DMDE 4.4

DM Disk Editor and Data Recovery Software

User Manual

Contents

Revision of the Manual 2025-10-09

About DMDE 4.4 Help

The most recent documentation can 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. **Hints** (GUI interface only) can be turned off and on through the menu ***Help***.

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 can 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 / macOS the commands **xdg-open** / **open** are used. You may use another command by setting the parameter **shellopen=**.

About DMDE

www.dmde.com   www.softdm.com

DMDE 4.4 (DM Disk Editor and Data Recovery Software)
Copyright © 2005-2025 Dmitry Sidorov

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

DMDE Features   DMDE Editions   File Description   Issues and Limitations   Setup

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, Juri Micheli, rk, Rogerio Copetti, Zhou Xiaodong, SALiH DRD

DMDE Features

www.dmde.com www.softdm.com

- Portable launch without installation,
  support for Windows, macOS, Linux, DOS (installation and run)
- Support for NTFS, FAT12/16, FAT32, exFAT, ReFS, Ext2/Ext3/Ext4, btrfs, HFS+/HFSX, APFS
- Thorough FS and Raw scan, FS reconstruction for data recovery in complex cases
- Simple partition manager for express search, diagnostics, and restoration of partitions
- Disk cloning and disk image creating, including I/O error handling, reverse copying, and other features
- 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; automatic calculation of RAID configurations
- Cluster map to investigate file allocation
- 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
- NTFS tools to work bypassing NTFS driver (copy, delete file, create, repair directory)
- Support for various device I/O interfaces and settings to work with damaged devices, disk images, NTFS compression and encryption, national names, large disks, large files, large sectors, and other features

DMDE Editions

You can get the most up-to-date information online:
dmde.com/buy.html
softdm.com/buy.html
DMDE Websites and Description

To unlock data recovery limits and other features of your license, you should activate a copy of the software (see Activation). You can manage your license in your DMDE account (restore your login).

Free Edition includes all basic features but a single recovery operation recovers up to 4000 files in the current panel only (you should first open a subdirectory in the current panel and then recover files in the panel). In paid licenses there is no this restriction, and recovery of nested directories is allowed.

Professional Edition provides additional features:

- rights to provide data recovery services
- portable use on different computers
- one-time activation on client computers (including remote use)
- file list export to HTML format (view sample online)
- data recovery reports (include logs and file checksums)
- read support for E01 disk image files
- using logs when copying a disk (resume copying, multiple passes)
- customizable I/O handler script
- recovery of NTFS alternate data streams
- DMA access in DOS (for ATA interface)

TOC · About DMDE

DMDE - File Description

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
changelog.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
listtmpl.htm - HTML file list export template (for DMDE Professional)
cwsdpmi*.exe - DOS Extender to run 32-bit DOS apps (Copyright © 2010 CW Sandmann, not a part of DMDE software)

Known Issues and Other Limitations

Data Recovery

- Data recovery can only be performed to another disk (partition), the target disk must be supported by the OS (e.g., in DOS/Win9x/ME you can recover data to a FAT formatted disk only)
- DOS, Win9x/ME: unicode symbols in file names outside the selected code page are replaced with underscores ("_") or transliterated according to the loaded translit tables
- DOS, Win9x/ME or FAT: files larger than 4GB can only be recovered with splitting in parts due to the FAT limitation
- File hard links are recovered as different files
- NTFS alternate data streams for directories can be recovered only after FS reconstruction
- NTFS encrypted file maximum size: 4GB
- NTFS permissions and owners are not supported
- Maximum name length supported: 255 characters
- Symbolic links, special attributes, other FS-specific options 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, macOS
- 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 may access even less memory
- up to 134 million directories and 1.7 billion items per directory can be handled in 64-bit versions (within available RAM blocks)
- no additional support for virtual disk image files in DOS
- NTFS alternate data streams for a file are displayed only in a single directory (no matter the file may be hard linked to multiple directories)
- 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

Attention! If a device has physical problems (input-output errors, slow performance, etc.) **it is recommended to contact specialists** - see Working with Failing Drives.

If the device is not accessible in macOS see disk access in macOS.

If the device has incorrect size see Wrong Drive Size.

Attention! Do not write anything to the disk containing lost data. **Recover data to a different disk, as well load operating system, run the software from a different disk,** otherwise the data may be erased irrevocably.

OS Environment for Data Recovery

It is highly recommended to work in a system with minimal disk access (especially in write mode). You can use bootable LiveCD/LiveUSB disks based on **Linux** or special environment like WinFE based on **Windows**, where disk access is minimized. Or create a DOS bootable disk to clone disks available in IDE mode.

For the most compliance with such environment it is recommended to turn off the "fast boot" option and turn on the "USB legacy" mode in your BIOS/UEFI. To boot from the Live disk you may also need to turn off Secure Boot and UEFI mode (switch to Legacy BIOS mode).

Installation

To install/reinstall and run the program just extract the entire software package into a single directory (it can be on a removable media) and run **dmde.exe** or **dmde**. Updates are performed in the same way.

You need Administrator / Superuser priveleges to access devices in Windows NT+, Linux, macOS. To run the software without Administrator priveleges add the parameter **notadmin=1** to the file **dmde.ini**.

For more information, see the "Operating Systems" below (Windows / Linux / MacOS / DOS).

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

Operating Systems

Windows Vista/7/higher

If SmartScreen prompts, click "More Info" and confirm that you want to run the software.

To run the program as an administrator, right-click the application icon, use the appropriate menu command, and confirm the UAC elevation request.

Attention! To work with drives larger than 2TB the most recent versions of controller drivers must be installed.

Windows 2K/XP

To use the program you should login as an Administrator.

Attention! For drives with a capacity of 128 GB or more, ServicePack updates must be installed (Windows 2K SP4 or Windows XP SP2) and LBA48 support must be enabed in the system registry (manually for Windows 2K, and it is enabled by default in Windows XP SP2). Windows 2K/XP does not support drives larger than 2TB with the exception of XP-compatible USB-drives/boxes.

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.

Linux

For information on how to run software as a **root** user, see your operating system documentation. Typically, this involves going to Terminal or running a Terminal emulator (Konsole, Gnome-terminal, etc.), going to the program's folder, and running **sudo ./dmde**, or first logging in as root with **su** and then running **/.dmde**.

You may also need to set execution permission for the file **dmde** (**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).

MacOS

Since modern versions of macOS try to prevent direct disk access and are less capable of handling faulty disks, it is recommended to use other operating systems. If you don't have another PC or can't connect your drive to it, you can try creating and booting Ubuntu LiveUSB by following the instruction.

If you cannot access disk in macOS you can try to **unmount the disk** using the system's **Disk Utility** to gain access. Another option is to create the disk image to a file using the Disk Utility and then work with the disk image.

If macOS blocks the application you can use **Control-Click / Right-Click Menu - Open** or go to System Preferences, choose Security & Privacy and click "Open anyway" for the blocked application: Safely open apps on your Mac.

On macOS the program can be launched directly from the **.dmg** package. However, in this case, it will not be able to save its settings in its folder. The same can happen with an extracted application, since macOS launches applications obtained from the Internet from a randomized directory. In this case, the program will offer to save the settings and keys in the default application support directory of the current user.

To enable portable launch (to store settings directly in the program folder), you can copy and paste all application files from the **.dmg** package, including the **dmde.ini** file, to a separate folder. If you first extracted the files (rather then copy-pasting then), you also need to remove the quarantine attribute by using the Terminal command
xattr -r -d com.apple.quarantine /<path to application>/dmde.app

It is also possible to run the console version of the software as root user from a terminal, see Linux instructions above.

DOS

If there are problems with video ouput you may need to change the value of the parameter **biosoutput=0** in the file **dmde.ini** into: **biosoutput=1**

If an error occurs when starting the software, there may be a problem with an extended memory manager. Try using the old memory manager by running the batch file **dmde_dpo.bat** however less memory will be available.

You may use DMDE on a regular partition supported by DOS or use DOS bootable disk. This can 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 can 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.).

0. For the most compatibility optionally first zero out the disk, then initialize disk as MBR and create and format a primary FAT partition using standard OS means (Windows Disk management).

1. Download the FreeDOS Package (the site section "Additional downloads") and extract all package files directly to the root directory 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 root directory on the disk as a source of boot sectors upon the subsequent request). Apply changes and close DMDE.

3. Download DMDE for DOS and unpack into the FAT partition.

4. Use BIOS boot menu / BIOS setup to boot from the specified disk. On modern UEFI systems you may need to enable BIOS legacy mode, turn off secure boot option, switch USB and other devices to legacy modes.

Wrong Drive Size

Do not work with a disk which size is detected incorrectly. For HDD drives or incorrectly connected devices it is sometimes possible to resolve size related issues by yourself - see the guide from an external source.

- **Capacity limit jumper** is not set correctly (device size is limited to **32GB**)
- **HPA** is applied incorrectly. The device size is usually **reduced by 1TB** or so (typical for some GIGABYTE motherboards). Try software: HDD Capacity Restore Tool (Windows), **hdparm** (Linux), HDAT2, MHDD, Victoria

(DOS)
- The **OS** does not support large disks - see below
- **Incorrect drivers** for SATA/USB controller (no correct support for **3TB+**). Device size is tipically detected as **2.2TB less**. Update drivers
- The **USB enclosure** / other adapter does not correctly support **3TB+**. Replace the adapter or attach the device directly
- The disk is detected as **zero size** or the SD card or USB drive is detected as **tiny size**: the device is not accessible, it is recommended to contact specialists

System Requirements

<u>Windows</u>   <u>Linux</u>   <u>MacOS</u>   <u>DOS</u>

Common Requirements

- Drives: without significant hardware and firmware problems
  Attention! If a device has physical problems (disk is not accessible, there are input-ouput errors, extraneous sounds, slow performance, there was a hard drive hit, etc.) it is recommended to address to specialists - see the section <u>Working with Failing Drives</u>
- See <u>Installation and Run</u> for recommended OS environment and disk access issues
- 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)
- See also <u>Issues and Limitations</u>

Windows

- Recommended OS version: Windows Vista and higher (see <u>Installation and Run</u>)
- 32-bit versions work in both 32-bit and 64-bit Windows
- Administrator privileges (except for Windows 98/ME)
- Drives: supported by OS

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 emulator (xterm, Konsole, Gnome-Terminal, etc.) for console version, or `gtk2.0` library for GUI version
- utf-8 locale support

MacOS

- OS: macOS 10.12+
- Administrator privileges

- Drives: supported by OS, see Installation and Run

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 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

Activation

To unlock all features of your license, you should activate a copy of the software the first time you run it on new/modified hardware or after entering your license key.

Please comply with the license terms, otherwise activation may be rejected. You can obtain an updated license key and manage your license in your DMDE account (restore login).

Online Activation

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

Manual Activation

If Online activation is not available you can manually obtain and enter the activation code.

1. Launch the software on the computer where it will be used. Select **Manual Activation** and write out your **Prod.ID** and **Inst.ID** codes (or save them to the file `dmdeinst.dat`).

2. Open the activation site ac.dmde.com or ac.softdm.com on any device and enter your **Prod.ID** and **Inst.ID** codes (or upload the file `dmdeinst.dat`). An activation code will be generated for you.

3. Return to the launched software and enter your activation code to complete the 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. Available in Windows (2K and higher), Linux, and macOS.

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**). It is recommended to use flash drives from **Kingston** and **SanDisk** as they are among the most compatible.
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, launch the software, and select the device if necessary.

3. One-time Activation

It is designed to run the software once on a client computer when portable activation is not possible (for example, for remote use).

For one-time activation, obtain your Client Key and one-time passwords in your DMDE account (restore login) to enter them on the client's remote computer

Settings

Settings are stored in the file **dmde.ini** in the software directory (for macOS possible exception see the section Installation and Run). 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

Unicode interface and Unicode file names are used if it is supported by OS. Otherwise **ANSI code page** is used in OS Windows 98/ME and **OEM code page** is used in DOS, whilest symbols outside the selected codepages are transliterated or substituted.

OEM code page is used also for national short name reading (8.3 format) on FAT volumes.

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.

> The context menu command **Open volume parameters** allows manually changing some parameters before opening the volume.
>
> Also, the volume can be opened from the Editor Window when view the volume boot sector or boot sector copy in a Boot sector mode.

Work with Special Storages

If the data storage is a RAID array / striped volume you should first Construct a RAID instead of opening a single physical device.

If the volume is contained inside a virtual machine / encrypted disk image or partition you should first mount / decrypt the image (or its copy, for safety) using the corresponding (virtualization) software and then work with the mounted / decrypted disk. However you may also open a flat image and VHD/VHDX image directly using the option **Disk Images** of the Device Selection box.

Data Recovery

[Damaged Devices](#)   [Removed Files](#)   [Errors in 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. 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 ([Installation and Run](#)).

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 / Virtual FS]** 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 opened panel per single command call (you should first open a directory with the necessary files on a panel and then select recovery from the 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"

File Categories

Select file categories for recovery - see [File Panel icons](#) for categories details. Checkboxes for excluded files will be unchecked unless the option "**keep selection marks for excluded items**" is used.

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" · File List

Create a list of selected files with their attributes.

Button "List" · HTML File List

Create a list of selected files with their attributes in HTML format (Professional Edition only).

Button "List" · Sector List

Obtain a list of sectors occupied by selected files (e.g. it can be used by a different software to create a partial disk clone). It is possible to report fragments boundaries in **sectors** or in **bytes**, and to specify **file paths** in the list.

The option **Sector ranges** allows listing only those files fragments which are located in the specified sector ranges. E.g. this allows obtaining a list of damaged files located in bad sectors when you recover from the image file (or from a healthy cloned disk) which does not report source bad sectors itself. If you created an image/clone using the function Copy sectors with the log file enabled then you can export a list of bad and skipped sectors from the log by using a command in **Copy sectors** Menu.

Data recovery restrictions are applied on a number of files and subdirectory processing if Free Edition is used. Alternatively, you can use the Cluster Map report without restrictions, in this case the lists will also be sorted by the position of the fragments on the disk.

Checkbox "Include NTFS altstreams" (**Professional Edition** only)

Recover NTFS alternate data streams. Alt. streams are displayed in the File Panel after FS reconstruction, an alt. stream name is separated by a colon.

Filters

Filtering by a name mask, size, modification date, ID is supported. Checkboxes for excluded files will be unchecked unless the option "**keep selection marks for excluded items**" is used. Use **0** (zero) as a second value for the size/ID if you need

not to limit the higher value for the size/ID.

Name masks are semicolon separated. Wildcards "*****" (any set of chars) and "**?**" (any char) are supported. Exclusion masks may also be 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***.

If the option **Process only directories with files** is used than directories will be recovered which contain files matching filters.

Create Report (**Professional Edition** only)

Save recovery report to a file (file name and other parameters are requested on recovery process start). Files and directories are listed, I/O errors for files are reported, checksums CRC-32, MD5, SHA-1, SHA-256 are calculated for a forensic report.

Additional Options

Checkbox "Unicode names"

The option is marked or used by default if an OS supports Unicode names. Without Unicode support symbols outside the selected code page will 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 can be split during recovery upon a request if OS is properly reporting this (which is not always the case) or if the option **Split large files** is used (recommended for FAT). Later you may merge file parts on another disk by using the system utility **copy /b**, for example. Pure DOS supports only FAT volumes.

File names in DOS, checkbox "Substitute names"

Without special drivers long files names are not supported in DOS. With the option **Substitute names** names are substituted during recovery. File **LRENAME.BAT** is being created in the destination directory for backward renaming. To restore original file names load OS Windows, open the destination directory and run the file **LRENAME.BAT**.

With the option **utf-8** the file **LRENAME.BAT** will be created in the utf-8 format fully supporting Unicode. Utf-8 format is supported in Windows 7 and higher. Without the **utf-8** option the OEM code page will be used and Unicode symbols outside the

selected code page will be transliterated or substituted.

Paths longer than 259 symbols (Windows NT and higher)

Support for extra long paths option (or manual prepending the prefix **\\?** when specifying the destination directory, e.g.: **\\?\D:**) allows recovering paths longer than 259 symbols. Such long paths may be inaccessible using standard OS means such as Explorer ("My Computer").

Duplicate Names, Errors in Files, and Other Events Handling

During recovery there may be **name duplicates** if there are different versions of the same file/directory (or different links), or directories are merged wrongly, or the destination file system doesn't distinguish between some names (case insensitivity), or there are already files in the destination directory before recovery. You need to specify what to do in such case.

The options ***rename, auto rename, skip/all, merge/all*** (for directories) are available. 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. Duplicates are not handled if ***name substitution*** is used in DOS.

During recovery there may be I/O errors and FS errors which obviously result in **errors in affected files**. You can also specify how to handle such files:
Skip to skip a file with such errors (the skipped file will remain marked after recovery),
Ignore to continue recovery with errors,
Ignore and Hide to continue recovery and hide a file (set the attribute "Hidden" in Windows / DOS, and set zero **0** permissions in Linux / macOS).
Move to $Bad to move the file to the directory **$Bad** within the destination directory.

You can specify handlers during the recovery or before recovery using the option ***Event Handler.*** As well you may change the destination, split the file if there is no more space on the destination.

NTFS Encrypted Recovery

DMDE recovers NTFS encrypted files without decryption. An encrypted file contains the encrypted data and encryption key which in its turn is encrypted with a certificate key. You need the certificate from the source OS or its backup to be installed on a destination OS to open encrypted files after recovery. Please refer to Microsoft documentation on exporting and importing certificates.

To store encrypted files NTFS encryption (EFS) must be supported by both OS and the destination file system (i.e. Windows and NTFS are required). In DMDE there are workaround options for EFS recovery when run on a different platform.

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 file data and key into a backup file (`.efb`). Backup files may later be restored into EFS files using the menu Tools - **Restore EFS from Backup...** (only when run on a EFS supporting platform).

Recover **to NTFS encrypted file**: recover at once into an encrypted NTFS file when run on a EFS supporting platform.

Working with Failing Drives

If you are in doubt about physical health of your device (input-output errors, slow responding, extraneous sounds, there was a hit, fall etc.) it is highly recommended to contact 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. In any case, this is only possible if the program has access to the disk (the contents can be read) and the disk size is determined correctly (see Installation and run).

Software Bads

Some errors may arise due to external reasons: bad contacts, power loss, 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 (a software bad). The device is working normally excepting errors when reading some sectors. In such case you should fix the problem source (check if PCB contacts are oxidised, replace wires, provide reliable power, etc.) and continue accessing the device safely if there are no new errors and no other problems (and of course there was no hit, the device is not responding slow, etc.) Anyway it is recommended to make a copy to a healthy disk.

Device Cloning

During data recovery there may be multiple access to the same disk areas: programs usually scan the entire disk, then read again FS structures when building a directory tree for the selected volume, then read again FS and data when recovering selected files; different settings and software may be tried as well. The every additional attempt damages the device more and more.

So it is highly recommended to first clone the failing disk to a healthy device and work further with the copy.

In DMDE you may use the tool Copy Sectors to make the device clone. However the most safe and effective way is to use special equipment by data recovery specialists.

Selecting OS and Working in OS

Typically, an OS can actively interact with the device, causing its additional damage and interfering with its normal operation (in addition to unintentionally overwriting lost data). Recommendations for choosing an OS are given in the section Installation and Run.

It is also recommended to use the mode **IO SCSI** (in Windows/Linux, or **ATA**

Interface in DOS) as it is faster and less aggressive.

If there is no way to avoid using an OS that interferes with normal disk operation, then *as a last resort* you can temporarily "switch off" a partition table. In DMDE you can do this by using the 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 (**MBR On / GPT On**).

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 unplug 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 disks** (as well on modern **SMR** HDD disks). 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.

Removed Data Recovery Based on a File System

Unlike Raw recovery (signature based), FS based recovery allows recovering files with their original names and paths, and accounting the fragmentation (see Full Scan).

However if the data is lost due to removal then FS based recovery ability depends strongly on a file system type. Some file systems destroy FS structures immediatelly while others only mark files as removed keeping them available until not overwritten. File systems may also support journaling however a journal keeps a small amount of the most recent operations only. Raw recovery is a last resort when FS based recovery is not possible.

File System	Availability by FS	Remarks
NTFS	high	Large files (>2GB) may be truncated to a zero size by Windows NTFS driver

ReFS, APFS, Btrfs	high	Copy-on-write strategy leaves old structures on the disk until they are overwritten (unless **trim** is used)
FAT, ExFAT	medium	Only the first file fragment address is known after removal, on FAT32 the first fragment address may be not known exactly as well, no journal
HFS+/X, Ext3/4	low	On Ext FS file names may retain but file location information is wiped out

Data Recovery after Format

Full format in Windows, starting with Windows Vista, overwrites the partition with zeros. Still, only not rewritten data can be recovered (for example, formatting was interrupted, previously there was a different partition size, etc.).

Fast formatting usually causes only some loss of the original FS structure, some files. However, the initial fragmentation of FS structures, changing formatting parameters, the presence of garbage, and interspersions of other FSs (for example, from disk images) can significantly worsen the result of FS reconstruction.

When FAT / exFAT is formatted cluster tables containing information for restoring fragmented files are cleared.

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, quick search, and open volumes, as well as to perform basic operations on partitions on MBR- and GPT-style disks. Other styles (e.g., hybrid, dynamic) are unsupported whilst partitions may be shown as found.

Warning! Any disk modifications may cause further data loss (read more)

To enable modifications the option **_Advanced_** must be turned on. All changes remain virtual until you apply them using the button **_Apply_** or the context menu command **_Apply Partitioning_**. Write operations must be confirmed or enabled through the option **_Allow Write_** in the device parameters. See also the section Manage Changes.

Partition management supports modifications that change only partition tables and boot sectors without touching partitions contents. These include partition removal and undeleting, restoration of a boot sector from its copy, and other basic operations.

Partitions Window

When the "**_found_**" checkbox is ticked, lost and deleted partitions are displayed (an express search is performed). For a more complex and in-depth search, use the Full Scan option.

> To prevent the default express search uncheck the box **_Show partitions_** when open a device or construct a RAID.
>
> When volume is opened from the Full Scan results it is also added to the list of found partitions and can be restored (inserted) if not corrupted.
>
> Other info mode checkboxes:
>
> - **_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
>
> Found errors are colored in red. For incorrect/non-standard partitioning correct displaying is

not guaranteed. Some partitioning errors can be fixed by removing and inserting back the partitions. Prior to DMDE 3.8 FAT basic FS structures (F indicator) were being tested only on opening a volume.

The **Menu** button displays the operations available on the selected partition or the entire disk.

Open Volume

Open the selected volume to view and recover files.

Open Volume Parameters

View and manually edit volume parameters before opening the volume.

Show Volume Letters

Display volume letters currently assigned to partitions by OS.

Full Scan within the Partition

Run Full Scan or load full scan results from a file.

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 can 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.

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

[Raw File Search](#)

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 the volume can be opened before the scan completes. When choosing a volume, you can be guided by quality indicators and other parameters (details are [below](#)).

Before working with special types of storages (striped volumes / RAIDs, encrypted storages, disk images), refer to the [appropriate instructions](#) (in the section [Open Volume](#)), otherwise Full Scan will give incorrect FS results if you simply run on individual source disks. For regular partition, it is recommended to [open a physical device](#) and scan the entire area the partition may occupy (including previous and new locations) rather than trying to work with the damaged logical disk. The software may read beyond the selected area if it is necessary.

Different volume versions can be provided for the same volume, which are usefull **only** in the most complex cases (e.g. interrupted FS transformation: move / split / merge / resize / cluster resize, or for spanned volumes). In such cases, some files can be recovered correctly from one version of the volume, and others from the other, even though the same files may be visible on both versions of the volume. Open the selected version as a volume and [recover](#) the files. If files are missing or are not recovered correctly you can continue scanning or try another version.

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.
Log-file	Options to save the scan state and results automatically to a file
Menu	Besides, allows changing volume sorting order.

The columns **%** and ***Check*** allow preliminary estimating data recovery quality of a

found volume version (the percentage and number of correct interconnections between FS structures and files found by signatures for **NTFS, exFAT, HFS**; and for other file systems only FS structures are checked).

Min. Size is the minimum size estimate of the area occupied by the volume files (only the best found FS fragments are taken into account).

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

Raw File Search

File signatures are used as a last resort 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. They are also used to mark and filter files found both by a file system and by signatures, that is files with a high probability of recovery (see File Panel icons).

Raw results are available under the directory ***$Raw*** in the file panel when any volume is opened. To exclude results found by a file system of an opened volume, you can use the symbolic directory ***$Raw - Filter Raw Results*** (cluster map features are used to determine and exclude the space occupied by files).

Normally there are no reasons to open Raw results individually when there are FS results. 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 it is assumed a file continues until the next found signature, so searching for all present file types may improve size detection.

For some built-in file types, consecutive fragments can be combined into one file, so the number of files in the file panel may be less than in the scan results.

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

File Signatures

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

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

In DMDE you may then specify a custom file ***signature*** as **%PDF** (or as

\x25\x50\x44\x46 in hex codes). The question mark (**?**) can be used to specify a variable byte of the signature. The code **\mXY** specifes a fixed bit mask for the previous byte where **XY** is a mask hex value (e.g. the code **\x02\m03** cpecifies that only two least significant bits of the byte **\x02** are fixed).

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.

Adding New File Types

In addition to built-in file types and corresponding signatures, it is possible to add new types by the user (only if these types have unique signatures): Full Scan (Parameters) - *Raw: File Signatures* - *Add*.

In simple cases, it is possible to automatically calculate the signature of a new type on the base of sample files. You should push the button "*(+) Use Samples*" and specify possible extensions (comma separated) and the path where working files of the necessary type are located. If the recovery is being made from the camera, you can take a different photo / video to the new media in the same format to obtain 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.

> The option *use for FS calculation* enables checking interconnections between FS structures and files found by the signature. Additionally the files in the FS will be marked in a File Panel as found also by a signature (or not found).
> Multiple comma separated *Extensions* can 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.

Disk / Task Selection

Select disk / image / log-file for further work, or choose another task.

If a device is not accessible or has incorrect size see Installation and Run.

For the choice between options **physical disks** and **logical volumes** see Volume Opening.

The button **Parameters** allows to setup the interface and other Device I/O Parameters when working with problem devices.

In **Linux** you may use the option **Disk Images / Logs** to manually specify some device / virtual volume paths if they are not listed, e.g. **/dev/mapper/truecrypt1** or **/dev/mmcblk0**.

The option **Disk Images / Logs** allows opening disk image files and loading logs. Click or **Enter** the topmost list item to add a new image for further work. Besides standard sector-wise images, VHD / VHDX and VMDK formats are supported (without compression / encryption / write support); Professional Edition also supports E01 / Smart images (with compression).

If you need to open an image file with a non-standard/absent extention you need to change a filter option in the open file box. On **macOS** you may also need to click **Details/Parameters**.

The option **Show partitions** enables partitions express search and displaying just after opening a device.

Other Tasks

You may additionally load a saved RAID configuration / RAID calculation log or Full Scan log by using the option **Disk Images / Logs**.

The options **Construct RAID** and **Copy Sectors** redirect to corresponding tasks.

Device I/O Parameters

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

Interface

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

RW Access: open a device with write permissions (OS specific).

Allow Write: enable write operations for the device (global read-only mode must be disabled).

IO FILE: use default OS functions to access disks (the most compatible, but may be not optimal for failing drives, not supported in DOS for devices)

IO SCSI (OS Windows, Linux): use SCSI driver (lower level, recommended to reduce the number of OS I/O attempts, and to bypass some OS write blocking)

IO ATA (OS Windows): use lower level ATA driver for IDE / SATA disks (may be poorly implemented in OS)

Overlapped: use overlapped I/O in OS Windows (no practical use, not recommended)

DOS

- **ATA Interface** (direct access, recommended for disks with damages, see the requirements)
  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** (the most compatible access way in DOS)
  Old BIOS Service old CHS access supporting 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 loaded)

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.

> ***Error List***: specify a comma-separated list of error codes that will be skipped when the skip option above is enabled. When the user selects "***Ignore all***" in the I/O dialog box the corresponding error code is added to this list. If the list is empty then all errors are skipped.

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**, or even moreover the entire **I/O buffer** contents may be undefined on error for some devices/drivers.
>
> 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.

Pattern for representing bad sectors (hex): 4-byte little-endian value (in hex) to fill the data buffer when reading bad sectors (on I/O errors).

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.

Pattern for representing skipped sectors (hex): 4-byte little-endian value (in hex) to fill the data buffer for skipped/over jumped sectors.

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 (OS dependent, may be unsupported).

Update device handle / devices list on error: may be needed if a device was disconnected.

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 working with damaged devices.

> 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 the current operation.

> The operation will be aborted. The error sectors won't be copied to the destination (this behaviour may change in the future versions). The button is inactive if the operation can't be aborted due to the process logic.

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. The button is inactive while I/O is in progress.

Ignore: continue the operation ignoring the error.

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

Ignore all: continue the operation ignoring the same errors.

> The current error will be added to the Error List of error codes that will be automatically skipped after a certain number of attempts (defined by parameters) without waiting for the user choice.

Update: update the list of devices, and update the device handle. Use the additional parameter ***reopen device handle on error*** for automatic updating on each error.

ATA Reset: 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
%ERROREX%	additional error code for IO SCSI, IO ATA access modes
%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 (OS Windows and DOS)

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

RESETDEVLIST - Update the list of devices, reopen disk handle

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 can 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
```

RAID Arrays

Dialog box "Construct RAID"

Use Menu - Disk - **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 correct directory structure recovery).

RAID: Auto Calculate (Scan) option allows calculating unknown parameters and / or disk order for arrays with striping. Some disks can be absent (however the initial number of disks must be specified exactly). Most probable RAID layouts are being calculated statistically. The more precise possible RAID parameters are specified for calculation, the faster and more exact results are being obtained. E.g., RAID-5 software implementation in Windows OS uses backward dynamic rotation and a stripe size of 64K, thus, *Min-max stripe sizes* can be specified (in bytes) as **65536**, **65536** and only *RAID-5 bkdyn* rotation can be selected for detection.

Many software RAID implementations use containers, so there may be an offset between the start of the disk and the start of the RAID data on each of the disks. If the offset is unknown, the **LBA Offset Step** field can be used to test different offsets (for example, if the value **16** is specified then the sectors **0**, **16**, **32**, etc. will be tried as the data offset. Windows OS uses 1MiB alignment so a step **2048** LBAs (=1MiB) can be used. Some older OSes may use **63** LBA alignment. For a completely unknown offset, you can specify a value of **1** LBA, which means that every sector will be tried as the possible offset, which slows down the process significantly.

If data containers are found on disks, it is preferable to specify the containers instead of using LBA offset step. After adding a disk, you can double-click it (or use the menu - Partition/Offset) to specify the exact offset or exact partition (container) on the disk. With the button *Partitions*, the partitions on the disk can be observed and selected. Or, the additional field *LBA Offset* can be used instead to specify the known data offset.

In both of the latter cases, when the exact offset of the RAID data is known, the

LBA offset step can be set to **0** (zero) to disable offset detection and speed up the process.

If the order of the disk is known it can be specified as the ***fixed order of disks*** in the form **0,1,2** where the disk numbers are the numbers provided within the DMDE list of the disks added to the RAID (**#0, #1, #2**).

If the RAID parameters are known exactly they can be provided manually instead of using the option ***RAID: Auto Calculate (Scan)***.

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 can be recovered.

RAID-1. Usually contains two disks, data is duplicated. Stripe size is not used. Supported in read-only mode. One healthy disk can 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 can 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.

46

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 (in bytes). 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

Double click the individual disk (or use the menu command ***Partition/Offset***) to specify the ***offset*** and ***size*** of the selected extent in sectors

Null disk: used instead of absent or damaged disk

Striping...: manually specify striping for **Custom RAID**

Save/Load: save/load RAID configuration

Read ahead: improve the performance of working with the array (for cases with a small stripe size, not recommended on disks with bad sectors).

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 / Virtual FS]** to virtually reconstruct file system and display all found and deleted 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 [**F2**]/[**Shift+F9**]/[**Shift+F10**] or right click the item for a context menu.

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

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.

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
	[.]	normal 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 normal and "found" files (FS error)
📁	**{f}**	"found" directory (FS block number in square brackets follows name)
📁	**{.}**	"found" directory containing normal files
📁	**{x}**	"found" directory containing deleted files
📁	**{.)**	"found" directory containing normal and deleted files
📁	**(x}**	"found" deleted directory
📁	**(r}**	"found" deleted directory containing deleted files
📁	**(f}**	"found" deleted directory containing not deleted files
📁	**(.}**	"found" deleted directory containing not deleted and normal files
📄	**.**	normal file (within the base file system structures)
📄	**+.**	[normal] FS file also found by signatures (if Raw-search was performed)
📄	**-.**	[normal] FS file not found by signatures at the corresponding disk location, *maybe* damaged file / wrong type / not recognized
📄	**f**	"found" file (within additionally found FS fragment, 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 by signature, size detected, probably fragmented or 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 (displayed after FS reconstruction)

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.

Read the warning on this link before applying changes to a disk.

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 *Disk* - 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 can 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=** can be used, wildcards are allowed (e.g., **editortemplates=template***).

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

guid:{GUID} - the template identifier.
o:1 - the template can be applied to a not sector-aligned structure (offset can be applied).
fuse:0 - do not use the template.
flow:0 - display a 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 decimal and hexadecimal (**0x** prefixed) integers.

Variables

$RECSIZE - size of the record
$RECDEVOFS - position of the current record on the disk (in bytes from the beginning of the disk)
$NEXTOFS - offset applied to **data blocks** (see below)
$OFFSET - additional relative offset applied to **data blocks**
$XOFS - additional column offset for the output (see below **x:X**)
$1 ... **$4** - global custom variables (64-bit signed integers)
$varname - a local variable (64-bit signed integer), where **varname** is case-sensitive and may contain digits, latin letters, and underscores. A local variable must be initialized with the assignment operator **:=** and is valid only within the block of the

instructions where it is initialized.

Data Block

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

{Z} defines range of **Z** bytes starting at offset **$NEXTOFS** (and **$NEXTOFS** is increased by **Z** then);
{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** can be any variables or constants,
several ranges can 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 64bit)
%d - signed integer (up to 64bit)
%X / %x - hexadecimal (up to 64bit)
%c - ANSI character (8bit)
C - array of ANSI characters
U - array of Unicode characters (UTF-16)
u - utf-8 encoded string
T - array of text characters (depending on encoding table)
CXm - byte-by-byte hex output in multiple lines
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,Format outputs data block **{...}** at column **X** with the max. width **W** using the format **Format**.
x:X,w:W,c:C,Text outputs **Text** at column **X** with the maximum width **W** in color **C**. The **Text** can be enclosed in quotation marks (**"Text"**).

Parameters **w:W** and **c:C** are optional (**0** - the default color, **1** - title, **8** - red, **10** - grayed).

The output is performed at the current line, the instruction **=** (equal sign) moves output to the next line.

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

Conditions, 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

The switch instruction **$1:=TOGGLE:N,x:X** outputs a toggle box **[+]** (**[-]**) at the column **X** of the current output line, where **N** is a unique switch number (a variable or constant); and the variable **$1** takes value **0** or **1** depending on the switch state (toggling is by clicking the box or by pressing [**Spacebar**] when focused). This allows changing the template output on the fly (e.g. to expand / collapse structures, where switch number **N** is convenient to set equal to the structure offset).

The hot link instruction **$VAR=Param** makes a current output line a hot link, i.e. allows jumping to an object offset / record / or to open another linked object in the editor window by double clicking a line (or by hitting [**Enter**] key when the line of focused). **Param** is an expression value used by a link (an offset, record, file number, sector, etc.) and **$VAR** can be one of the following:
$GOTOREC, $GOTORECOFS - go to the specified record number and record offset of the object
$GOTOOFS - go to the specified object offset
$OPENLBA - open the specified disk sector
$OPENDEVOFS - open the specified disk byte
$OPENVOLSEC - open the specified volume sector
$OPENCLUSTER - open the specified volume cluster
$OPENVOLPAGE - open the specified volume page / object id
$OPENFILENUM - open a file record by the specified number
$OPENFILENUMDATA - open file contents by the specified number
$OPENFILERECOFS - open a file by FS entry offset in the object
$OPENATTROFS - open a data stream by FS attribute entry offset in the object.
Optionally specify a template to apply to the opened object by the instruction:
$OPENTEMPLATE='{GUID}'**; specify a template record offset: **$OPENTEMPLOFS=Param.

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** can 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 can be assigned in this section: e.g., **{U,Y}:={X,Y}** copies **Y** bytes at offset **X** to offset **U**.

Data postprocessing: instructions between lines **FLUSHSTART** and **FLUSHEND**. It is used for reverse operation when modified data is being written back to the disk

(data blocks may also be assigned).

Defines

Defines are used to replace repetitive blocks of instructions. The define can be declared in any template in the following way

DEFINE DefineTitle(%1%,%2%,...)

   ... (the define instructions where substitution variables **%1%,%2%,...** can be used)

ENDDEFINE

It can be used later in any template. On processing it will be replaced with the corresponding instructions and substituted variables:

DefineTitle($varname1,$varname2,...)

Cluster Map (File Allocation)

Cluster Map Report

The cluster map shows which clusters of the volume are in use and which files they belong to. Use the menu command **Windows · Cluster Map** to view / close the cluster map and **Cluster Map Compact Mode** to switch between viewing individual clusters and the whole file fragments.

To use cluster map first open volume and then **Update Cluster Map** (menu **Tools**) specifying the necessary parameters. You can specify to process only marked files, to process additionally FS structures and alternate data streams (the support depends on a file system), select File Categories.

The option **Use FS reconstruction** requires preliminary FS reconsruction in a memory and provides more possibilities (support for found and removed files, full file paths, and allows opening a corresponding file in the File Panel by double clicking the cluster (or hitting [**Enter**]) in the Cluster map). If [**Ctrl**] key is additionally pressed then file is opened in the Disk Editor instead. You can move the cluster map to a new tab to pin it and open files in a different panel.

The cluster map status line provides information for a selected cluster / file fragment:
[current cluster/clusters number] "file (stream) name" vcn: cluster number per file

To navigate between clusters you can use the Menu commands:
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.

Then the Menu command
Editor · Cluster Map [**Ctrl+M**] opens the cluster map at the given position,
Editor · File Data [**Ctrl+Alt+F**] opens a file located at the given disk position.

Cluster Map Report

The cluster map report allows obtaining a list of files located at specified sectors as well as obtaining a list of clusters occupied by specified files. To create a list of clusters occupied by specified files you should mark the necessary files in the directory tree and then use menu **Tools · Update Cluster Map** to rebuild the map with the option **marked only** and then run a report.

You can specify to report **only files**, to report file **paths**, and to report fragments boundaries **in bytes from the disk start** to use the report with different software.

It is also possible to obtain information about file allocation without building a cluster map by using the option **List** in the dialog box "Recover".

Menu

- *Disk*
    - *Select Disk / Task...*
    - *Open Image / Log...*
    - *Construct RAID...*
    - *Partitions...*
    - *Lock Volumes...*
    - *Device I/O Parameters...*
    - *Global Read-only Mode*
    - *Load Rollback Data/Dump from File...*
    - *Dump Changes to File...*
    - *Undo Changes*
    - *Redo Changes*
    - *Reset All Changes*
    - *Apply Changes*
- *Tools*
    - *Full Scan...*
    - *All Found / Virtual File System...*
    - *Search in Folders (by Name)...*
    - *Recover / Create File List...*
    - *Restore EFS from Backup...*
    - *Update Cluster Map*
    - *Cluster Map Report...*
    - *Reopen Volume Parameters...*
    - *NTFS Tools*
    - *Copy Sectors...*
    - *Fill Sectors...*
    - *Search...*
- *Windows*
- *Editor*
- *Mode*
- *Edit*

Menu "Disk"

- *Select Disk / Task...*
- *Open Image / Log...*
- *Construct RAID...*
- *Partitions...*
- *Lock Volumes...*
- *Device I/O Parameters...*
- *Global Read-only Mode*
- *Load Rollback Data/Dump from File...*
- *Dump Changes to File...*
- *Undo Changes*
- *Redo Changes*
- *Reset All Changes*
- *Apply Changes*

Manage Changes

Warning! Direct modifying disk contents may have further consequences including data loss or inability to load and run the system.

When an OS discovers new FS errors (as well gains access to FS with old errors) it can run a disk checking utility to remove errors and this may also cause the removal of user data. The OS can run the checking utility online or at boot time.

Partitioning modification may also cause impossibility to boot from the disk or inaccessibility of some partitions. Never modify partitioning on your current boot or system disk. Run another system from different bootable disk or connect your disk as slave to another computer. Do not run the software from the disk which will be modified.

Direct disk changes take effect to the system after restarting the computer, updating disk configuration, or reconnecting the pluggable device. OS Windows may disallow direct writing to some disk areas being in use - see Volume locking for details. Some antiviruses may also block direct writing to a disk. Under Windows you may also try the interface option **IO SCSI** to circumvent the restrictions.

Changes made with commands Copy Sectors and Fill Sectors are written to the disk immediately.

Changes made in Disk Editor and Partition Manager remains virtual until you explicitly apply them to the disk. You can use the following **Disk** menu commands to manage virtual changes.

Global read-only mode

Globally disable all write operations (for all devices and device images).
See also a device specific option **Allow Write**.

Load Rollback Data/Dump from File...

You can load rollback data (if it was saved when applying previous changes) and thus revert the previous changes. Also you can use the command to load virtual changes dumped to a file.

Dump Changes to File...

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

Undo Changes

Undo the group of recent changes. To undo individual actions use the corresponding commands in Disk Editor (menu Edit) or Partition Manager.

Redo Changes

Redo the group of virtual changes.

Reset All Changes

Discard all unapplied changes.

Apply Changes [*Ctrl+W*]

Write all changes directly to the disk.

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 **Disk** · **Lock Volumes for Write**.

Menu "Tools"

- *Full Scan...*
- *All Found / Virtual File System...*
- *Search in Folders (by Name)...*
- *Recover / Create File List...*
- *Restore EFS from Backup...*
- *Update Cluster Map*
- *Cluster Map Report...*
- *Reopen Volume Parameters...*
- *NTFS Tools*
- *Copy Sectors...*
- *Fill Sectors...*
- *Search...*

Volume Commands

Search in Folders (by Name)

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

You can move the search panel to a new tab to pin it and open directories in a different panel.

Update Cluster Map

Update file allocation information.

Reopen Volume Parameters...

View the parameters and reopen the volume without the results of Full Scan and file system reconstruction.

Virtual File System Reconstruction

Open virtual directory **[All Found / Virtual FS]** 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.

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).

Non-FAT File Systems

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

65

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 (can be useful if there is not enough RAM for the entire reconstruction).

improve INDX merging

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

auto INDX processing (NTFS)

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

process FS journal (NTFS)

Include information from the NTFS $LogFile

Include Shifted (NTFS)

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

Include 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).

Include Tech. Records (NTFS)

Include special MFT records into the reconstruction for analysis.

Volume FS Fragments

Manually select FS fragments (found during the Full Scan) to use in the reconstruction. Fragments are grouped by probability of belonging to the open volume:
the best, ***correct*** - most likely and likely to belong to the open volume,
related - probably related to other versions of this volume (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.

ReFS

In ReFS v3 any directory node contains a volume id. ***Other*** nodes in the ***FS Fragments*** list are nodes with an id which is different from the opened volume id.

In a correct FS ***top*** nodes contain links for all ***rest*** nodes (intermediate and leaf nodes). Upon any change ReFS writes new node versions in a new location and update links so there may be many ***old directory nodes*** without references. However for a damaged FS there may be also lost nodes among them. By default ***old directory nodes*** are processed as well but a file version control is used to exclude old file refernces (unless the option ***include all old file entries*** is turned on).

66

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

Working with Failing Drives   Opening Disk Image Files

A tool for creating and restoring disk images, cloning entire disks, making dumps.

When you select a file as a destination an image file will be created.

When you instead select a disk/partition as a destination, the write is performed directly to a disk, **all data on the destination disk will be overwritten**, the destination file system will be erased.

If you write something to a disk containing lost data (e.g. applying a patch) **this may cause further data loss**.

Source and Destination

Device	Select Device, Logical disk (volume), Image, virtual RAID as a source/destination
Partition	Select Partition as a source/destination (use ***Device*** button to change the device),
	to select the entire disk double click the topmost item in the partitions list
File	Use file as a source/destination (***Device*** · ***Disk Images*** can also be used)
Editor	Use the object opened in the Disk Editor window as a source
Selected	Use the selected block in the Disk Editor as a source

Fields ***Start Sector/Byte***, ***End Sector/Byte***, ***Number of Sectors/Size in Bytes*** are filled in automatically when the source/destination is set by a button. If a ***File*** or ***Editor*** is used as a source then fields are specified **in bytes** (otherwise they are **in sectors**).

You can manually specify a range within the source object for copying and the starting offset for the destination. ***Start Sector/Byte*** of the **source** will be copied to the ***Start Sector/Byte*** of the **destination** regardless the direction of copying. For a partition, ***Start Sector*** is relative to the partition beginning (not to the device beginning).

The button **Split** allows splitting the destination file into parts. A RAID configuration `raidinfo.ini` is also created to open later the splitted image or to continue writing the splitted image (to specify/open using Device - ***Disk Images / Logs***).

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 in the log 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).

Menu

Save/Load: save/load settings to a file/from a file.

Export log to ddrescue mapfile: convert log-file to a ddrescue mapfile format.

Export bad and skipped ranges from log: get a list of bad and skipped sectors. The list can be then used to get information about damaged files located in bad sectors (see Data Recovery).

Fill Sectors

Write a hex or a file pattern to a range of device sectors. Pattern is being written multiple times, covering all specified sectors.

Read the warning on this link before modifying disk contents. Write is performed directly to the disk, all data will be overwritten, the operation cannot be undone.

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 a 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.

Settings for searching a repeating pattern

X times in a row: search for a pattern that repeats in a row the specified number of times
Seek for end of repetitions: if the pattern starts repeating in a row then seek just for the end of repetitions (instead of seeking every repetition)

These settings, for example, can be used to find the beginning and end of arrays of zeros or other values.

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](#)
- [Cluster Map](#)
- [Preview](#)
- [Search Panel](#) (Search in Folders)
- [Move to a New Tab](#)

Preview

When the option is enabled, the File Panel opens graphic files in the image view mode by default.

Move to a New Tab

Open the current panel in the new interface tab. If it is a Search Panel or Cluster Map then the panel will be pinned and new objects will be opened in a different panel.

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