DMDE 3.4

DM Disk Editor and Data Recovery Software

User Manual

Revision of the Manual 2017-08-31

Contents

About DMDE 3.4 Help 2

About DMDE 3

- Updates, Downloads, Restoration 3
- Acknowledgement 3
- DMDE Description 4
- DMDE Editions 5
- File Description 6
- Known Issues and Other Limitations 7

Installation and Run 9

- Operating Systems 9
- System Requirements 11
- Activation 13
- Settings 14
- Languages/Localizations 15

How to Use the Software 16

- Find and Open Volume 17
- Data Recovery 18
- Partition Management 24
- Full Scan 27
- Device Selection 29
- Device I/O Parameters 30
- Disk Images and Clones 35
- RAID Arrays 36
- File Panel 38
- Disk Editor 40
- Disk Editor Templates 41
- Cluster Map (File Allocation) 44

Menu 45

- Menu "Drive" 46
- Menu "Tools" 49
- Menu "Windows" 58
- Menu "Editor" 59
- Menu "Mode" 60
- Menu "Edit" 61

About DMDE 3.4 Help

The most recent documentation may be found on the software sites.

In the manual the following designations are used: elements of the software interface are written in ***bold italic***, keyboard keys and shortcuts are in square brackets [***F1***], text for typing and file names are in **`green`**.

Due to the lack of images reading the manual is recommended while running the software. **Context sensitive help** is also available (except in DOS) by pressing keys [***F1***], [***Shift+F1***], or [***Shift+F11***], and by ***Help*** buttons. **Help tips** (Windows GUI only) may be turned off and on through the menu command Help · Help Tips.

Understanding of some software settings may require special knowledge of file systems and disk structures and their discussion is beyond the scope of this guide. If the role of such parameters is not clear it is recommended to leave defaults (auto).

In order to open CHM files in Windows you should copy them to a local disk and unblock (right-click menu - Properties - Unblock).

Context sensitive help may be provided as CHM files (Windows only) or as a set of HTML files. HTML files are provided with Linux distributions in the directory **`man_en`** and other **`man`** directories in the subdirectory **`locals`**. To use HTML help files in Windows remove CHM files and place the directory **`man_en`** to the software working directory and other **`man`** directories to **`locals`**.

The default OS application is used for opening HTML files. In Linux the command **`xdg-open`** is used. You may use another command by setting the parameter **`shellopen=`**.

About DMDE

http://dmde.com/ http://softdm.com/

DMDE 3.2 (DM Disk Editor and Data Recovery Software)

DMDE Description DMDE Editions File description Issues and Limitations

Updates, Downloads, Restoration

The most recent versions are available on the software sites (see above). License owners may download and update licensed copies in the personal site section. To restore personal section entrance address please fill in the form on-line: http://dmde.com/request.html

Acknowledgement

Thanks for all your help with comments, bug reports, suggestions, feedback, support, publicity.

Special thanks to:
Igor (jsfhd), Alex (box2134), Konstantin Volkov, Leonid Arkadjev, Antech, Yatagan, 9285, okzo, Nirvanowiec, Migol21, gsm_virus, BIGOLSEN, Stepan Martinek, Przemyslaw Iwanowski, H.Mohamadi, Veniamin Khozyainov

DMDE Description

http://dmde.com/ http://softdm.com/

A tool to search, edit, and recover data on disks.

- A number of freeware features, additional paid features (DMDE Editions)
- Supported OS: Windows 98/..XP/..7/..10 (GUI and Console), Linux, DOS (console)
- Portable run without installation (Setup)
- Support for NTFS/NTFS5, FAT12/16, FAT32, exFAT, Ext2/Ext3/Ext4, HFS+/HFSX file systems
- Thorough FS and Raw scan, FS reconstruction, and data recovery in complex cases
- Simple partition manager for partitioning express diagnostic of partitions and to quickly find and restore accidentally removed or lost partitions
- Disk cloning and disk image creating, including I/O error handling, reverse copying, and other features
- Disk editor compatible with the most recent Windows versions which allows viewing, editing, and navigating through different disk structures (using built-in and custom templates)
- RAID constructor for virtual RAID reconstruction supporting levels RAID-0, RAID-1, RAID-4, RAID-5, RAID-6, delayed parity, custom striping, JBOD/spanned disks; a tool to automatically detect RAID striping
- NTFS tools to work bypassing NTFS driver (copy, delete file, create, repair directory)
- Support for national names, large disks, large files, large sectors, NTFS compression and encryption, various device I/O settings, and other features

DMDE Editions

Actual information is available on the software sites: Editions, Purchase.

To use all professional features you should download the licensed copy from your personal site section. In other cases you may just enter an actual license key into the basic version to remove data recovery limit. You may update, upgrade, restore your licensed copy in the personal section (restore your login).

Free Edition includes all features but single command recovers up to 4000 files from the current panel only. License key removes the limitation.

Professional Edition has additional features and grants extended rights to provide data recovery services:

- Portable use on different computers
- One-time activation on client computers (including remote use)
- Native 64-bit versions (access more than 3GB RAM, Windows and Linux only)
- Data recovery reports (include logs and file checksums)
- Using logs when copying a disk (resume copying, several passes)
- Customizable I/O handler script
- Recovery of NTFS alternate data streams
- DMA access under DOS (for ATA interface)

File Description

Executable

dmde.exe or **dmde** - the program

Localization

***.tbl** - code page tables
***.lng** - user interface language files

Documentation

eula*.txt - license agreement (EULA) files
versions.txt - information about some version changes
readme*.txt - readme files
dmde*.chm - help files in Windows CHM format
man*/* - help files in HTML format

Other files

dev9x.dll, **dev32.dll** - drivers to run under Windows 9x/ME
dmde*.ini - program settings files
template.txt - disk editor templates file
cwsdpmi*.exe - DOS Extender to run 32-bit DOS apps (Copyright © 2010 CW Sandmann, not a component of DMDE product)

Known Issues and Other Limitations

Data Recovery

- Data recovery destination is paritions/volumes supported by OS (in particular, DOS and Win9x/ME support only FAT as a destination)
- DOS, Win9x/ME: unicode symbols in file names outside the selected code page are replaced with "_" or transliterated according to the loaded translit tables
- DOS, Win9x/ME or FAT: unable to recover file greater than 4GB as a single file due to a FAT limitation
- File hard links are recovered as different files
- Symbolic links are not supported
- NTFS alternate streams can be recovered to NTFS volumes only
- NTFS directories alternate data streams can be recovered only after FS reconstruction
- NTFS encrypted file recovery up to 4GB
- NTFS permissions and owners are not supported
- Ext2/Ext3/Ext4 permissions and owners are not supported
- Win9x/ME: drives greater then 128GB are not handled correctly without special patches applied to Windows drivers
- Maximum path length supported: 259 characters for DOS, 4096 for Windows, Linux
- Maximum path length supported by DOS: 79 or 127 characters (for DOS-names) depending on DOS version.
- Subdirectory depth is up to 1024 (can be increased using **`maxrecoverdepth=`** parameter)
- If name substitution is used under DOS then name handling (such as auto renaming) is not supported

NTFS write support

- Journal is not supported
- SECURITY INFORMATION is not supported on creation (to access files and directories it is necessary to set permissions using OS features)
- OBJECTID is not supported
- Additional creation of DOS-compatible names is not supported
- Compressed files are not supported
- Alternate data streams are not supported
- To allocate/append a file a contiguous free space is required

Other

- the software may work within the available RAM. 32-bit versions may use up to 3GB of RAM only (up to ~10 million files/streams and 2 million directories); DOS extenders may provide even less memory
- up to 2K windows/panels/objects for GUI version, or width-limited for console

- up to 2K items in the list of Devices
- up to 16K custom data rotation items (**striping**) when constructing Custom RAID
- up to 2K visible items per section for Full Scan (`showvolumesnum=` parameter)
- up to 16 million MFT fragments of NTFS for Full Scan (`ntfsmaxmftruns=` parameter)
- maximum search depth for File Panel Search is up to 1K directories
- keyboard and mouse input is limited in Linux (keyboard shortcuts and national input may not be supported)

Installation and Run

System Requirements Activation Settings Languages

Attention! If in doubt about physical health of the disc (error messages, slow performance, etc.) it is recommended to address to specialists. Further work with the disk (including starting up) may worsen the problem or cause complete inability to recover data. If device is damaged. If device size is detected incorrectly.

Attention! Do not write anything to the source disk/partition where lost/removed/damaged data is located. Use another disk to load operating system, install or run the software, and to recover data otherwise the data may be erased irrevocably. It is highly recommended to use an environment where writing is turned off where possible (e.g. load a system from LiveCD/USB, bootable DOS disk, etc.).

To install/reinstall and run the program just extract the entire software package into a single directory (may be on removable media) and run `dmde.exe` or `dmde` depending on a version. You need Administrator / Superuser priveleges to access devices in Windows NT+, Linux. To run the software without Administrator rights add the parameter `notadmin=1` to the file `dmde.ini`. Updates are performed in the same way.

To uninstall the software just remove the extracted files and directories.

Operating Systems

Windows 98/ME

Attention! Without special patches Windows 9x/ME handles 128GB and larger drives incorrectly.
To avoid the problem reload computer in MS-DOS mode and use DMDE for DOS to access such drives via BIOS or ATA-interface or load another OS.

Windows 2K/XP

To use the program you should login as Administrator.
Attention! To work with 128GB and larger drives you need Windows 2K SP4 with LBA48 support manually switched on in the registers or Windows XP SP2 (LBA48 is switched on by default).

Windows Vista/7/higher

To run the program as Administrator right-click the program icon and use the corresponding command in the context menu or confirm UAC elevation request.
Attention! To work with drives larger than 2TB the most recent versions of controller drivers must be installed.

Linux

At first go to the Root Terminal or run Terminal emulator (Konsole, Gnome-terminal, etc.). Go to the software directory and type `./dmde`. You should run the software as a superuser to access devices. Please, refer to your operating system documentation for this information (e.g., run `sudo ./dmde` in Ubuntu, go to root with `su` command in Debian at first, etc.). You may also need to set execution permission for the file `dmde` (type `chmod 755 ./dmde`). To run 32-bit software versions on 64-bit operating systems you may have to install additional 32-bit shared libraries (e.g., run `sudo apt-get install libc6-i386` in Ubuntu to install `libc6-i386` package).

DOS

You may use DMDE on a regular partition supported by DOS or use DOS bootable disk. This may be helpful when there are problems using specific disk. SATA disks must be switched to IDE-compatible mode in BIOS settings for direct ATA-access. SCSI, USB, and other devices may be accessed via special DOS drivers or using BIOS functions if they are supported.

Some files for creating DOS boot disks are available on the software site in the section Additional Downloads.

Bootable Floppy

To create bootable floppy download and unpack **FDD image**. You may use the function Copy Sectors to write the image to a floppy disk. Use the image as a source file and fdd as a destination device. ISO images for bootable CD may also be created on the base of boot FDD images with the help of different software.

DOS Bootable Disk

Loading from the specific disk must be supported by BIOS (see BIOS boot menu). To create bootable USB/HDD you need a freshly formatted primary FAT16/FAT32 partition on the disk. Non-standard disks/partitions are not supported (non-MBR, 3TB+, etc.). You may probably need to zero the disk to properly create compatible primary FAT partition using standard OS means.

1. Download the **FreeDOS Package** and unpack files into the root of the FAT partition.
2. Open the disk as physical device in DMDE and use the command ***Write Boot Sectors...*** in the Partitions menu to make the device bootable (specify the location of unpacked files as a source of boot sectors). Apply changes and close DMDE.
3. Download DMDE for DOS and unpack into the FAT partition.

System Requirements

Common Requirements

- Drives: without significant hardware/firmware problems

  Attention! If there are doubts about device physical state (extraneous sounds, error messages, slow performance, the device is after hitting, etc.) it is recommended to address to specialists (data recovery from a damaged device)
- CPU: Intel compatible (i486 and higher)
- To save/use results, reports, and settings a valid partition is required which is supported by the operating system (in particular, FAT partition is required to work in DOS/Win9x/ME)

If Device Size Is Detected Incorrectly

This must be resolved before any further work. Some issues are quite simple - see guide from external source. Some issues are below:

- **Capacity Limitation Jumper** is misplaced (device size is limited to **32GB**)
- **HPA** is erroneously applyed. Device size usually becomes **1TB less** or similar (typical to some GYGABYTE motherboards). Try software: HDD Capacity Restore Tool (Windows), `hdparm` (Linux), HDAT2, MHDD, Victoria (DOS)
- No large drive support in **OS** - see below
- **Incorrect drivers** for SATA/USB controller (no correct **3TB+** support). Device is displayed tipically **2.2TB less**. Update drivers
- **USB enclosure** / another adapter has no correct **3TB+** support. Change adapter or attach device directly
- **Zero size**: the device is not accessible, you should address to specialists

Windows 2K+ (2K/XP/Vista/7/higher)

- Recommended: Windows XP + SP2 and higher
- 32-bit versions work in both 32-bit and 64-bit Windows
- Administrator privileges
- Drives: supported by OS
- For drives **larger than 2TB**, Windows Vista/7/higher is required with the most recent versions of controller drivers installed
- For drives larger than 128GB in Windows 2K/XP certain ServicePacks must be installed (Windows 2K SP4 and Windows XP SP2) and LBA48 support must be switched on in the system registry of Windows 2K. Windows 2K/XP does not support drives larger than 2TB with the exception of special XP-compatible USB-boxes

Windows 98/ME

- Drives: supported by OS

  Attention! Windows 9x/ME handles 128GB and larger drives incorrectly unless special patches are installed. You may reload computer in MS-DOS mode and use DOS version to access drives via BIOS or ATA-interface

DOS

- OS: MS-DOS 5.0+ compatible
- Drives: supported by BIOS or supported by DOS (using ASPI drivers) or supporting ATA-interface (IDE or SATA). SATA disks must be switched to IDE-compatible mode in BIOS settings for direct ATA-access
- DOS Extender (file CWSDPMI.EXE © 2000 CW Sandmann in the software directory is enough)
- At least 200 KB of low memory free
- At least 64 MB of extended memory
- For extended name duplicates handling during data recovery long name driver is required (e.g., DOSLFN.COM © Haftman software)
- If you wish to use a mouse then mouse driver for DOS is required

Linux

- OS: Latest Linux distributions with **`libc`** library
- 32-bit shared libraries to run 32-bit versions on 64-bit Linux (e.g. **`libc6-i386`** package)
- Drives: supported by OS
- Superuser rights
- Root terminal or terminal emulator (xterm, Konsole, Gnome-Terminal, etc.)
- utf-8 locale support

Activation

Attention! After purchasing you should obtain a licensed copy. To get all Professional features you should get the software in your personal section. In other cases you may just enter your license key into the existing version. Please adhere to license terms to avoid license blocking and revoking. You may get an updated key as well as information about blocking in your personal section (restore login).

You should activate the licensed copy when first run on a new/modified hardware or after entering a license key.

Online Activation

Just push the button ***Online Activation*** (Internet access is necessary, not available in DOS). Upon a successful connection to a server the activation code will be saved to dmde.ini.

Manual Activation

If Online activation is not available you may manually get and enter activation code.
1. Run the software on a computer where it will be used. Click ***Manual Activation*** and save the file **dmdeinst.dat** (or write out your **Prod.ID** and **Inst.ID**).
2. Open the activation site http://ac.dmde.com/ or http://ac.softdm.com/ on any computer. Upload the file **dmdeinst.dat** (or enter your **Prod.ID** and **Inst.ID**). An activation code will be generated for you.
3. Enter your activation code to complete Manual Activation.

Activation Types (Professional Edition)

1. Stationary Activation

Stationary Activation associates the software use with a specific computer: select ***Stationary*** activation (click or hit [***Space bar***]) and use Online or Manual Activation.

2. Portable Activation (Portable Use)

Portable Activation associates the software use with a removable USB flash drive that allows using on different computers without Internet access. Windows (2K and higher) & Linux only.

Initial Association with a Device

1. Select ***Portable*** activation and specify the removable device for association. The device must always report the same correct serial number (**12 or more digits and capital Latin letters**).
2. Use Online or Manual Activation to complete the association (will be stored to dmde.ini).

Further Use on Different Computers

Attach the device to a computer, run the software, and select the device if necessary.

3. One-time Activation

It is designed to one-time run the software on a client computer when it is not possible to use portable activation (e.g. for a remote use).

Download the basic software version (without Professional features) and enter special license key (Client Edition) and one-time password which are issued in the personal section (restore login) and then activate Online or Manually.

Settings

Settings are stored in the file `dmde.ini` in the software directory. See the file for the parameter descriptions.

Device I/O Parameters can be changed at runtime.

Languages/Localizations

Dialog Box "Select Language"

Apply Codepages. If the option is checked then ANSI and OEM code pages from the selected localization file are used.

Translit. Transliterate interface if there are problems displaying national symbols.

RTL dismiss (Linux only). Disable string reversal for RTL written languages if terminal emulator already supports it to avoid double reversal (e.g. Konsole).

Code Pages - Interface and Data Recovery

ANSI code page is used in DMDE interface under Windows and Linux. **OEM code page** is used in DMDE interface under DOS. Symbols outside the selected interface codepage are transliterated or substituted.

OEM code page is used also when read short names (8.3 format) from FAT volumes in all DMDE versions. DMDE for DOS uses **OEM code page** for file name recovery also. DMDE for Windows and for Linux uses Unicode (if it is supported by OS) during recovery regardless of the selected code page.

By default OS code pages are used (if applicable). Code pages from Ini-file (if defined) override them. Code pages from the selected lng-file (`*.lng`) override everything if ***Apply Codepages*** option is checked in the dialog box **"Select Language"**. Lng-file string `107` is for ANSI, lng-file string `108` is for OEM codepage.

INI-file settings

usecodepage= ANSI code page
oemcodepage= OEM code page
translitenable= (`Translit`)
viewtranslit= (`Translit`)
rtldismiss=

How to Use the Software

- Find and Open Volume
- Data Recovery
- Partition Management
- Full Scan
- Device Selection
- Device I/O Parameters
- Disk Images and Clones
- RAID Arrays
- File Panel
- Disk Editor
- Cluster Map (File Allocation)
- Menu Commands

Find and Open Volume

To browse, view, edit, and recover files and directories you should find and open a volume containing data. Use the following methods depending on the availability.

1. If the volume is fully available by its letter (`C:`, `D:`, ...), you may open it through the option ***Logical Disks*** (Volumes / DOS Services) in the dialog box Select Drive. This method is suitable to recover deleted files from a healthy volume. If you cannot open this way or files are being recovered incorrectly try the next methods.

2. Select a ***Physical Device*** containing the volume in the dialog box Select drive. The window "Partitions" will pop up. Select and open your volume if it is available there. If it is not available or data cannot be properly recovered then try the next method for the most complicated cases.

3. Run Full Scan and open one of the found volume variants. Read the section Full Scan for more efficient recovery in this case.

If the initial storage is a RAID you may Construct a RAID instead of opening a single physical device.

The context menu command ***Open volume parameters*** allows manually changing some parameters before opening the volume.

Also, the volume may be opened from the Editor Window when view the volume boot sector or boot sector copy in a Boot sector mode.

Data Recovery

Damaged Device Removed Files Encrypted Files Complex Scan

Attention! Make sure the device is not damaged beforehand (if device is damaged).

Attention! Do not write anything to the source disk/partition. Recover data to another disk only. It is highly recommended to recover data to another physical device. You may recover to another partition of the same device only if you are sure that the source and destination partitions don't overlap and the device has no physical problems. Do not load system or run software from the partition where lost data is located.

Attention! If files are recovered incorrectly or there are other problems make sure you are using the most recent software versions from a trusted source.

Data Recovery Steps

- Find and open a volume containing lost data;
- open/reopen the virtual directory ***[All Found + Reconstruction]*** to get more or less found files (see virtual FS Reconstruction);
- mark items for recovery and use the menu Tools · Recover to recover them to another disk.

With Free Edition you may recover up to 4000 files from the current panel per single command call (you should first open a directory with the necessary files on a panel).

If you cannot open/play files after recovery you should try different found volume variants - return to the Full Scan results or to Partitions and open the next found volume.

Dialog Box "Recover"

Button "Size"

Calculate the size of the data to recover. Global indicator of recovery process is working if the size is calculated. You should not calculate the size to avoid unnecessary device load on devices with bad sectors.

Button "List"

Create a list of selected files/sectors occupied by files.

"Deleted", "Found", files "by signatures"

Recover corresponding file categories - see File Panel.

Checkbox "Include NTFS altstreams" (Professional Edition only)

Recover NTFS alternate data streams (Alt. stream name and file name are separated by a colon, i.e. `FileName:StreamName`).

Create report (Professional Edition only)

Save recovery report to a file. List files and directories, log file I/O errors, calculate checksums CRC32, MD5, SHA1.

File filter

Define name masks (semicolon separated) to recover matching files only. Wildcards "**`*`**" and "**`?`**" may be used. Exclusion masks may be also defined by prepending a backslash "**`\`**". Masks at the beginning have a priority. E.g. the name **`abc.tmp`** matches **`a*;\*.tmp`** and does not match **`\*.tmp;a*`**.

File size limits

Filter files by size. Use **`0`** (zero) as a second value if you wish to set a lower limit only.

Additional Issues

Checkbox "Unicode names"

The option is set if there is Unicode support in OS, otherwise symbols outside the selected code page must be transliterated or substituted (see Locales). Unavailable in DOS.

Recover to a FAT formatted volume, checkbox "Split large files"

FAT volumes do not support files greater than 4GB (or sometimes 2GB). Larger files may be split during recovery upon a request if OS properly reports (which is not always the case) or if the option **Split large files** is used (preferable). Later you may merge file parts on another disk by using the system utility **`copy /b part1 + part2 + part3 result`**, for example. Pure DOS supports only FAT volumes.

File names in DOS, checkbox "Substitute names"

Without special drivers long files names are not supported under DOS. Option ***Substitute names*** allows name substitution during recovery. File **`LRENAME.BAT`** is being created in the destination folder for backward renaming. To restore original file names load OS Windows, go to the destination folder and execute the file **`LRENAME.BAT`**. An appropriate OEM code page must be selected when working in DOS in order to restore national symbols correctly. Unicode symbols outside the selected code page will be transliterated or substituted.

You may add the parameter **`substnamesutf8=1`** or use the option **utf-8** to create **`LRENAME.BAT`** in the utf-8 format fully supporting Unicode regardless the selected code page. Utf-8 format is supported in Windows 7 and higher.

Paths longer than 259 symbols under Windows NT+

In order to recover paths which are longer than 259 symbols you should manually prepend the destination directory name with a prefix **`\\?\`**, e.g.: **`\\?\D:\`**. Such long paths may be inaccessible using standard means such as Explorer ("My Computer").

Name Duplicate Handling

During recovery there may be name duplicates if there different versions of the same file/directory are found, if the destination file system doesn't distinguish between some names whilst the source FS distinguishes, if there are already files on the destination path before recovery.

If the OS reports the file/directory exists already you are prompted to choose how to handle the duplicate. You may rename/skip the object manually or choose the "auto rename" or "skip all" options to automatically handle all subsequent objects too. It is also possible to merge directories with duplicated names.

Additionally you are asked for a threshold (maximum number of duplicates for the same name). When the threshold is exceeded you are asked again for a desired action. Parameters `maxfilerenames=` and `maxdirrenames=` are used as default threshold numbers.

The destination file system designates duplicates. Duplicates are not handled if ***name substitution*** is used under DOS.

Event Handler... preset recovery events handling (the coincidence of names, I/O errors, and others).

NTFS Encrypted Recovery

NTFS encrypted files are recovered without decryption. You need the certificate from the source OS or backup to be installed on a destination OS to open encrypted files after recovery. Please refer to Microsoft documentation on exporting and importing certificates.

Encryption (EFS) is NTFS (+OS) feature, an encrypted file contains the encrypted data and encryption key which in its turn is encrypted with a certificate key. Since the feature may be unsupported in OS or for destination FS there are recovery options to workaround this.

Recover ***to separate streams***: recover data streams and keys into separate files (`.efs` and `.efk` extensions are assigned). Further processing of these files is a user task.
Recover ***to backup (portable format)***: recover data and key into a single backup file (`.efb`). The backup file may be stored on any FS, no OS support is required also. The backup file may later be restored into EFS encrypted file using the menu Tools - ***Restore NTFS encrypted from backup...*** (then both OS and FS must support EFS there).
Recover ***to NTFS encrypted file***: recover at once into an encrypted NTFS file. EFS support from both destination file system and operating system is necessary (i.e. NTFS, Windows).

Data Recovery from Damaged Devices

If you are in doubt about physical health of your device (error messages, slow responding, extraneous sounds, there was a hit, fall etc.) it is highly recommended to address to specialists having specialized equipment. Any further work with the disk (including even starting up) may worsen the problem and cause complete inability to recover data. So any further attempts are at your own risk. Anyway it is possible only if the software can access the device (contents are readable), and device size is detected correctly (info on incorrect device size).

Software Bads

Some errors may be due to external reasons: bad contacts, unexpected power loss or disconnection, etc. If a sector is not completely or incorrectly written then the device reports an error when reading such sector even there is no physical damage (software bad). The device is working normally excepting errors when reading several sectors. In such case you should fix the problem source (check if PCB contacts are oxidised, replace wires, etc.) and continue accessing the device safely if the doings don't become worsen (some software bads may still remain). Of course this does not apply to cases where a hit was, the device is responding slow, etc. Anyway it would be safer to make a copy of the entire device before continue recovering.

Device Cloning

It is highly recommended to first clone the damaged device and to work further with the copy.

During data recovery there may be multiple disk access retries: software usually scan the entire device, then again analize some structures on the disk, then read data again during actual recovery; different settings and software may also be tried. This causes reading the device again and again damaging the problem device more and more. Making a sector-wise copy (clone) eliminates this problem, the damaged device is read only once.

In DMDE you may use the tool Copy Sectors to make the device clone.

Special Environment and Settings

Normally OS may also work actively with a device damaging it more, interfere while working with it, as well may write unwantedly on the source disk (see also the section "Installation"). So you should choose a more suitable OS, prepare the device, configure the parameters.

It is better to use direct access in **DOS**. However there is a number of restrictions in DOS so the way is suitable primarily to create a clone when devices are visible in IDE compatible mode. **Linux** (and especially "Live" images) don't work actively with device partitions until you mount them so there are no special settings. Regarding **Windows** it is highly recommended to use special "Live" images, even more special environment like WinFE which don't work with device partitions until you mount them. It is also recommended to use IO SCSI mode as it is more fast and soft.

To completely prevent an OS from working with the device partitions you may temporaryli "switch off" a partition table (e.g. for MBR disk it is enough to change the signature **55 AA** in the lst two bytes of the disk starting sector). In DMDE you may use commands MBR Off / GPT Off in the Partitions window, then Apply, and restart the OS or reconnect the disk. Later you may similarly "turn on" the partition table on the disk or its copy if it is necessary.

If Cloning is Impossible

Sometimes the device state does not allow entire cloning. Full copy may take an unreasonable time but you need to recover only some important files. In such case you may try data recovery without performing a full scan.

DMDE allows opening volumes and finding some files without preliminary full scan. You may try opening the volume if it is visible in the window Partitions just after selecting the device. You may also try opening volumes as soon as they appear in the results of Full Scan without waiting its completion. Make sure to ***Save*** scan results to avoid future rescanning in case of some failure. Open the volume and recover the most important files first and check if they can be opened/played.

Removed Data Recovery

Removed recovery is generally performed the same way as other data recovery. However the possibility to recover removed files depends strongly on a file system, device type, and operating system activity.

Overwritten Is Not Recoverable

Really **overwritten data cannot be recovered** using any modern means. For modern HDD's there are no technologies to recover data even after single cycle of overwritting. But it depends on an OS activity and a fortuity how fast the data is overwritten after removal. Removed files on a system disk may be overwritten during several seconds on a modern OS. It may even worth to immediately unplag the PC power instead of regular shutting it down if you wish to prevent overwriting of accidentally removed data.

Another obstacle to recover removed data is a **trim technology used on SSD drives**. It is used to mark removed blocks as empty so these blocks are not readable after the removal. Theoretically it is possible to read memory banks directly using special equipment but there is constant rotation as well as another transformation of data on SSD disks which make such recovery job unreliable (however some random fragments of data still can be found).

Anyway it is not possible to surely predict if the data is already overwritten so you should try data recovery anyway. Regarding recovery of super valued data it also worths to contact specialists having special equipment and knowledge.

File System vs Signature Based (Raw) Recovery

FS based recovery allows recovering files with their original names and paths, whilst Raw (signature) based recovery provides only a bunch of files named randomly (see Full Scan). Of course the both may help only if the data is not overwritten.

FS based recovery possibility depends strongly on a file system when removed files are being recovered. Some file systems keep removed FS structures almost untouched after removal while others wipe FS structures immediately. File systems may also support journaling however it may help to recover only the most recent removed data. Raw recovery is a last resort when FS based recovery is not possible.

File System	FS Recovery Availability	Journal Support	
NTFS	high	+	Large files (>2GB) may be truncated to a zero size due to a specific behaviour of Windows NTFS driver
FAT, ExFAT	medium	-	Fragmented files are not recoverable correctly (only the first fragment position is being kept)
HFS+/X, Ext3/4	low	+	Only journal may help to recover some removed files. On Ext FS file names may remain but location info is wiped.

Previous File Versions and Copies

You should take into account there may be previous file versions or temporary copies or copies on other disks.

Windows restore points is enabled may keep previous file versions before removal. Right click the file/directory and check their properties. There may be previous versions available for recovery.

Partition Management

A tool to view, search, and open volumes, as well as to manage partitions basically on MBR and GPT-style disks. Other styles (e.g., hybrid, dynamic) are unsupported whilst partitions may be shown as found. Partition management includes removing and undeleting of removed and lost partitions, restoring boot sectors from their copies and other tools. Make sure to read the warning before use.

Marking the checkbox ***found*** causes express-search and displaying of lost and removed partitions. For more complex and deeper search use Full Scan. Volumes opened after the Full Scan are added to the list of found partitions and thus may be managed (e.g. may be undeleted if possible).

To prevent express search of partitions unmark the box ***Show partitions*** when open a device or construct a RAID.

Info mode checkboxes:

- ***found***: display found partitions
- ***tables***: display AMBR and GPT partition tables
- ***GiB***: display sizes in binary units
- ***detailed***: display multiple lines per partition in accordance with the source of information

(A) - bootable (active) partition status (**Partition** column).

Indicators indicate the presence of structures:

- **T** – partition table
- **E** – table entry
- **B** – volume boot sector
- **C** – boot sector/GPT copy
- **F** – basic FS structures (e.g., first MFT file record for NTFS)
- **f** – MFTMirr for NTFS
- **x** – structure is absent or damaged
-   (indicator is absent) – structure is not tested (however it may present). E.g. **F** indicator may be disclosed only after volume opening.

Found errors are colored in red. For incorrect/non-standard partitioning correct displaying is not guaranteed.

Dialog Box "Partitions" · Commands

Attention! Partitioning modification may cause impossibility to boot from the disk or inaccessibility of some partitions after applying. Never modify partitioning of your boot or system disk. Use alternative boot disk or connect your disk as second to another computer. Run the software from the device which won't be modified and save current partitioning to a file on a device which won't be modified. After applying restart the computer or reconnect pluggable device for changes to take effect. Cancel the **chkdsk** utility (by pressing any key) if it runs automatically at startup as it may cause non-recoverable damage of the data.

Use the ***Menu*** button to view the operations available for the selected partition.

Partitioning changes are pending until you push the button ***Apply*** or use menu command Apply Partitioning. See also Applying Changes for details.

Open Volume

Open volume to view and recover files.

Open Volume Parameters

View and manually edit volume parameters before actually open volume.

Show Volume Letters

Display volume letters assigned by OS.

Full Scan within the Partition

Full Scan

Create Image/Clone

Call the dialog box Copy Sectors to create partition image or clone the partition.

Insert the Partition (Undelete)

Insert lost or removed volume into the partition table. Invalid partitions may prevent insertion so they must be removed before insertion. Completely damaged partitioning may be resetted using the command Toggle MBR Signature On/Off applied to the topmost item. You should add partitions from the beginning to the end of the disk.

Remove the Partition

Remove the existing partition or a partition table. Some partitioning errors may be fixed by removing and inserting the partition.

Create RAW Partition

Create unformatted (RAW) partition within the unallocated space. If there are no partitioning errors it is highly recommended to use only standard system tools for creating and removing partitions.

Restore Boot Sector from the Copy

Replace damaged boot sector with a backup copy.

Toggle Bootable (Active) Partition Status

Set or reset active **(A)** status of a bootable partition.

Toggle MBR/Boot Signature On/Off (MBR On/Off)

Set or reset boot sector signature (**`0xAA55`**).

Reset GPT+MBR Signatures (GPT Off)

Reset boot sector signature (**`0xAA55`**) and signatures of GPT tables (**`EFI PART`**).

Set GPT+MBR Signatures (GPT On)

Set GPT and boot signatures. Available only on GPT disks after resetting.

Write Boot Sectors

Make DOS bootable disk.

Undo Action

Undo the last action.

Redo Action

Redo the last undone action.

Undo Partitioning Changes

Undo all partitioing changes.

Reset All Changes

Reset all unapplied changes.

Load from File

Load disk partitioning or rollback data from a file. See Changes for details.

Save Current Partitioning to File

Save current partitioning to a file for a backup purposes. Available only if the device is partitioned and there are no unapplied changes.

Apply Partitioning

Write changes to the disk. See Applying Changes for details.

Full Scan

Comprehensive search method allows you to find and virtually rebuild the directory structure of a damaged file system or to find lost files of known types by signatures if a file system can not be used.

If a file system is not seriously damaged then volumes may be opened before the scan completes. In the most complex cases different volume variants may be given. You should select and ***Open volume***, reconstruct file system and recover files to estimate the quality of a variant. If the selected volume variant does not contain necessary files or files are recovered incorrectly you should continue the search or try another variant. There are indicators which allow preliminary estimating the quality of a variant (details are below). If data was moved (due to partition move/split/merge/resize operation) then some files may be recovered correctly from one volume variant, and others from another one, despite that files are visible on both volume variants.

It is recommended to scan selected area of a physical device (including old and new location of the volume if it was moved) rather than scanning a logical disk. The software may read beyond the selected area if it is necessary.

Open volume	Open volume selected in the list.
Save...	Save search log to a file.
Load...	Load search log from a file (to continue previous search or to use previous search results).
Start/Stop	Start/stop scan. If scan parameters are not changed the scan will be continued.
Parameters	Select scan area, FS, parameters to gather data and calculate volumes. ***Jumping scan*** is used when it is necessary to faster find the main results over the entire disk without waiting the scan completes. Jumping scan is carried out selectively over the entire disk gradually covering the entire scan area. Volumes with the large number of files are being found first.
Menu	Besides, allows changing volume sorting order.

NTFS, exFAT, extFS, HFS

The columns ***%*** and ***Check*** allow estimating data recovery quality of a found volume variant (the percentage and number of successful reference matchings).

Min. Size is the minimum size containing all files in the best volume FS blocks.

FAT

The column ***Indicators*** shows the absence (damage) ("-") or presence ("+") of the corresponding FS structures (***B*** - boot sector/copy, ***F*** - FAT table/copy, ***R*** - root directory).

The column ***Check*** shows the number of found directories corresponding to the volume variant.

Raw File Search

File signatures are used to find lost files when recovery by file systems (above) is not possible. Additionally file signatures are used to improve and control the quality for NTFS, exFAT, HFS search. Normally there are no reasons to open Raw results individually when FS volumes above are found. Raw results are available under the directory ***$Raw*** in the file panel when volume is opened.

Raw recovery is possible only for files which have special signatures. Raw recovery does not work properly for fragmented files (only the first file fragment is recovered correctly). Raw recovery does not provide original file names and directory structure. Raw recovery does not always allow detecting original file sizes. If there is neither end file signature nor built-in size info for a given file type then file recovery continues until the next found file, so searching for all present file types may improve size detection.

Raw: File Signatures: select file types for search, add new file signatures.

File Signatures

File Signature is a fixed sequence of bytes at the beginning of files of the same type. E.g. if you open a PDF file in a hex editor you can see the first 4 bytes are the same for all pdf files:

```
0000:   25 50 44 46 2D 31 2E 33  0A ...       %PDF-1.3...
```

In DMDE you may then specify custom file ***signature*** as `%PDF` or as `\x25\x50\x44\x46` (non-ANSI characters must be specified as hex codes `\x..`). Question mark (`?`) designates any byte (if it may vary in the signature), `\m..` designates a bit mask for the previous byte (unset bits may vary, e.g. `\x34\m3F` is a 1-byte signature where 6-7th bits may vary).

If the signature position in a file is not fixed then minimum and maximum offsets must be specified. Entire signature must fit in the first 4096 bytes.

In simple cases, the ***Detect*** function automatically calculates a signature of a new type and fills in the required fields by analzing available valid files of this type (you must specify the location of sample files).

The ***Test*** button allows you to check the signatures on a specific file and find out whether files of this type are defined correctly.

Not all file types have special signatures and may be found using Raw search.

Multiple comma separated ***Extensions*** may be specified for ***FS calculation***.

Mark signature as ***reliable*** if it contains at least 4 different bytes at a fixed position (do not mark otherwise not to truncate other files if false positives happen).

File-container may contain other file types inside (mark in order not to truncate when other type files are met).

EOF Signature is an optional signature at the end of a file.

Device Selection

Select disk for further work.

If device size is detected incorrectly, refer to System Requirements issues.

Options

Physical Devices / Interfaces / Logical disks and DOS Services

Preferable access type depends on the problem (e.g. for data recovery see volume opening).

DOS Options

ATA Interface	Use direct device access (for IDE and SATA devices in IDE mode). The preferable way to access devices with bad sectors
BIOS Services	Use BIOS functions to access devices
DOS Services	Use DOS functions to access logical disks
DOS ASPI	Access to SCSI, USB, and other devices via ASPI drivers (if any)

Other

Disk Images: click or ***Enter*** the topmost list item to add new image for further work.

RAID option forwards to RAID constructor.

Show partitions option causes express searching and displaying partitions just after opening device.

Parameters button allows manually setting device I/O parameters.

Devices in Linux

If some specific devices are not listed you may use the option ***Disk Images*** and manually enter required device path, e.g. **`/dev/mmcblk0`** or **`/dev/mapper/truecrypt1`**.

Device I/O Parameters

Device I/O Dialog Box Device I/O Handler Script

Interface

A way to access disk/volume (may be set when selecting device).

OS Windows

IO FILE	use ReadFile/WriteFile functions to access device
IO SCSI	use SCSI driver (lower level, recommended to reduce the number of OS I/O attempts, to bypass OS write blocking)
IO ATA	use ATA driver
Overlapped	use overlapped I/O (does not affect I/O, not recommended)

DOS

- ***ATA Interface*** (direct access, recommended for devices with damages)

ATA Interface	for HDD/SSD
ATAPI Interface	for CD/DVD
use DMA	use faster DMA access for ATA (Professional Edition only)
use LBA48	to access more than 128 GiB
raw CD access	to access non-digital CD

- ***BIOS Services***

Old BIOS Service	old CHS access up to 8.4GB
Ext BIOS Service	LBA access

- ***DOS Services*** (for logical disks)

DOS Int 25/26	DOS functions
Win9x-FAT32 Calls	new DOS functions
MSCDEX Services	for CD/DVD

- ***DOS ASPI*** (access for SCSI, USB, and other devices if ASPI drivers are present)

I/O Errors

A way to handle I/O errors.

Skip I/O Errors: an option to automatically continue an operation after a number of ***auto retries*** when I/O errors happen. If turned off the user choice is required within I/O dialog box.

Do not wait if device is not ready: continue an operation even if a device is not ready (not recommended).

always: apply to future operations.
now: apply to current operation.

Auto retries number on CRC Error: a number of additional I/O retries when an I/O error happens. Do not use ***IO FILE*** interface to avoid OS Windows additional retries and delays.

> Zero value (`0`) stands for no additional retries, herewith the rest sectors of the block after the error will not be read to the **I/O buffer**.
>
> If the value is non-zero then sectors are being read one by one until an I/O error is encountered. The error sector is being read until the number of auto retries is reached or the sector is read succesfully.
>
> Increasing of retries number increases the percent of data successfully read but decreases device lifetime due to heavy load.

Auto retries number on Seek Error: additional I/O retries for errors other than "CRC".

> Input error is assumed to be a seek error if I/O buffer contents is not changed after I/O (or it is zero filled). Usually seek errors significantly slow down I/O and not recoverable on retries.

Fill errors (hex): an option to fill read buffer data with a 4-byte value when I/O error is reported, to reveal damaged data output.

Jump over sectors after error: skip the given number of sectors when an unrecoverable error is encountered.

> If there are areas of sequential bad sectors then skipping significantly reduces loading and further device damaging. Herewith the recovery quality reduces if there are only individual bad sectors.

Read back after jump: read skipped sectors backward after a jump until an I/O error is encountered.

> The option is unavailable if the number of sectors to skip is too big. The feature improves the percent of the data succesfully read.

Fill skips (hex): 4-byte value to fill unread data due to skips/jumps.

Additional Parameters

I/O buffer size in sectors: the maximum number of sectors transferred per I/O operation.

Timeout, msec.: time in milliseconds to wait for a device responce (for DOS ***ATA Interface***).

SCSI/ATA Timeout, s.: timeout for ***IO SCSI/ATA*** interfaces (Windows, may be ignored by OS).

Reopen device handle on error: check if the device is detected by OS after an error to prevent idling (Windows and Linux).

Retry I/O on device connection: automatically resume the operation when the device is reconnected (Windows GUI only).

ATA software reset if busy: perform ATA soft reset if device is busy after a ***timeout*** (for DOS ***ATA interface***).

ATA software reset timeout, ms.: time to wait for a device readiness after ATA soft reset.

Script...: set custom Device I/O Handler.

SMART: show SMART report.

Device I/O Dialog Box

Device I/O Parameters   Device I/O Handler Script

If an error occurs during device I/O then the dialog box appears where you can choose how to handle the error. **Attention!** Make sure to read the information on damaged device using.

> The following information is displayed: **[W]** indicates that error occurred when writing to device, **LUN** - disk number in the virtual RAID, sector numbers where error has occurred, attempt number, error number, and error description (given by OS in Windows). After a number of retries (defined by parameters) program waits for a user reaction (unless skipping errors mode in parameters is selected) and the button ***Retry*** becomes active. Dialog box appears before the block of data is processed.

Abort: abort current operation.

> Pressing Abort causes error sectors not to be copied to the destination (this behaviour may change in the future versions). Button is inactive if the operation is not abortable.

Retry: retry I/O.

> In some cases it is possible to successfully read a sector and continue without data distortion after a number of retries. Button is inactive while I/O is in the progress.

Ignore: continue operation ignoring the error.

> The current operation will be continued but most probably with data distortion. When reading file system structures this may cause loss of some files and directory structure damage.

Ignore all: continue operation ignoring all analogous errors.

> After a number of retries (defined by parameters) program does not wait for user choise but ignore the error and continue the current operation.

Reset: reopen device (Windows NT+ only).

ATA software reset if busy: perform ATA soft reset. Available only if device is opened via ***ATA interface*** under DOS.

Parameters: Device I/O settings

Confirm, Cancel: used for **`%CONFIRM%`** variable in the Device I/O Handler Script

Device I/O Handler Script

Device I/O Parameters Device I/O Dialog Box

The script can be used for I/O error logging and for extended processing of errors (include external program calls). The feature is available in the Professional Edition only.

Script can be loaded or modified via Device I/O Parameters (button ***Script***).

The file **`ondevhsc.txt`** contains a brief description of available commands and some script examples.

Possible script lines are of the form:
`IF CONDITION COMMAND`
or
`COMMAND`
or
`:LABEL`
where **`CONDITION`** is an inequality or equality (**`!=`**, **`>`**, **`<`**, **`>=`**, **`<=`**, **`=`**) of two quantities,
and quantity is either an integer constant (**`0`**, **`1`**, ...) or a variable (the list is below) or a simple mathematical expression (operators **`+`**, **`-`**, **`*`**, **`%`**, **`/`**, without parentheses, ignoring the mathematical order of operators), e.g. **`%ERROR%=0`**

Comments are preceded by two minuses (**`--`**)

Variables

Zero based

`%CONFIRM%`	=**1** if ***Confirm*** pressed, =**0** if ***Cancel*** pressed (confirmation dialog box appears)
`%DISKNUM%`	disk number in RAID array
`%TRYNUM%`	i/o retry number
`%LBA%`	first i/o sector number
`%SECNUM%`	number of i/o sectors
`%ERROR%`	i/o error
`%ATASTATUS%`	ATA Status Register value (defined if BSY bit is cleared, DOS ATA only)
`%ATAERROR%`	ATA Error Register (defined if ERR bit of **`%ATASTATUS%`** is set)
`%LINE%`	current line number in script
`%SERVICE%`	i/o service: **0**-ATA **1**-ATAPI **3**-BIOSINT13OLD **4**-BIOSINT13 **5**-DOSINT25 **6**-DOSINT73 **7**-DOSASPI **8**-DOSMSCDEX **9**-DOSFILE **11**-WINFILE **12**-WINSCSI **13**-WIN9XINT13 **14**-WINATA **20**-LINUXFILE **21**-LINUXSCSI
`%LASTRES%`	result of the previous command
`%LASTERR%`	error of the previous command

Commands

`SHOWDLG` - force device i/o dialog box popping up

`WAIT` - wait for user choice

`DELAY N` - delay for N msec

`EXECCMD CMDLINE` - execute external command CMDLINE using Command interpreter
(equals "cmd CMDLINE" in Windows)

`EXECCMDQ CMDLINE` - execute without creating new console window (quiet)

`EXEC "FILENAME" CMDLINE` - call external programm FILENAME with parameter CMDLINE

`EXECQ "FILENAME" CMDLINE` - call without creating new console window (quiet)

`MSDOS` - call Command processor (not supported in Linux)

`GOTO LABELNAME` - jump to label LABELNAME in script (to the string `:LABELNAME`)

`RETURN` - break script execution, handle error according to device I/O params

`RETRETRY` - break script as ***Retry*** button pressed

`RETIGNORE` - break script as ***Ignore*** button pressed

`RETABORT` - break script as ***Abort*** button pressed

`ADDLOG "FILENAME" LOGLINE` - write LOGLINE to file FILENAME
(string LOGLINE may contain variables)

`CANCELIO` - Call **CancelIO** (WinNT+ only)
(available if a device is opened with ***overlapped*** option)

`OVLRESLT N` - Check **Overlapped** result (N=1: wait; N=0: not wait) (WinNT+ only)
(must be used if a device is opened with ***overlapped*** option)

`RESETHANDLE` - Reopen disk handle (WinNT+ only)

`ATARESET` - ATA Soft Reset (DOS ATA only)

`ATARESETDET` - ATA Soft Reset followed by ATA Identify (DOS ATA only)

Format specifiers

To format output integer a variable name may be followed by a format specifier after a colon, e.g.
`%LBA:8x%` -- width: 8, hexadecimal

Sample

```
IF %ERROR%=0 RETURN -- return if no error on last i/o
IF %ERROR%=128 GOTO LABEL1
IF %ERROR%=5 GOTO LABEL1
RETURN

:LABEL1
IF %CONFIRM%=0 RETRETRY
-- retry if Cancel button is pressed
-- continue script execution if Confirm is pressed
EXECCMD /K ECHO error %ERROR% at LBA: %LBA% (%SECNUM%) try: %TRYNUM%. Type
EXIT to return.
IF %TRYNUM%<2 RETRETRY
DELAY 500
ADDLOG "C:\ERRORS.LOG" error %ERROR:x% at LBA: %LBA:10% (%SECNUM%) try:
%TRYNUM%
RETIGNORE
```

Disk Images and Clones

- Copy sectors (menu "Tools"): creating and writing images, disk cloning
- Disk images (Menu "Drive"): opening and using image as device

RAID Arrays

Dialog box "Construct RAID"

Use Menu - Drive - **Select Drive** - Construct RAID

A tool to construct a virtual RAID using individual disks (or partitions and images) when it is not possible by the means of the controller (hardware RAID) or the operating system (software RAID), as well as to detect RAID layout automatically when some RAID parameters are unknown.

If it is necessary to detach disks from the RAID controller and attach to a computer as individual disks.

Incorrect selection of a RAID type, data rotation (striping), or disk order causes incorrect data recovery (despite possible the correct directory structure recovery). Use autodetection to discover unknown parameters or disk order (***RAID Layout: Recognize***) - most probable RAID layouts are calculated statistically.

RAID types

RAID-0. Usually contains two disks on which information is written alternately by blocks (stripes). If one disk is absent only small files which fit into single block may be recovered.

RAID-1. Usually contains two disks, data is duplicated. Stripe size is not used. Supported in read-only mode. One healthy disk may be opened as a single device without loss of data.

RAID-4. Usually contains three disks. One disk is for parity. Data is alternated between the rest disks. It is possible to recover all data without any one of the disks. Use ***NULL device*** instead of the absent device.

RAID-5. Usually contains three disks (or more). Data is alternated between all disks herewith one block of parity per the rest blocks of data. There are four types of data rotation on a RAID-5. Some RAIDs (usually hardware) may have delayed parity. It is possible to recover all data without any one of the disks (which should be replace with ***NULL device***).

RAID-6. Usually contains four disks (or more). It is similar to **RAID-5** with exception that two blocks of parity (different types) are used. Only type of parity is supported so only one disk may be replaced with ***NULL device***. Supported in read-only mode.

RAID: *Custom rotation*. You may define your own **striping** (use ***Menu*** - ***Striping...***). Use integer numbers from `0` to specify data blocks according to their order on disks, `-1` to specify **XOR**-parity block, and `-2` to specify unused block.

E.g. for the diagram:					Specify striping as:			
Disk#0	Disk#1	Disk#2	Disk#3					
A1	A2	Ap	Aq	=>	0	1	-1	-2
B1	Bp	Bq	B2		2	-1	-2	3
Cp	Cq	C1	C2		-1	-2	4	5
Dq	D1	D2	Dp		-2	6	7	-1

JBOD/Spanned. Just a bound of disks which are used as one continuous disk. There is no data alternation, stripe size is not used. Data located on any disk will be lost if the disk is absent. JBOD is supported in read/write mode.

RAID parameters

Stripe Size: data rotation block size. Not used in **RAID-1**, **JBOD**
Delay: rotation delay (special **RAID-5** and **RAID-6** implementations)
Offset: data start offset on disk (usually used on software RAIDs)
Offset and Size: RAID logical unit offset and size on an individual disk
Null disk: used instead of absent or damaged disk
Striping...: manually specify striping for **Custom RAID**
Save/Load: save/load RAID configuration.

File Panel

File panel displays virtual directory structure and allows browsing through directories and selecting items to recover or to open them in the Disk Editor. You should open a volume to access the file panel.

Open virtual directory ***[All Found + Reconstruction]*** to virtually reconstruct file system and display all found and removed files and directories. The results of Full Scan are used for reconstruction.

To recover files marked as "found" or "deleted" (see icon description below) the corresponding options must be selected in the data recovery dialog box.

> Directory structure is virtual and may differ from the one that can be found directly on the disk or in the standard browser. Virtual names are enclosed in square brackets or begin with the sign **$** and are usually placed in the topmost level of the directory structure. Virtual directory **$Root** generally corresponds to a root directory of a volume in the standard browser.

Press [***Enter***] key or double click to open the directory, or open the file in the Disk Editor.

Press [***Ctrl+Enter***] to open the item in the Disk Editor.

Press [***Insert***] key or [***Space Bar***] or click the checkbox to toggle marking the item for recovery.

Press [***Shift+F9***]/[***Shift+F10***] or right click the item for a context menu.

File Attributes

`E/C/s`: encrypted/compressed/sparce (NTFS)
`D`: directory
`R/H/S/A`: hidden/read only/system/archive
`x/f`: "deleted"/"found"

File Icons

GUI	*Console*	*Icon Description*
📁	`[.]`	general directory containing files
📁	`[ ]`	directory not containing files
📁	`[x]`	directory containing deleted files
📁	`[f]`	directory containing "found" files
📁	`[x}`	directory containing deleted and "found" files
📁	`(x)`	deleted directory
📁	`(f)`	deleted directory containing "found" files
📁	`(.]`	deleted directory containing not deleted files (some kind of FS error)
📁	`(f]`	deleted directory containing not deleted and "found" files (FS error)
📁	`{f}`	"found" directory (FS block number in square brackets follows name)
📁	`{.}`	"found" directory containing general files
📁	`{x}`	"found" directory containing deleted files
📁	`{.)`	"found" directory containing general and deleted files
📁	`(x}`	"found" deleted directory
📁	`(r}`	"found" deleted directory containing deleted general files

📁	(f}	"found" deleted directory containing not deleted files
📁	(.}	"found" deleted directory containing not deleted and general files
📄	.	general/standard file (within the base file system)
📄	f	"found" file (in separately found FS fragment, FS fragment number is in square brackets, [J] indicates FS journal)
📄	x	deleted file
📄	xf	"found" deleted file
📄	+.	file found by signature (Raw)
📄	+.+	file found by signature, end file signature (EOF) is found
📄	+.-	file found by signature, end signature not found
📄	+.~	file found by signature, size detected, but probably partially overwritten
📄	+.~	file found by end signature
📁	[>]	directory which content is not completely known
📄	|.|	virtual directory containing several FS records belonging to a single file
📁	[.]	**$MetaData** - virtual directory containing special FS files
📁	[.]	**$NonameFiles**: contains files whose names and parents are lost
📁	{f}	**$Raw**: contains files found by signatures
📄	.	**FileName:StreamName** - NTFS alternate data stream

Disk Editor

Disk editor allows you to view and edit disk objects such as sectors, files and other file system and disk structures in hexadecimal and formatted (structured) modes including the use of Custom Templates to parse data.

Select Object:

- use the menu Editor to select disk or file system object or to jump to a specified position
- use File panel to select a file or directory (right click for a context menu or press [***Ctrl+Enter***])

Select View Mode (Template): menu Mode.

Enable Edit Mode: menu Edit - ***Edit Mode*** [***Ctrl+E***].
NTFS file editing is supported in RAW mode only.

Apply Changes: menu ***Drive*** - Apply Changes [***Ctrl+W***].

Use the menu Tools for additional functions:

- Copy sectors... to write the selection or the entire object to a file.
- Fill sectors... to fill sectors with a pattern.
- Search string to search string in the selected object.
- Search spec. sector to find some data structure on a disk.

> Yellow text color in the Disk Editor indicates modified but not written data.
>
> Grayed text color indicates unused data (unused part of the cluster, unused entry, etc.).
>
> Green data color indicates I/O errors while reading corresponding sector.
>
> Red text color indicates logical errors in the data.

Disk Editor Templates

Apart built-in templates, Disk Editor supports custom templates to view and edit different disk structures. Conditions, "goto" operators, simple arithmetics may be used to parse complex disk structures, such as MFT records.

By default templates are loaded from the file **`template.txt`**. To use different files ini-parameter **`editortemplates=`** may be used, wildcards are allowed (e.g., **`editortemplates=template*`**).

See files **`template.txt`** and **`template.tx_`** for template samples.

Template File Structure

Each template begins with a name in square brackets **`[Template Name]`** followed by the template parameters and instructions (one per line).

Template Parameters

`flow:0` - display single record at a time. **`flow:1`** - display records in a flow.
`big-endian:1` - use big-endian byte order.
`h:Header` - display static **`Header`** at the top.

Constants

Constants are specified as decimals and hexadecimals (with **`0x`** prefix).

Variables

`$RECSIZE` - size of the record.
`$OFFSET` - relative offset applied to **data blocks**.
`$1` ... **`$64`** - custom variables (64-bit signed integers).

Data Block

Data block is usually a single byte/word/dword at a fixed position but it also may be a range or series of bytes/bits which are processed as a single variable. Data block is specified in braces **`{...}`**.

`{X,Z}` defines range of **`Z`** bytes starting at offset **`X`**,
`{X:Y,Z}` defines range of **`Z`** bits starting at offset **`X`** byte and **`Y`** bits,
where **`X`**, **`Y`**, and **`Z`** may be any variables or constants,
several ranges may be separated by a semicolon, e.g. **`{0x00,4;$1:$2,4}`**.

Data Block Formats

Format defines how data block is represented and edited (e.g. as integer / char / string, etc.).

The following formats are supported:
`%u` - unsigned integer (up to 32bit)
`%D` - signed integer (32bit)
`%I` - signed integer (64bit)
`%X` - hexadecimal (up to 32bit)
`%IX` - hexadecimal (up to 64bit)

`%c` - ANSI character (8bit)
`C` - array of ANSI characters
`U` - array of Unicode characters (UTF-16)
`T` - array of text characters (depending on encoding table)
`UNIXDATE` - Unix date (seconds since epox)
`FILETIME` - Windows file time (nanoseconds since 1601)
`F:ABCD..` - Flags (where `A` is displayed if bit 0 is set, and `B` if bit 0 is clear, etc.)

Output

Output instruction defines the position on the screen and format for a data block or variable or simply outputs text.

`{...},x:X,w:W,c:C,f:Format` outputs data block `{...}` at column `X` with the max. width `W`.
`x:X,w:W,c:C,f:Text` outputs `Text` at column `X` with the maximum width `W` in color `C`.
Color `c:C,` is optional (`0` - default, `1` - title, `8` - red, `10` - grayed).
`=` (equal sign) specifies the end of line (line feed).

Operators and Expressions

Expression is a combination of variables, constants, data blocks, and operators (`~`, `NOT`; `*`, `/`, `%`; `+`, `-`; `<<`, `>>`; `<`, `<=`, `>`, `>=`; `=`, `!=`; `&`; `^`; `|`; `AND`; `OR`), e.g.

```
$2+{0x08:$1,5}
```

Assignment operator `:=` is used to copy the result of the expression to the variable, e.g.

```
$1:=$2+{0x08:$1,5}
$2:=$OFFSET & 8
```

Transitions, Cycles, and Transitions

Conditions

```
IF Expression1
    ... (instructions for performing if Expression1 result is true/non-zero)
ELSEIF Expression2
    ... (else instructions for performing if Expression2 result is true/non-zero, optional)
ELSE
    ... (else instructions for performing in other cases, optional)
ENDIF
```

Cycles

```
WHILE Expression3
    ... (instructions for performing while Expression3 result is true/non-zero)
    go to the start of cycle:
    CONTINUE
    break loop:
    BREAK
ENDWHILE
```

Transitions

Line **`LABEL:N`** defines a label, and instruction **`GOTO:N`** is a jump to line **`LABEL:N`**, where **`N`** is any constant. Inaccurate use of **`GOTO`** may cause infinite loop.

Switches and Hot Links

Switches can be used e.g. to hide/show some lines by clicking or pressing [***Spacebar***]. E.g., instruction **`$1:=TOGGLE:N,x:X`** outputs button **`[+]`** (**`[-]`**) at column **`X`**, where **`N`** is a unique switch number (variables and constants are allowed, the best way is to use the next data offset); variable **`$1`** gets value **`0`** or **`1`** depending on the switch state.

A hot link allow jumping to an object offset or record having value **`variable`** by double clicking (or hitting [***Enter***] key) a line where the instruction **`$GOTOREC:=variable`** or **`$GOTOOFS:=variable`** is placed.

Additional Template Sections

Record size calculation: instructions between lines **`CALCSIZESTART`** and **`CALCSIZEEND`**. It is used when record size may be greater than the sector size and may depend on disk data. Variable **`$RECSIZE`** may be assigned in this section only.

Data preprocessing: instructions between lines **`LOADSTART`** and **`LOADEND`**. It is used e.g. for MFT USN processing (restoration of last two bytes of each sector). Data blocks may be assigned in this section: e.g., use **`{U,Y}:={X,Y}`** to copy **`Y`** bytes at offset **`X`** to offset **`U`**.

Data postprocessing: instructions between lines **`FLUSHSTART`** and **`FLUSHEND`**. It is used for reverse operation when writing modified data to disk (data blocks may be assigned).

Cluster Map (File Allocation)

Volume cluster map allows finding the file located at a given cluster/sector. Supported for NTFS and FAT/exFAT.

To use cluster map first open volume and update cluster map (menu ***Tools***).

To go to a specific location use Menu:

- ***Editor-Volume Cluster*** [***Alt+C***] to go to a specific volume cluster
- ***Editor-Volume Sector*** [***Alt+S***] to go to a specific volume sector
- ***Editor-Physical Sector*** [***Alt+P***] to go to a specific device sector

To identify the file located at the specific location use Menu:

- ***Editor-Cluster Map*** [***Ctrl+M***] to open cluster map at a selected location.
- ***Editor-File Data*** [***Ctrl+F***] to open file at a selected location.

Status line of the cluster map has the following format:
[current volume cluster/total clusters number] "file (stream) name" vcn: cluster number per file

Menu

- *Drive*
  - *Select Drive...*
  - *Open Image...*
  - *Construct RAID...*
  - *Partitions...*
  - *Lock Volumes...*
  - *Device I/O Parameters...*
  - *Load Rollback Data/Dump from File...*
  - *Dump Changes to File...*
  - *Undo Changes*
  - *Redo Changes*
  - *Reset All Changes*
  - *Apply Changes*
- *Tools*
  - *Recover...*
  - *Search in found (by name)...*
  - *Reconstruct File System...*
  - *Restore NTFS encrypted from backup...*
  - *Full Scan...*
  - *Cluster Map*
  - *Update Cluster Map*
  - *Clear Directory Tree*
  - *NTFS Tools*
  - *Copy Sectors...*
  - *Fill Sectors...*
  - *Search...*
- *Windows*
- *Editor*
- *Mode*
- *Edit*

Menu "Drive"

- *Select Drive...*
- *Open Image...*
- *Construct RAID...*
- *Partitions...*
- *Lock Volumes...*
- *Device I/O Parameters...*
- *Load Rollback Data/Dump from File...*
- *Dump Changes to File...*
- *Undo Changes*
- *Redo Changes*
- *Reset All Changes*
- *Apply Changes*

Manage Changes

All changes made in Disk Editor and Partition Manager remains virtual until you explicitly apply them to the disk.

Load Rollback Data/Dump from File...

You may load rollback data (if saved when applying previous changes) and thus undo the last changes. Also you may use the command to load changes which were dumped to a file instead of applying to the disk.

Dump Changes to File...

You may dump changes to a file instead of applying them directly to the disk. Later you may reopen the disk and load all unapplied changes from the file.

Undo Changes

Group undoing the last changes. To undo individual actions use the corresponding commands of Disk Editor (menu edit) or Partition Manager.

Redo Changes

Group redoing the last changes.

Reset All Changes

Discard all unapplied changes.

Apply Changes [*Ctrl+W*]

Write all changes to the disk. For changes to take effect in operating system you may probably have to restart computer. Under Windows you may not be allowed writing to some disk areas used by volumes - see Volume locking for details. Some antiviruses may also block direct writing to disk. Under Windows you may also try interface option ***IO SCSI*** to circumvent some restrictions.

Volume Locking

Volume locking is used under Windows NT+ for direct write access to prevent simultaneous OS access to a volume at the same time.

DMDE tries locking volumes automatically when it is required. However, locking may fail if the volume is being used by system or other applications. In this case, DMDE requests to retry or dismount not locked volumes. Close all applications which may use the volume and retry.

Retry Retry volume locking

Force Dismount the volume. **Warning!** All used volume handles will be lost, e.g. unsaved data in opened files will be lost

Ignore Try direct writing without locking the volume. **Warning!** File system may be damaged or direct write access may be denied by operating system

Cancel Cancel the operation

Warning! You should not lock or dismount the volume from where DMDE software runs from. You may not lock or dismount system volume.

DMDE unlocks volumes when closing the device currently used.

Additionally you may manually manage volume locking using menu ***Drive*** · ***Lock Volumes for Write***.

Menu "Tools"

- *Recover...*
- *Search in found (by name)...*
- *Reconstruct File System...*
- *Restore NTFS encrypted from backup...*
- *Full Scan...*
- *Cluster Map*
- *Update Cluster Map*
- *Clear Directory Tree*
- *NTFS Tools*
- *Copy Sectors...*
- *Fill Sectors...*
- *Search...*

Volume Commands

Search in found (by name)

List files matching the specified pattern. See Data Recovery for options. Search is performed over already found volume files. Reconstruct file system to list deleted and other found files too.

Update Cluster Map

Update file allocation information.

Clear Directory Tree

Reset reconstructed virtual directory tree. View directory structure without the results of Full Scan and file system reconstruction.

Virtual File System Reconstruction

Open virtual directory ***[All Found + Reconstruction]*** in the File Panel to virtually reconstruct directory structure and prepare found and removed files and directories for recovery. You may reopen the directory again to get more or less results in the reconstruction depending on your need.

The results of the Full Scan are used (if the scan was performed). If files are recovered incorrectly you should try found different volumes - return to the results of the Full Scan or to the window Partitions and open the next volume.

NTFS, extFS, HFS

Pure FS Reconstruction

The option provides results within the original FS only. For slightly damaged FS these are enough and the most accurate results.

Default Reconstruction, Less (--) or More (++) Results

Use the results of Full Scan. You may decrease and increase the number of results with buttons -- and ++. The color indicator schematically shows the quality and the number of the selected results (the more additional results the worse their quality).

Advanced

Advanced parameters are primarily for specialists to make some corrections for the reconstruction if a file system is significantly damaged.

MFT Numbers (NTFS)

Filter files by MFT numbers for a partial reconstruction (may be useful if there is not enough RAM for the entire reconstruction).

prevent wrong merging

Prevent possible wrong merging of directory branches by considering time stamps.

process INDX'es (NTFS)

Better (but slower) directory structure reconstruction using information from INDX'es on top of omitted MFT info (***max*** - force INDX processing).

Shifted (NTFS)

Include shifted MFT records (Full Scan must be performed before).

Extra Found (NTFS)

Include extra found files into the reconstruction (may contain more garbage but may help if some file is not recoverable other ways).

Tech. Records (NTFS)

Include special MFT records into the reconstruction for analysis.

MFT / FS Fragments

Manually select FS fragments to use in the reconstruction. Fragments are grouped by the relevance to the opened volume (calculated statistically):
the best, correct - most relevant and relevant,
related - may be relevant to the opened volume variants (cf. Full Scan),
unknown - relevant volume is not detected,

extraneous **-** relevant to a different volume,
small (extra found) **-** too small for statistics,
disabled **-** incompatible due to different FS parameters.

FAT/exFAT

Rescan current file system

Rescan the entire volume if the results within the current FS format only are needed and Full Scan is not fulfilled.

Default reconstruction

Use the results of the Full Scan.

FAT Tables Options ("*Parameters*")

FAT tables contain cluster chains to assemble fragmented files and directories. By testing tables the software selects which table to use or not to use both. However you may try different options to achieve better results of the reconstruction and data recovery in the whole or for specific files.

You may choose which table copy to use (**FAT1** or **FAT2**) or not to use both. The option ***check (do not use bad sectors)*** may allow avoiding some wrong chains if there is some damage in tables. ExFAT has only one FAT table which cannot be properly tested (values are correctly defined for fragmented file chains only).

NTFS Tools

Warning! The use of NTFS Tools on incompatible NTFS volume may damage the file system. Do not use the tools if the program does not recognize file system properly or you are not sure if it is.
Never execute the commands on the system volume (where operating system files are located) otherwise it will be damaged.
Volume locking is performed when writing to volumes. After executing the commands you should check the volume with the system utility **chkdsk**.
It is not recommended to create file names with special symbols.

Open NTFS volume and enter the necessary directory in the file panel before using tools.

Copy File

Copy, overwrite, or append file to the opened NTFS volume bypassing NTFS system driver.

Open destination directory in the file panel to copy file to.
Call the command from menu.
Select the source file.
Enter destination file name.
Choose either to overwrite or append file if the destination file exists.

A continuous free space on the volume is required for writing the file.

Create Directory

Create a directory on the opened NTFS volume bypassing NTFS system driver.

Directory is being created in the current folder of the file panel.

Remove File/Empty Directory

Remove selected file or empty directory from opened NTFS volume bypassing NTFS system driver.

Repair Directory INDX-records

Repair NTFS directory currently opened in the file panel.

Copy Sectors

A tool for creating disk images, clones, writing images back to disk, and copying different disk objects. Write is performed directly to the destination, the operation may not be undone. For changes to take effect in OS you may probably have to reload computer. See Volume locking also.

Source and Destination

Button	***Result***
Device	Select Device, Logical disk (volume), Image as source/destination sectors
Partition	Select Partition as source/destination sectors (use ***Device*** button to select the disk before),
	To select the entire current device/disk/RAID double click the topmost item in the partition list
File	Use file as source bytes / a destination (alternatively ***Device*** · ***Open Image*** to use file as sectors)
Editor	Use the object opened in the Disk Editor window as source bytes
Selected	Use selected block of the Disk Editor object as source bytes

Source fields ***Start Sector/Byte***, ***End Sector/Byte***, ***Number of Sectors/Size in Bytes*** are filled in automatically on source selection by buttons. If a ***File*** or ***Editor*** is used as a source then fields are specified **in bytes**. You may manually specify a range inside the source object to copy.

Destination field ***Start Sector*** is also filled in automatically on destination selection by the button ***Device*** or ***Partition***. ***Start Sector*** is an offset relative to the beginning of the destination object. ***Start Sector*** of the **source** is copied to the ***Start Sector*** of the **destination** regardless the direction of copying.

If RAID is constructed then it may be used as a source/destination.

Using the ***Device*** button you may select a file (disk image) as a source or a destination. Then you may specify parameters available for a disk only.

If ***Editor*** or ***Selected*** is used as a source then some logical errors (such as invalid cluster or sector numbers) are silently ignored.

Parameters

Log file contains a report on the copied sectors and errors. If a ***partition*** is selected as a source then sector numbers are relative to the beginning of the disk (not the partition).

The option "***use log to continue copying***" allows skipping already copied sectors (the option is available in Professional Edition only). In particular if copy settings and **parameters** are not changed then copying will continue from the last position when started. The option may also be used for multi-pass copying with decreasing jumps on bad sectors, preliminary copying of important areas, copying in different directions.

Changing some device I/O parameters (decreasing ***jump over sectors***, increasing ***auto retries number*** from `0`, decreasing ***I/O buffer size*** if there are `0` retries) on a new pass will cause copying sectors previously skipped due to jumps after bad sectors.

The option "***Retry copying bad sectors from log***" is to copy bad sectors again when all skipped sectors are already copied (Professional Edition only). On each pass bad sectors are being copied from the beginning, continuation is not possible.

Flush log when number of sectors copied: flush buffer log to disk (update log-file) regularly when number of sectors is copied.

The button "***Parameters***" allows setting device I/O parameters to handle errors (such as retries, jumps through bads, etc.)

Reverse copy: copy backward (from the end to the beginning) to create an image of a disk with problems.

Double thread: perform read and write operations in parallel threads to increase the speed. Do not use the option for copying disks with errors.

The field ***Continue from sector/byte*** specifies the position to continue copying. The position is relative to the source beginning. Copying in ***reverse*** direction starts previos to the specified position and goes backward (e.g., if set to **`10`** then sector **9** is copied first and copying goes to the beginning).

The option ***Lock the source for copy*** is for locking the volumes located over the source area to prevent modification of the source during copying (Windows only).

Fill Sectors

Write a hex or a file pattern to a range of device sectors. Write is performed directly to the disk, the operation may not be undone. Pattern is being written multiple times, covering all specified sectors.

The maximum size of the hex pattern is 8 bytes. For a larger pattern use a file.

Sector alignment. If a successive pattern copy does not fit in the current sector write it partially and write the next copy from the beggining of the next sector. Otherwise write the pattern copies successively ignoring sector boundaries.

Report (Professional Edition only). Write a report to a text file. To create a formatted report put a pattern file **`fillreport.html`** to the software directory and add the parameter **`fillsecreport=fillreport.html`** to the ini-file. You may download the default **`fillreport.html`** file by right-clicking the link.

Special Sector Search

Search for special disk or file system structure. Search is performed over disk sectors starting next to the current sector.

Search String in Object...

Search for a string in the Editor object.

Options to search blocks filled with a pattern

x times in a row: search for the repeated pattern;
rotate match: search alternately for matching and nonmatching data (search for the end of repetitions after the pattern is found).

Search Again

Continue searching the string over the current object or specific sector over the disk starting from the current position in the Disk Editor.

Menu "Windows"

- Directory Tree
- File Panel
- Editor
- Search Panel (Search in Found)

Menu "Editor"

Select object to open in Disk Editor.

Partition Table	Open all disk sectors. Go to the MBR sector (the first disk sector).
Boot Sector	Open all volume sectors. Go to the volume boot sector (the first sector of the volume opened).
Boot Sector Copy	Open all volume sectors. Go to the volume boot sector copy.
Root Directory	Open volume root directory.
FAT/MFT	Open FAT1 (main cluster table) for FAT volume, MFT for NTFS.
FAT Copy/MFT Mirror	Open FAT2 (cluster table copy) for FAT, MFTMirr for NTFS.
MFT Record...	Open MFT. Go to the specified MFT file.
Volume Cluster...	Open all volume sectors. Go to the specified volume cluster.
Volume Sector...	Open all volume sectors. Go to the specified volume sector.
Physical Sectors...	Open specified drive sectors. Go to the specified drive sector.
File Data	Open file located at the current cluster.
Directory Entry	Open directory containing current file. Go to the file entry.

FAT Cluster/MFT Record	For FAT: open FAT1;
	go to the FAT1 cluster corresponding the current volume cluster.
	For NTFS: open MFT;
	go to the MFT record corresponding the current file or volume cluster.

Cluster Map	Open Cluster Map. Go to the map cluster corresponding the current volume cluster.
Goto Object Offset...	Go to the specified position of the opened object.

Except for commands ***Partition Table*** and ***Physical Sectors*** you should open a volume at first.

Menu "Mode"

Toggle Disk Editor view mode.

Custom Templates	Select custom template
Hexadecimal, Text	Use menu item ***Encoding*** to change view code table.
FAT Directory	Press [***Enter***] key to open file data or directory in the Editor window (if FAT volume is opened).
FAT12/FAT16/FAT32	FAT cluster table. Press [***Enter***] key to jump to the corresponding volume cluster.
Partition Table MBR/GPT	Use the command again or press [***F6***] to switch between MBR / GPT / Copy view. Press [***Enter***] key to jump to the corresponding sector.
FAT/FAT32/NTFS Boot Record	Volume boot sector. Use the command again or press [***F7***] to select the next file system. Press [***Enter***] to open volume.
NTFS Directory	NTFS directory INDX-record. Press [***Enter***] to open file or directory in the Editor window (if NTFS volume is opened).
MFT Record	Press [***Enter***] key to open selected attribute or directory in the Editor window. Press [***Space bar***] to view attribute details.
NTFS File RAW Sectors	Switch between decoded file data and RAW sector data (as on drive).
Encoding...	Select code table to translate text in ***Hexadecimal***/***Text*** mode.

Menu "Edit"

Commands to use in Disk Editor.

Fill Zero Block	Fill the selected block with zeroes
Copy	Copy the selected block to the internal buffer
Paste	Paste internal buffer data at the current cursor postition
Paste File...	Paste file at the current cursor postition
Select Block	Toggle selection mode. Or hold down [***Shift***] and use arrow keys or hold down left mouse button
Select All	Select the entire object
Export to Text File	Export the view of a current or selected records to a file as text
Reread block	Read data from the disk again
Hide/Show Changes...	Switch between displaying the initial data and the modified data
Undo	Undo the last edit (in the mode hex/text - all changes within the sector)
Redo	Redo editing
Edit Mode	Toggle edit/read only mode