestDisk, PhotoRec or QPhotoRec under Windows*
- *Running TestDisk, PhotoRec under Linux*
- *Running TestDisk, PhotoRec under macOS*

9.2 Log creation

- Choose Create unless you have a reason to append data to the log or if you execute TestDisk from read only media and can't create it elsewhere.
- Press Enter to proceed.

Note: Windows users, if you have difficulties to find the testdisk.log file, consult https://support.microsoft.com/en-us/kb/865219 on how to show file name extensions in Windows Explorer.

9.3 Disk selection

All hard drives should be detected and listed with the correct size by TestDisk.

- Use up/down arrow keys to select your hard drive with the lost partition/s.

- Press Enter to Proceed.

Note: macOS - If available, use raw device `/dev/rdisk*` instead of `/dev/disk*` for faster data transfer.

Warning: Windows - Do not select C:, D: or another drive letter. It's useless to search partitions inside a partition.

9.4 Partition table type selection

TestDisk displays the partition table types.

- Select the partition table type - usually the default value is the correct one as TestDisk auto-detects the partition table type.
- Press Enter to Proceed.

9.5 Analyze current partition table

- Select **Analyse**
- Confirm with the Enter key
- TestDisk will list the current partition table.

If a partition is damaged or a partition entry corrupted, the problem will be listed and the partition listed twice. By example, if you see "Invalid NTFS or exFAT boot" on a partition (partition size is OK, the partition doesn't overlap another one...) you want to access, it's better to fix this problem (*TestDisk: Repairing NTFS boot sector*) before searching other partitions.

- Confirm at **Quick Search** to proceed

9.6 Quick Search for partitions

TestDisk displays the first results in real time. If necessary, you can choose Stop to abort the quick search. TestDisk lists all partitions it has found. To list the files of a FAT, exFAT, NTFS, ext2/3/4 filesystem, highlight this partition and press **P**. Press **Q** to return to the partition list.

9.7 Search for more partitions

If a partition is still missing, choose **[Deeper Search]**. It can take a few hours, so you need to be certain that your computer will not sleep (Power management feature...)

9.8 Partitions selection

Partitions listed as D(eleted) will not be recovered if you let them listed as deleted. Use the arrow keys to switch the partitions you want to recover (check the partition size, list the file contents...) from D(eleted) to *(bootable), P(rimary) or L(ogical). Only one partition can be listed as *(bootable). It is not a problem if a partition is marked as

bootable on a disk you will not start from (e.g. an external disk) but there MUST be a bootable partition on a disk you want to start your computer from.

Once all the partitions you want to keep and all the partitions you want to recover are properly marked as non deleted, continue on next screen. Review the partitions list. If all partitions are listed and only in this case, confirm at Write with Enter, y and OK. Now, the partitions are registered in the partition table.

If a FAT32 or an NTFS partition was found using its backup boot sector, TestDisk will let you rewrite the main boot sector with the content of the backup boot sector: to copy the backup of the boot sector over the boot sector, select Backup BS, validate with Enter, use y to confirm.

Restart your computer.

CHAPTER 10

How to make the system bootable again

Check that

- all partitions are listed in the partition table
- a partition with your computer os is listed as *(bootable)
- you can list the files from the bootable partition

10.1 DOS - Window 95/98

If your OS doesn't boot, you can reinstall the system files with `sys c:`.

10.2 Windows 2000/XP/2003

- Run fixmbr from the Recovery Console

```
fixmbr \Device\HardDisk0
```

If you still have the problem,

- Run *fixboot* to repair NTFS boot sector.
- Check `c:\boot.ini` content

10.3 Windows Vista/Windows 7/..., Windows Server 2008/...

- Run `bootrec.exe /fixmbr` from the Recovery Console
- For legacy / PC Intel partition table, check `c:\boot.ini` content
- For EFI GPT, check the output of `bcdedit /v`. To modify the settings, use the `bcdedit /set` command.

- Run `bootrec.exe /fixboot` to repair NTFS boot sector.

10.4 Linux/FreeBSD

- Update your /etc/fstab to reflect the new partition order.
- Update your multiboot configuration
  - Lilo: /etc/lilo.conf
  - Grub: /boot/grub/grub.conf
  - Grub2: /etc/grub2-efi.cfg
- Reinstall the multiboot in the Master Boot Record.

```
lilo
grub-install device
grub2-install device
```

CHAPTER 11

Recovering deleted files using PhotoRec

PhotoRec doesn't recover the original filenames or the file structure but it can recover lost files even from corrupted filesystem. PhotoRec is a signature based file recovery utility (a file carver) and may be able to recover your data where other methods failed.

Remember, you must avoid writing anything on the filesystem that was holding the data. If you do, deleted files may be overwritten by new ones.

11.1 Start photorec

- *Running TestDisk, PhotoRec or QPhotoRec under Windows*
- *Running TestDisk, PhotoRec under Linux*
- *Running TestDisk, PhotoRec under macOS*

11.2 Disk selection

Available media are listed. Use up/down arrow keys to select the disk that holds the lost files.

- Use up/down arrow keys to select your hard drive with the lost partition/s.
- Press Enter to Proceed.

Hint for macOS: If available, use raw device `/dev/rdisk*` instead of `/dev/disk*` for faster data transfer.

11.3 Source partition selection

Choose

- `Search` after selecting the partition that holds the lost files to start the recovery,

- `Options` to modify the options,
- `File Opt` to modify the list of file types recovered by PhotoRec.

11.4 PhotoRec options

- `Paranoid` By default, recovered files are verified and invalid files rejected. Enable `bruteforce` if you want to recover more fragmented JPEG files, note it is a very CPU intensive operation, it's started after the normal scan process.
- The `expert mode` option allows the user to force the file system block size and the offset. Each filesystem has his own block size (a multiple of the sector size) and offset (0 for NTFS, exFAT, ext2/3/4), these value are fixed when the filesystem has been created/formatted. When working on the whole disk (i.e. original partitions are lost) or a reformatted partition, if PhotoRec has found very few files, you may want to try the minimal value that PhotoRec let you select (it's the sector size) for the block size (0 will be used for the offset).
- Enable `Keep corrupted files` to keep files even if they are invalid in the hope that data may still be salvaged from an invalid file using other tools.
- Enable `Low memory` if your system does not have enough memory and crashes during recovery. It may be needed for large file systems that are heavily fragmented. Do not use this option unless absolutely necessary.

11.5 Selection of files to recover

In `FileOpts`, enable or disable the recovery of certain file types, for example,

```
[X] riff RIFF audio/video: wav, cdr, avi
...
[X] tif  Tag Image File Format and some raw file formats (pef/nef/dcr/sr2/cr2)
...
[X] zip  zip archive including OpenOffice and MSOffice 2007
```

The whole list of file formats recovered by PhotoRec contains more than 300 file families representing more than 480 file extensions.

11.6 File system type

Once a partition has been selected and validated with `Search`, PhotoRec needs to know how the data blocks are allocated. Unless it is an ext2/ext3/ext4 filesystem, choose `Other`.

11.7 Carve the partition or unallocated space only

PhotoRec can search files

- from the whole partition (useful if the filesystem is corrupted) or
- from the unallocated space only (available for ext2/ext3/ext4, FAT12/FAT16/FAT32 and NTFS). With this option only deleted files are recovered.

11.8 Select where recovered files should be written

Choose the directory where the recovered files should be written. Use the arrow keys (up, down, left, right) to navigate, you can also use the enter key to enter into a directory.

- Dos/Windows/Os2: To get the drive list (C:, D:, E:, etc.), use the arrow keys to select `..`, press the `Enter` key - repeat until you can select the drive of your choice. Validate with `Y` es when you get the expected destination.
- Linux: File system from external disk may be available in a `/media`, `/mnt` or `/run/media` sub-directory. Mount your destination drive if necessary.
- macOS: Partitions from external disk are usually mounted in `/Volumes`.

Warning: Do not store the recovered files on the source filesystem. Otherwise lost data may be overwritten and definitively lost.

11.9 Recovery in progress

Number of recovered files is updated in real time.

- During pass 0, PhotoRec searches the first 10 files to determine the blocksize. This step is skipped when searching files from the unallocated space only, the blocksize value found in the filesystem structure is used.
- During pass 1 and later, files are recovered including some fragmented files.

Recovered files are written in recup_dir.1, recup_dir.2... sub-directories. It's possible to access the files even if the recovery is not finished.

11.10 Recovery is completed

When the recovery is complete, a summary is displayed. Note that if you interrupt the recovery, the next time PhotoRec is restarted you will be asked to resume the recovery.

- Thumbnails found inside pictures are saved as `t*.jpg`
- If you have chosen to keep corrupted files/file fragments, their filenames will beginning by the letter `b` (roken).
- Windows: You may have disabled your live antivirus protection during the recovery to speed up the process, but it's recommended to scan the recovered files for viruses before opening them - PhotoRec may have undeleted an infected document or a Trojan.
- Hint: When looking for a specific file. Sort your recovered files by extension and/or date/time. PhotoRec uses time information (metadata) when available in the file header to set the file modification time.

Note: Windows - You may need to take ownership of the recup_dir.* folders: https://technet.microsoft.com/en-us/library/Cc753659.aspx

Note: macOS / Linux - To change the owner of the files, run `sudo chown -R username recup_dir.*`

11.11 PhotoRec: file name and date

By default, files are saved in directories named recup_dir.1, recup_dir.2... A new directory is created each new 500 files (The thumb files are not included in this count, nor the report.xml file). A filename begins by a letter followed by a number (7 digits or more) and ends, if any, by a file extension.

Letter meaning:

- f=file
- b=broken
- t=jpeg embedded thumbnail

The number is calculated by using the file location minus the partition offset divided by the block size. For some filesystems like NTFS, exFAT, ext2/3/4, this number may be identical to the original cluster/block number. Using metadata information embedded in the recovered file, the file may be renamed to include the documentation title (example, Microsoft Office doc/xls/ppt or Acrobate pdf files) like `recup_dir.1/f0016741_Prudent_Engineering_Practice_for_Cryptographic_Protocols.pdf`.

By default, the file creation and modification times are corresponding to the data recovery time. Some file format may embedded date/time information (ie. jpg pictures taken by a digital camera, Microsoft Office documents), PhotoRec will try to reuse them. This way, it may be easier to sort the recovered files. For forensics purpose, do not trust this information blindly: the date/time information may be off by a few hours (no or wrong timezone information) or totally wrong (the original device clock may have a wrong date/time setting.)

CHAPTER 12

Creating custom signature for PhotoRec

PhotoRec recognizes numerous file formats. More than 480 file extensions (about 300 file families) are referenced. In example, PhotoRec is able to identify the JPEG file format and it can recover lost files using this format whatever the original file extension (jpg, jpeg, JPG...).

To check if a file format is already recognized, you can

- consult the file formats.
- submit a sample file to the PhotoRec online checker.
- use fidentify on a file sample (See *Running fidentify under Windows* or *Running fidentify under Linux or macOS*)

```
[kmaster@adsl ~]$ fidentify /home/kmaster/src/testfiles/sample.pfi
/home/kmaster/src/testfiles/sample.pfi: unknown
```

In this case, the file type is listed as **unknown**, so PhotoRec can't recover this kind of file, at least for the moment. We will check if it's possible to add a custom signature for it.

If instead of unknown an extension is listed, PhotoRec knows this file format, it may recover the file with another extension than the extension you are used to.

12.1 Signature Syntax

The file must contain one signature definition per line. A signature is composed of

- extension name
- offset of the signature
- signature or magic value

The magic value can be composed of

- a string, e.g. "data". Special characters can be escaped like "b", "n", "r", "t", "0" or "".

- hexadecimal data, e.g. 0x12, 0x1234, 0x123456... Note that *0x123456*, *0x12 0x34 0x56* and *0x12, 0x34, 0x56* are equivalents.
- space or comma delimiters are ignored

By using an hexadecimal editor, you can see that the pfi file from our example begins by a distinctive string *PhotoFiltre Image* at offset 0.

```
[kmaster@adsl ~]$ hexdump -C /home/kmaster/src/testfiles/sample.pfi | head
00000000  50 68 6f 74 6f 46 69 6c  74 72 65 20 49 6d 61 67  |PhotoFiltre Imag|
00000010  65 03 40 06 00 00 b0 04  00 00 40 19 01 00 40 19  |e.@.......@...@.|
00000020  01 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
```

The signature can be written as

```
pfi 0 "PhotoFiltre Image"
```

or

```
pfi 0 "PhotoFiltre", 0x20, "Image"
```

or if you prefer hexadecimal

```
pfi 0 0x50686f746f46696c74726520496d616765
```

From fidentify/PhotoRec point of view, the signatures are identical.

Warning: Be careful, hexdump displays non-printable chars as dots. The following signature is wrong:

```
pfi 0 "PhotoFiltre Image."
```

This signature using an hexadecimal value instead of a dot is correct:

```
pfi 0 "PhotoFiltre Image", 0x03
```

12.2 File location

PhotoRec searches for the signature file named

- Windows: *photorec.sig* in the *USERPROFILE* or *HOMEPATH* directory, e.g. *C:\Documents and Settings\bob* or *C:\Users\bob*.
- Linux and macOS: *.photorec.sig* in the *HOME* directory, e.g. */home/bob*
- *photorec.sig* in the current directory

This file doesn't exist by default, you need to create one. Using a text editor (e.g. notepad, vim...), create the signature file and add the signature you have identified.

12.3 Check your custom signature with fidentify

fidentify now perfectly identify the file

```
[kmaster@adsl ~]$ fidentify /home/kmaster/src/testfiles/sample.pfi
/home/kmaster/src/testfiles/sample.pfi: pfi
```

If fidentify doesn't recognize the signature,

- check your signature, it may be incorrect
- **verify that the signature file is a true ASCII text file**. It must not begin by *EF BB BF* (UTF-8 Byte Order Mark) or *FF FE* (UTF-16 LE BOM) by example.
- verify the filename of your signature file

12.4 Run PhotoRec

You are now ready to use PhotoRec with your custom signature to recover your files. If a signature file is present, PhotoRec will use it by default.

12.5 Improved file recover

To control all aspects of the recovery (file content check, file size control, footer detection...), the best way to add a signature, if you are developer, is to modify PhotoRec itself.

Commercial support is also available from the author grenier@cgsecurity.org.

CHAPTER 13

Recovering lost videos from a memory card using PhotoRec

Due to the way videos are recorded, all videos created by some digital camera (i.e. Canon 5D Mark III, Panasonic DMC-TZ80's photos in burst mode) are fragmented on the memory card. Data recovery software, photorec included, expect non fragmented files.

If all videos (.mov / .mp4) recovered by PhotoRec are unreadable, you are probably in this case. Note this chapter does not concern copies or downloaded files, only files written by some digital camera, not by your computer.

When using PhotoRec, in FileOpts, enable

```
[X] mov/mdat Recover mdat atom as a separate file
```

and next start the recovery.

If you sort the files by name, you should see that the names alternates between _ftyp.mov and _mdat.mov. You need to concatenate each ftyp file with a mdat file:

- If using Windows, run `cmd` to start a terminal, use `cd directory_name` to go where your files are, and run

```
type file2_ftyp.mov file1_mdat.mov > test.mov
```

If you do not have the permissions to write to the directory, before using the type command, take ownership of the directories or run `cmd` using right click run as administrator.

- Under macOS and Linux, start a terminal/console, use `cd directory_name` to go where your files are, and run

```
cat file2_ftyp.mov file1_mdat.mov > test.mov
```

f you do not have the permissions to write to the directory, before using the cat command, change the files and directories ownership using `chown -R username:groupname recup_dir.*`

Play the resulting test.mov file. If it works, you need to do the same with each couple of files.

This solution works only for videos written in two fragments. Videos from GoPro HD2, Hero3-Black Edition, HERO4 Silver are stored in more than 2 fragments, so special software solutions are needed to recover such videos. This chapter does not concern copies or downloaded files, only files written by some digital camera, not by your computer.

CHAPTER 14

After using PhotoRec

Usually PhotoRec and QPhotorec recover a lot of files but without the original filenames, it may be hard to locate the files you are interested in.

14.1 Sorting the files by extension

14.1.1 Using a powershell script under Windows

https://github.com/lconte/Copy-PhotoRecFilesbyExtension.ps1

14.1.2 Using a Python script

Python comes preinstalled on macOS and most Linux distribution. It can also be installed under Windows. The Python program sort-PhotorecRecoveredFiles

- sorts all files by file extensions into own folders.
- limits the number of files/folder by creating subfolders if a certain numbers is exceeded. The file/folder number can be customized.
- For all ‘’‘jpgs’’’: it put them into their own folders per year (EXIF-Data). Within a year, folders for every event are created, e.g. all photos taken at one weekend or vacation are sorted into one folder.

14.2 Renaming files using exiftool

exiftool can use meta-data from several popular file formats to rename files. All Linux distributions comes with a package for exiftool (perl-Image-ExifTool for RedHat, CentOS and Fedora) but otherwise it is available for Windows, Linux and macOS from http://www.sno.phy.queensu.ca/~phil/exiftool/

```
exiftool -r -ext jpg '-FileName<DateTimeOriginal' -d sorted_jpg/%Y%m%d/%Y%m%d_%H%M%S%
↪%-c.%%e jpg/
exiftool -r -ext tif '-FileName<DateTimeOriginal' -d sorted_tif/%Y%m%d/%Y%m%d_%H%M%S%
↪%-c.%%e tif/
exiftool -r -ext avi '-FileName<DateTimeOriginal' -d avi/%Y%m%d_%H%M%S%%-c.%%e avi/
exiftool -r -ext doc '-FileName<CreateDate' -d doc/%Y%m/%%f.%%e doc/
exiftool -r -ext mov '-FileName<CreateDate' -d mov/%Y%m%d_%H%M%S%%-c.%%e mov/
exiftool -r -ext mp3 '-FileName<mp3/${artist;} - ${Album;} - ${Track;} - ${Title;}%-c.
↪%e' mp3/
exiftool -r -ext mp4 '-FileName<CreateDate' -d mp4/%Y%m%d_%H%M%S%%-c.%%e mp4/
exiftool -r -ext m4p '-FileName<m4p/${Artist;} - ${Album;} - ${Title;}%-c.%e' m4p/
exiftool -r -ext mkv '-FileName<%f_${Title;}%-c.%e' mkv/
exiftool -r -ext ttf '-FileName<ttf/${FontName;}%-c.%e' ttf/

exiftool -r -ext jpg '-FileName<IMG_${FileIndex}%-c.%e' recup_dir.*
```

14.3 Removing duplicated files

Under Linux, fslint can be used to remove duplicated files

```
/usr/share/fslint/fslint/findup -d jpg/
```

CHAPTER 15

SMART status - Disk health monitoring

The smartmontools package contains two utility programs (smartctl and smartd) to control and monitor storage systems using the Self-Monitoring, Analysis and Reporting Technology System (SMART) built into most modern ATA/SATA, SCSI/SAS and NVMe disks. In many cases, these utilities will provide advanced warning of disk degradation and failure.

This package is installed by default on most Linux distribution. For Windows and macOS, there are respectively a setup.exe and an dmg available from https://sourceforge.net/projects/smartmontools/files/smartmontools/

```
sudo smartctl -a /dev/sda
=== START OF INFORMATION SECTION ===
Model Family:     Western Digital Green
Device Model:     WDC WD20EZRX-00D8PB0
Serial Number:    WD-WMC4M0875073
LU WWN Device Id: 5 0014ee 058f9952c
Firmware Version: 80.00A80
User Capacity:    2,000,398,934,016 bytes [2.00 TB]
Sector Sizes:     512 bytes logical, 4096 bytes physical
Device is:        In smartctl database [for details use: -P show]
ATA Version is:   ACS-2 (minor revision not indicated)
SATA Version is:  SATA 3.0, 6.0 Gb/s (current: 6.0 Gb/s)
Local Time is:    Mon Oct  3 13:16:17 2016 CEST
SMART support is: Available - device has SMART capability.
SMART support is: Enabled

=== START OF READ SMART DATA SECTION ===
SMART overall-health self-assessment test result: PASSED
...
SMART Attributes Data Structure revision number: 16
Vendor Specific SMART Attributes with Thresholds:
ID# ATTRIBUTE_NAME          FLAG     VALUE WORST THRESH TYPE      UPDATED  WHEN_
↪FAILED RAW_VALUE
  5 Reallocated_Sector_Ct   0x0033   200   200   140    Pre-fail  Always       -
↪ 0
```

Even if the SMART health status is *PASSED*, it doesn't mean the disk is OK. You should also check the "Reallo-

cated_Sector_Ct" attribute.

When the hard drive finds a read/write/verification error, it marks that sector as "reallocated" and transfers data to a special reserved area (spare area). This process is also known as remapping, and reallocated sectors are called "remaps". The raw value normally represents a count of the bad sectors that have been found and remapped. Thus, the higher the attribute value, the more sectors the drive has had to reallocate. This allows a drive with bad sectors to continue operation; however, a drive which has had any reallocations at all is significantly more likely to fail in the near future. While primarily used as a metric of the life expectancy of the drive, this number also affects performance. As the count of reallocated sectors increases, the read/write speed tends to become worse because the drive head is forced to seek to the reserved area whenever a remap is accessed. If sequential access speed is critical, the remapped sectors can be manually marked as bad blocks in the file system in order to prevent their use.

I recommend to replace a harddisk when the first bad sectors appears.

CHAPTER 16

DDRescue: data recovery from damaged disk

A bad sector is a sector on a computer's disk drive that is either inaccessible or unwriteable due to permanent damage, such as physical damage to the disk surface. Flash memory may also have "bad sectors" (even if technically there is no sector in flash memory) due to permanent damage like failed flash memory transistors.

Instead of working directly on the damaged disk, it's recommended to create a copy and to work on the clone. Two possibilities: create a disk image (a file) or overwrite a new/empty disk.

ddrescue can be found for Linux or macOS. If your computer is using another operating system, no problem, create a Linux LiveUSB! (See *Creating a live USB*)

16.1 ddrescue on Linux

drescue is available on all Linux distribution.

- CentOS: `yum install ddrescue`
- Debian/Ubuntu: `apt install ddrescue`
- Fedora: `dnf install ddrescue`

Use `lsblk` or `testdisk -lu` to identify all the disks.

16.2 ddrescue on macOS

To install ddrescue:

- Press Command+Space and type `Terminal` and press enter/return key.
- Run in Terminal app:

```
ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/
↪install)"
brew install ddrescue
```

Done! You can now use `ddrescue`. Use `diskutil list` to get information on all available disks and their partitioning.

16.3 DDRescue: disk to file image

It's the recommended method for forensic purpose. You need enough space to store the file: if you want to create a clone of a 1TB disk, you need at least 1TB free on a filesystem. Avoid FAT filesystem for the destination as they are limited to 4GB file.

In the following example, an image named sdb.dd will be created from the second disk /dev/sdb.

```
ddrescue /dev/sdb sdb.dd sdb.log
```

The log file `sdb.log` can be used to restart the recovery. It can take a few hours to several days to clone a disk with a lot of bad sectors.

16.4 DDRescue: disk to disk copy

The destination disk must be at least as big as the original one. Be careful, two disks of the same announced capacity from different vendors or sometimes from different models of the same vendor can differ slightly in size (a few 100 MB).

Ie. WD10EZRZ and WD10EZEX are two models sold by Western Digital as 1TB model, in fact the first one is 1,000,000 MB, the second one 1,000,204 MB.

Before beginning, disconnect all disks, usb device, cd/dvd reader/writer not needed: there is less chance to overwrite the wrong disk.

```
ddrescue /dev/sdb /dev/sdc sdb.log
```

The log file `sdb.log` can be used to restart the recovery.

16.5 ddrutility: restricting ddrescue to NTFS allocated data block

When a disk contains a lot of bad sectors, it may be safer to use ddrutility to limit the copy to allocated data block from an NTFS partition.

```
testdisk -lu /home/kmaster/data/data_for_testdisk/ntfs.dd
TestDisk 7.1-WIP, Data Recovery Utility, August 2016
Christophe GRENIER <grenier@cgsecurity.org>
http://www.cgsecurity.org
Please wait...
Disk /dev/sdb - 130 MB / 124 MiB - CHS 16 255 63 (RO)
Sector size:512


Disk /dev/sdb - 130 MB / 124 MiB - CHS 16 255 63 (RO)
     Partition                  Start        End    Size in sectors
 1 * HPFS - NTFS                32     255487     255456 [NTFS]
     NTFS, blocksize=512
```

In this example, the first NTFS partition begins at sector 32 and the sector size is 512 bytes.

```
ddru_ntfsbitmap /dev/sdb -i $((32 * 512)) sdb1_domain
ddrescue /dev/sdb sdb.dd sdb.log -m sdb1_domain
```

CHAPTER 17

Scripted run

TestDisk and PhotoRec can run automatically using their own built-in commands. A script file (such as .cmd or .bat batch files under MS-DOS/Windows, or some shell under Linux) may also be helpful.

17.1 Automating recovery using TestDisk

Syntax:

```
testdisk [/debug] [/log] [/logname file.log] /cmd [file.dd|file.e01|device] cmd
```

17.1.1 Some examples

```
testdisk /debug /log /cmd /dev/hda analyze,search
testdisk /debug /log /cmd partition.dd partition_none,geometry,H,32,analyze,list,
↪advanced,boot,rebuildbs,list
```

17.1.2 Device selection

Use the device name, e.g. */dev/hda*, */dev/hdb*, */dev/sda*.

For DOS version, use */dev/sda128* for first disk, */dev/sda129* for the second and so on... You may have to use single quote, i.e. `'c:\input dir\image.dd'`, if the path or file name contains spaces. For Encase files, you can use `file.e??` if you have less than 100 files, otherwise use `file.???`

17.1.3 Partition type selection

- partition_i386
- partition_gpt

- partition_humax
- partition_mac
- partition_none
- partition_sun
- partition_xbox
- ask_type: the user will be asked for the partition type (new in 6.9)

If no partition type is specified or asked, TestDisk will detect it automatically.

17.1.4 Main menu

- advanced
- analyze
- delete
- geometry
- mbr_code
- options
- list

17.1.5 Analyse menu

- backup: save to backup.log file the current partition structure
- number: select a partition found during Quick Search or Deeper Search
- list: list of the content of the selected partition (first one by default, new in 6.10)
- search: Deeper Search for more partitions
- noconfirm,write
- write

17.1.6 Advanced menu

- type
- addpart: add a partition entry (not written to disk)
- boot: for FAT12/FAT16, FAT32, exFAT and NTFS partition, go to the specific menu
- copy: backup the partition to the file image.dd (new in 6.9)
- list: list the content of the partition (new in 6.10)
- list,recursive: list the content of the whole partition (new in 6.10)
- list,recursive,fullpathname: list the content of the whole partition with the whole pathname (new in 6.11)
- list,filecopy: list and copy all the files (new in 7.1)
- superblock: search ext2/ext3 superblocks or go to HFS+ menu depending of the partition

- undelete: go in the undelete menu (FAT12/16/32, NTFS, exFAT, ext2)
- number: the partition number to select

Add partition

- PC Intel
  - c,XX starting cylinder
  - h,XX starting head
  - s,XX starting sector
  - C,XX ending cylinder
  - H,XX ending head
  - S,XX ending sector
  - T,XX type
- EFI GPT, Mac, XBoX
  - s,XX starting sector
  - s,XX ending sector
  - T,XX type
- Humax, Sun
  - c,XX starting cylinder
  - C,XX ending cylinder
  - T,XX type

FAT12/FAT16 boot menu

- dump
- list (new in 6.9)
- list,recursive: list the contents of the whole partition (new in 6.10)
- list,recursive,fullpathname: list the contents of the whole partition with the whole path name (new in 6.11)
- rebuildbs
- repairfat
- initroot

FAT32 boot menu

- dump
- list (new in 6.9)
- list,recursive: list the contents of the whole partition (new in 6.10)
- list,recursive,fullpathname: list the contents of the whole partition with the whole path name (new in 6.11)
- rebuildbs

- repairfat
- originalfat
- backupfat

FAT rebuild menu

- list
- list,recursive: list the contents of the whole partition (new in 6.10)
- dump
- noconfirm,write
- write

exFAT boot menu

- dump
- originalexFAT
- backupexFAT

NTFS boot menu

- rebuildbs
- dump
- list
- list,recursive: list the contents of the whole partition (new in 6.10)
- list,recursive,fullpathname: list the contents of the whole partition with the complete path name (new in 6.11)
- originalntfs
- backupntfs
- repairmft
- noconfirm,backupntfs
- noconfirm,repairmft

NTFS undelete menu

- allundelete (new in 7.1): list and recover all deleted files. WARNING: stores them in current local directory.

NTFS rebuild menu

- list
- list,recursive: list the contents of the whole partition (new in 6.10)
- list,recursive,fullpathname: list the contents of the whole partition with the complete path name (new in 6.11)

- dump
- noconfirm,write
- write

HFS+ superblock menu

- dump
- originalhfsp
- backuphfsp

17.1.7 Geometry menu

- C,number of cylinders
- H,number of heads
- S,number of sectors
- N,sector size

17.1.8 Options

- dump
- nodump
- align
- noalign
- expert
- noexpert

17.2 Automating recovery using PhotoRec

```
photorec [/debug] [/log] [/logname file.log][/d recup_dir] [/cmd <device> <command>]
```

General syntax:

- /debug: switch on debug mode
- /log: switch on logging (a log file named `photorec.log` will be created/appended to in the current working directory
- /logname file.log: log will be written to file.log instead of photorec.log
- /d recup_dir: specify directory to store the recovered files into. This should be on a device different from the one you are recovering from. PhotoRec will add a numeric extension to the path specified, starting with ".1" - and increase this number as long as a directory with this name already exists.
- /cmd: introduces the command section for scripted run
- <device>: the device (or image file) to recover from (Hint: use single-quote if the image file contains spaces)

- <command>: the command list (see below)

17.2.1 Some examples of data recovery using PhotoRec

Recover from the second IDE drives i386 partition the user selects

```
photorec /debug /log /cmd /dev/hdb partition_i386,select,search
```

Recover from the first IDE drives i386 partition #5, which is using ext2/ext3/ext4

```
photorec /debug /log /cmd /dev/hda partition_i386,options,mode_ext2,5,search
```

Recover from a given disk image file named "disk.dmp" which only has a single ext4 partition (or a part of it) Restore all file types known to PhotoRec to `/mnt/recover/disk`.

```
photorec /debug /log /d /mnt/recover/disk /cmd disk.dmp options,mode_ext2, \
  fileopt,everything,enable,search
```

The same without debug and log - but recover only **.gif* and **.jpg*

```
photorec /d /mnt/recover/disk /cmd disk.dmp options,mode_ext2,fileopt,everything,
↪disable, \
  jpg,enable,gif,enable,search
```

Recover jpg from the freespace of the first partition

```
photorec /cmd /dev/hda fileopt,everything,disable,jpg,enable,freespace,search
```

Recover all files from freespace from each partition as detected by testdisk

```
PARENT=`pwd`
DEVICE=/dev/sda
testdisk -l $DEVICE | tee testdisk.log | \
  egrep "[[:digit:]][[:space:]][P,E,L,D,*][[:space:]].+([[:space:]]+[[:digit:]]+){3}"␣
↪| \
  cut -f 2 -d\  |while read PARTITION
do
  mkdir $PARTITION && cd $PARTITION &&
  xterm -e photorec /log /debug /d ./ /cmd $DEVICE freespace,$PARTITION,search
  cd $PARENT
done
```

17.2.2 Command list

Below you find a list of available command options, grouped into categories. It is best to use them in the order they are mentioned here. These options must be separated by a comma. Partition type selection and options from the main menu can be used directly.

17.2.3 PhotoRec - Partition type selection

- partition_i386
- partition_gpt
- partition_humax

- partition_mac
- partition_none
- partition_sun
- partition_xbox
- ask_type: the user will be asked for the partition type

If no partition type is specified, it is auto-detected.

17.2.4 PhotoRec - Main menu

- fileopt: change file types to recover
- inter: PhotoRec usage becomes interactive
- options
- number: the partition number to select
- blocksize: force the block size - followed by the block size in bytes.
- geometry
- wholespace / freespace : files will be recovered from the whole partition or only from the free space (new in 6.10)
- ext2_group: carve the group whose number is following (new in 6.10)
- ext2_inode: carve the group whose following inode belongs to (new in 6.10)
- search: start the recovery

17.2.5 PhotoRec - fileopt menu

- everything,enable: use the values by default (may be different than the saved values, new in 6.9)
- everything,disable: empty the list of file formats to locate (new in 6.9)
- jpg,enable: will search for jpg
- jpg,disable: will not search for jpg

You can use the same syntax for all file formats.

17.2.6 PhotoRec - Options menu

To use anything from the options menu, you must specify the keyword “options” first.

- expert
- keep_corrupted_file_no (new in 6.10)
- keep_corrupted_file
- paranoid_no / paranoid / paranoid_bf (new in 6.10)
- lowmem
- mode_ext2

17.3 Windows UAC

If you run TestDisk and PhotoRec, Windows User Account Control will ask "Do you want the following program from an unknown publisher to make changed to this computer ?" (or something similar). As administrator rights are unneeded for disk images, you may want to avoid this UAC prompt with the `__COMPAT_LAYER` environment variable. Example:

```
set __COMPAT_LAYER=RunAsInvoker
photorec_win.exe /cmd image.dd search
```

CHAPTER 18

TestDisk and PhotoRec in various digital forensics test cases

To learn to use TestDisk and PhotoRec, various test cases are available to practice in safe conditions.

18.1 DFTT: Undelete files from a FAT16 filesystem

Download the small FAT filesystem image archive and extract all the files. This test image is a 6MB FAT16 file system with six deleted files and two deleted directories. The files range from single cluster files to multiple fragments.

To undelete all files manually,

- run *testdisk 6-fat-undel.dd*
- Choose *Proceed*.
- A non partitioned media is detected automatically, press Enter to confirm.
- Choose *Undelete*.

All files and directories are deleted, they are listed in red.

- Press 'a' to select all files.

The selected files and directories are now listed in green and prefixed by '*' or '<' for the current highlighted file.

- Press 'C' (uppercase) to copy all selected files and directories.
- Choose a destination to copy all the files: use the arrow keys (up, down, left, right) to navigate, you can also use the enter key to enter into a directory.
- Press 'C' when the destination is correct.

All files are copied.

- Press 'q' to quit
- Choose [Quit] until you have exited all menus

The usual filenames for a FAT filesystem are composed of 8 chars for the name and 3 for the extension. When a file is deleted, the first character of the filename is overwritten. TestDisk represents the lost char by a underscore _ (e.g. *_RAG1.DAT* instead of *FRAG1.DAT*) If a long filename (> 8 characters) is present, it will be use instead. A benefit is that the whole filename can be displayed (e.g. *System Volume Information*)

All files are recovered successfully except the 3 fragmented files. The size of these 3 files is correct but the content is wrong. When a file is deleted, the linked list formed by the cluster numbers used by the file are marked as free in the FAT tables. TestDisk assumes there is no fragmentation but it's not the case here.

18.2 DFTT: Undelete files from a NTFS filesystem

Download the small NTFS filesystem image archive and extract all the files. This test image is a 6MB NTFS file system with eight deleted files, two deleted directories, and a deleted alternate data stream. The files range from resident files, single cluster files, and multiple fragments. No data structures were modified in this process to thwart recovery. They were created in Windows XP, deleted in XP, and imaged in Linux.

To undelete all files manually,

- run *testdisk 7-ntfs-undel.dd*
- Choose *Proceed*.
- A non partitioned media is detected automatically, press Enter to confirm.
- Choose *Undelete*.

TestDisk lists all lost files successfully. The alternate data stream is listed as *./mult1.dat:ADS*, alternate streams are not listed in Windows Explorer, and their size is not included in the file's size. Malware has used alternate data streams to hide code. As a result, malware scanners and other special tools now check for alternate data streams. Forensics analyst should also search for them as they may be used to hide documents.

- Press 'C' (uppercase) to copy all selected files and directories.
- Choose a destination to copy all the files: use the arrow keys (up, down, left, right) to navigate, you can also use the enter key to enter into a directory.
- Press 'C' when the destination is correct.

All files are copied.

- Press 'q' to quit
- Choose [Quit] until you have exited all menus

18.3 DFRWS 2006 Forensics Challenge

DFRWS 2006 Forensics Challenge is a data carving challenge. It's possible to use PhotoRec to recover most files:

- run *photorec dfrws-2006-challenge.raw*
- Choose Proceed
- Go In Options menu
- Set "Paranoid : Yes (Brute force enabled)"
- Set "Keep corrupted files : Yes"
- Use "Quit" to return to the main menu
- Chose Search

- Confirm the filesystem type "[Other]"
- Use 'C' key to confirm the destination of the recovered files (current directory)
- Wait for the recovery to finish
- Quit

All these steps can also be automated in a single command:

```
photorec /log /d recup_dir /cmd dfrws-2006-challenge.raw options,paranoid_bf,keep_
↪corrupted_file,search
```

The file to analyze contained 32 files (not including the embedded files, such as pictures in Word documents or the files inside of ZIP files). The 32 files were used to create 22 different scenarios. Each scenario was designed to test a specific situation that might occur in a real file system.

Category 1 focused on HTML files with ASCII text:

- 1a) One HTML non-fragmented ✓
- 1b) One HTML fragmented with a JPEG in between
- 1c) One HTML fragmented with Unicode text in between
- 1d) Two HTML files that are intertwined

PhotoRec doesn't recover fragmented HTML correctly.

Category 2 focused on Microsoft Office documents:

- 2a) One Word file, non-fragmented ✓
- 2b) One Word file, fragmented with 3 fragments and random data in between
- 2c) One Excel file fragmented with random data in between
- 2d) One Word file fragmented with a JPEG in between ✓
- 2e) One Word file fragmented with text in between

Category 3 focused on JPEG files:

- 3a) One JPEG non-fragmented ✓
- 3b) One JPEG non-fragmented, larger than a typical default max file size ✓
- 3c) One JPEG non-fragmented, but sector before it has 0xffd8 in the first two bytes ✓
- 3d) One JPEG fragmented with text in between ✓
- 3e) One JPEG fragmented with a Word document in between ✓
- 3f) One JPEG fragmented with random data in between ✓
- 3g) One JPEG fragmented with a JPEG in between ✓
- 3h) Two JPEGs that are intertwined
- 3i) One JPEG non-fragmented that is REALLY big ✓
- 3j) One JPEG fragmented with singe sector in between that starts with 0xffd9 ✓

PhotoRec has good results in the JPEG category.

Category 4 focused on ZIP files:

- 4a) One ZIP file, non-fragmented ✓
- 4b) One ZIP file fragmented with text in between ✓

- 4c) One ZIP file fragmented with random data in between

	Filename	Location	Size	md5
	f0000000.html	0-8	4608	
1a	f0000009_Alice_in_Wonderland_[...].html	9-44	18147	✓
2c	b0002051.doc	2051-3867 4429-4435 4557-7963 ...	4428800	X
3a	f0003868.jpg	3868-4428	287186	✓
1d	f0004436_A_STUDY_IN_SCARLET_1.1.html	4436-4455	10240	X
1d	f0004456_1_Stave_1_Marley_s_Ghost.html	4456-4501	23544	X
1d	f0004502.html	4502-4556	27875	fragment
2d	f0007964_National_Park_Service.doc	7964-8284 9474-10031	450048	✓
2d	f0008285.jpg	8285-9473	608703	✓
3d	f0011619.jpg	11619-11822 11849-12017	190720	✓
3d	f0011823.txt	11823-11848	12828 (+2)	X
3b	f0012222.jpg	12222-26116	7113968	✓
1b	f0027496_Comedy_of_Errors_Entire_Play.html	27496-27606	56832	X
1b	f0027607.jpg	27607-27977	189534	✓
1b	f0027978.html	27978-28196	111693	fragment
1c	f0028244_Chapter_cxxxiv_-_THE_CHASE_[...].html	28244-28306 (X)	31850	X
1c	f0028307.html	28307-28344	18995	fragment
4a	f0028439_4n6rodeo3-fix_copy.zip	28439-28726	147150	✓
4b	f0028729_file1.zip	28729-29528 29896-31368	1163745	✓
4b	f0029529_The_Tempest_Entire_Play.html	29529-29895	187793 (-2)	X
3h	b0031475.jpg	31475-31532	29696	X
3h	b0031533.jpg	31533-31887	181760	X
2a	f0032837_Fact_Sheet_-_Permitted_and_[...].doc	32837-33397	287232	✓
2e	b0034288.doc	34288-34398 34413-36291 36641-36997	1201664	X
2e	f0034399.txt	34399-34412	6781	fragment
3c	f0036292.jpg	36292-36640	178659	✓
2b	b0036998.doc	36998-40637 41220-41238 41610 ...	3133440	X
3f	f0040638.jpg	40638-41219 41239-41609	487473	✓
3g	f0041611.jpg	41611-43433 44029-44200	1021085	✓
3g	f0043434.jpg	43434-44028	304413	✓
3e	f0045566.jpg	45566-45963 46104-46826	573499	✓
3e	f0045964_Statements_of_Financial_Condition.doc	45964-46103	71680	✓
3i	f0046910.jpg	46910-94836	24538540	✓
3j	f0094846.jpg	94846-95628 95630-96653	924877	✓

18.4 Forensics: write blockers

The content of a disk may be modified by simply connecting it to a computer:

- LVM driver will sync two RAID1-like volumes if they are out of sync
- Linux Raid and fake Raid will also resync the disks if they are out of sync
- Auto-mouting of the filesystem will modify the last-mount date and the mount count
- ext3 and ext4 will replay the journal if the filesystem is dirty.
- The NTFS file system may attempt to commit or rollback unfinished transactions, and/or change flags on the volume to mark it as "in use".

- The operating system will update the access time for any file accessed
- Windows may create hidden folders for the recycle bin or saved hardware configuration
- Virus infections or malware on the system used for analysis may attempt to infect the disk being inspected.
- Auto-indexation of the files may creates new files on the disk

Forensic disk controllers or hardware write-blockers are most commonly associated with the process of creating a disk image, or acquisition, during forensic analysis. Their use is to prevent inadvertent modification of evidence. Protecting an evidence drive from writes during investigation is also important to counter potential allegations that the contents of the drive were altered during the investigation. Of course, this can be alleged anyway, but in the absence of technology to protect a drive from writes, there is no way for such an allegation to be refuted.

A hardware write-blocker prevents modifications from the computer but it doesn't prevent a disk from modifying itself (i.e. SMART status updates in service area each time the device is powered-on.). It remains the best solution to prevent accidental modifications.

Without a hardware write blocker, it's still possible to reduce the risks of accidental modifications. Using a Linux computer without graphical interface and without auto-mounting *may* be considered a good enough solution.

Under Linux, *blockdev* or *hdparm* can be used to switch a disk to read-only:

```
blockdev --setro /dev/sdb
hdparm -r1 /dev/sdb
```

In practice, it doesn't work! TestDisk will open these devices in read-write.

Loopback device is a safer alternative:

```
losetup -r /dev/loop0 /dev/sdb
testdisk /dev/loop0
```

This way testdisk is forced to open the device in read-only.

Loopback can also be used to mount a filesystem in read-only: .. code-block:: none

losetup -r /dev/loop0 /dev/sdb partprobe /dev/loop0 mkdir /mnt/p1 mount -o ro /dev/loop0p1 /mnt/p1