

# Acronis DiskEditor User's Guide



Copyright © SWsoft 2001-2002. All rights reserved.

Linux is a registered trademark owned by Linus Torvalds.

OS/2 is a registered trademark owned by IBM Corporation.

Unix is a registered trademark owned by The Open Group.

Windows is a registered trademark owned by Microsoft Corporation.

All other mentioned trademarks may be registered trademarks of their respective owners.

Distribution of materials from this Guide, both in original and/or edited form, is forbidden unless a special written permission is obtained directly from it's author.

THIS DOCUMENTATION IS PROVIDED «AS IS». THERE ARE NO EXPLICIT OR IMPLIED OBLIGATIONS, CONFIRMATIONS OR WARRANTIES, INCLUDING THOSE RELATED TO SOFTWARE MARKETABILITY AND SUITABILITY FOR ANY SPECIFIC PURPOSES, TO THE DEGREE OF SUCH LIMITED LIABILITY APPLICABLE BY LAW.

# Table of Contents

<b>INTRODUCTION</b> .....	<b>5</b>
<b>1. GENERAL INFORMATION</b> .....	<b>8</b>
1.1 FILES. PARTITIONS. CONNECTING A HARD DISK. BIOS SETTINGS.....	8
1.1.1 FILES, FOLDERS, FILE SYSTEMS .....	8
1.1.2 HARD DISK PARTITIONS AND SECTORS .....	9
1.2 CONNECTING A HARD DISK TO THE COMPUTER.....	10
1.3 SETTING BIOS .....	10
1.4 INSTALLING THE PROGRAM .....	11
1.5 USER INTERFACE .....	19
1.5.1 USING MOUSE .....	20
1.5.2 USING KEYBOARD.....	20
1.5.3 WINDOW .....	21
1.5.4 DIALOG, AND ITS CONTROL ELEMENTS .....	23
1.5.5 INFORMATIONAL ELEMENTS.....	25
<b>2. BEGINNING TO WORK WITH THE PROGRAM</b> .....	<b>27</b>
2.1 PARTITION OR DISK SELECTION WINDOW .....	27
2.2 MAIN WINDOW OF THE PROGRAM .....	29
2.3 DISK MENU .....	31
2.4 EDIT MENU.....	33
2.5 VIEW MENU .....	37
2.6 SEARCH MENU .....	38
2.7 HELP MENU.....	40
2.8 TOOLBAR .....	41
2.9 CONTROLLING THE PROGRAM WITH KEYS.....	41
2.10 THE LIST OF ENCODINGS .....	42
<b>3. MAIN WINDOW VIEW MODES</b> .....	<b>43</b>
3.1 SELECTING PARTITIONS (DISKS) FOR WORKING .....	43
3.2 MASTER BOOT RECORD (MBR).....	44
3.3 CREATING PRIMARY PARTITION .....	46
3.4 FORMATTING A PARTITION .....	48
3.5 CREATING A LOGICAL PARTITION.....	51
3.6 FILE ALLOCATION TABLE (FAT) AND ROOT FOLDER (ROOT) .....	52
3.7 FOLDERS AND SUBFOLDERS.....	58
<b>CONCLUSION</b> .....	<b>59</b>
<b>APPENDIX A. GENERAL INFORMATION. HARD DISKS</b> .....	<b>60</b>
A.1 HARD DISK ORGANIZATION .....	60
A.2 INSTALLING HARD DISKS ON COMPUTERS.....	61
A.2.1 INSTALLING A HARD DISK. GENERAL SCHEME.....	61
A.2.2 MOUNTING HARD DISKS IN CASES.....	62
A.2.3 MOTHERBOARD SOCKETS. IDE CABLE. POWER CABLE .....	62
A.2.4 INSTALLATION MODES. JUMPERS.....	63

A.3	BIOS .....	64
A.3.1	SETUP UTILITY .....	65
A.3.2	STANDARD CMOS SETUP MENU .....	66
A.3.3	ARRANGING BOOT SEQUENCE. ADVANCED CMOS SETUP MENU .....	68
A.3.4	AWARDBIOS .....	69
A.3.5	HARD DISK INITIALIZATION ERRORS .....	74
A.3.6	BOOTING OPERATING SYSTEMS .....	75
<b>APPENDIX B. OPERATING SYSTEM AND HARD DISKS.....</b>		<b>76</b>
B.1	HARD DISK PARTITIONS .....	77
B.2	CREATING PARTITIONS FOR OTHER OSES .....	78
B.3	PRIMARY AND LOGICAL PARTITIONS .....	79
B.4	FORMATTING HARD DISKS .....	79
B.5	FILE SYSTEMS .....	80
<b>APPENDIX C. FILE SYSTEMS .....</b>		<b>82</b>
C.1	FAT16 .....	82
C.2	FAT32 .....	83
C.3	NTFS .....	83
C.4	LINUX EXT2 .....	83
C.5	LINUX EXT3 .....	84
C.6	LINUX REISERFS.....	84
<b>APPENDIX D. GLOSSARY .....</b>		<b>85</b>

# Introduction

## Storing information on a hard disk

A hard disk is a device for long-term information storage. From an end-user's point of view, information is simply a set of documents or files.

Documents can be stored on a hard disk as files of various types or formats – text files for example, in popular MS Word format; tables in MS Excel format; databases, images in JPG or GIF formats; Internet HTML documents; music recorded in mp3 format; video records (AVI, MOV); and many-many others.

Irrespective of the type of information (or data), all files are stored on a hard disk as sequences of elementary information storage units chained together called hard disk clusters.

## Hard disk

A hard disk is a very effective device for storing information. Modern hard disks feature huge volumes of stored information, very high recording densities measured in Gb (gigabytes) or Mb (megabytes) per square inch, and very high access speed to retrieve stored information.

A hard disk is usually an extremely reliable device. One of its parameters – non-failure operating time – may reach 20,000 to 500,000 hours.

If one calculates continuous round-the-clock work of a hard disk for one year its operating time equals 8760 hours. Hence, the expected operating life of a hard disk drive should be 3 to 5 – or even more years. The vast majority of hard disks will work without failures until they become obsolete.

## Information storage errors

This does not mean, however, that storing information on hard disks is 100% error-free. Information on hard disks maybe threatened by a number of factors every user should be aware of. These threats are both environmental (external) and those in one's information system (internal).

Information storage errors are caused by various factors: software errors (from an operating system or an application), computer viruses, power supply failures, and others. User's also face data files accumulating mistakes, files may collapse, with applications that may not be accessible or executable. User's must diligently keep their personal computers operating safely and efficiently by providing the computer with a constant power supply, maintenance of back-ups covering the full contents for their hard disk drives, and perform preventive maintenance of

disk regularly (check for file system errors and bad sectors, defragment files and drives, and check for possible computer viruses).

### **Hard disks utility programs**

There are program tools that allow a user to inspect various computer systems, including disk subsystem. Disk managers included in these tools, are capable of defragmenting hard disks, finding and correcting files and folders structure imperfections, as well as finding and marking bad sectors. Special utilities are capable of finding and destroying computer viruses that threaten your information integrity.

In some cases a user will require a much more powerful set of tools, known as a **disk editor**, to repair their hard disk drive. The Acronis DiskEditor is exactly the software for this job.

Acronis DiskEditor is a professional-grade program that performs several actions on a hard disk: to restore boot records and files and folders structure, to find lost clusters, and even to remove computer virus code from a disk.

This program has much to offer a beginner as well. It can be used in an educational setting. It is one thing to read a few books about hard disk construction and data storage, and it's quite another to observe it yourself with the Acronis DiskEditor.

### **How to find the necessary Guide section**

A review of hard disk basic properties, methods of storing information, as well as information about Acronis DiskEditor installation can be found in the **1. «General information»**.

For information on the basic features of the program and its functionality, please see **2. «Beginning to work with the program»**.

Information about modes of viewing hard disk and the overall performance of working with a disk see **3. «Selecting partitions (disks) for working»**.

Additional information about hard disk construction and interacting with operating and file systems is contained in Appendices: **Appendix A. «General Information. Hard Disks»**, **Appendix B. «Operating System and Hard Disks»**, **Appendix C. «File Systems»**.

Finally, at the end of this Guide you will find our description of terms concerning hard disks and file systems: **Appendix D. «Glossary»**.

## **Notational convention**

The following notational conventions are used in this User's Guide.

Names of screen buttons used are marked with the following font in the text:  
**OK, CANCEL, HELP.**

The following notation and font are used to mark keys and their combinations:  
**F2** and **ALT+P.**

Names of program windows are marked bold in the text (without quotes):  
**Search, Go to sector.**

The same bold font marks names of control elements: fields, lists, checkboxes, and radiobuttons.

For checkbox control elements the expression «check a box» is used, for example, «check the **Match case** box».

# 1. General information

A hard disk is a rather complex information storage device. One should have the base knowledge of hard disks and their interaction with the computer to work effectively on them using the Acronis DiskEditor program. If you are ready to work on a hard disk with a disk editor, it is necessary to remember that mistakes can cause severe problems: you could violate the disk's information integrity, not only making any data inaccessible, but also unserviceable.

It is impossible to offer complete information on organizing data storage on a hard disk and on its interacting with operating system and applications in such a short Guide. Thus we will inform you only about the most vital data below skipping many complex technical details of information storage and the internal construction of a hard disk. The general outline will be on the capabilities of Acronis DiskEditor program.

Capabilities of Acronis DiskEditor will be demonstrated using the example of the FAT32 file system – the most widely used file system in the personal computer world.

If you have difficulties understanding what operation the program performs on your hard disk, please refer to the **Appendices** at the end of this Guide. The Appendices detail both hard disks construction, and how data storage is organized on them.

At the end of this Guide is the **Glossary** where you will find explanations of unfamiliar or obscure **terms** you may meet in the body of the Guide.

## 1.1 Files. Partitions. Connecting a hard disk. BIOS settings

### 1.1.1 Files, folders, file systems

You know well that data is stored on hard disks in the form of **files**. Files are grouped into **folders** and subfolders. Folders, in turn, are arranged into **folders tree**. For Windows operating system the **root folder** is of special value. It is usually denoted as, for example, C:\ (root folder on C: hard disk). The user addresses files and folders by their names. Operating systems allow users to create files (and folders), to move/copy files and subfolders from one folder to another, to rename and move files, etc.

Files can be executed (so-called executables) or be opened with an application that created them.



An operating system provides its user with ability to work with data arranged as files by supporting some type of **file system**. Most operating systems support several file systems. For example, Windows 98/Me support FAT16 and FAT32 file systems, Windows NT/2000/XP support both FAT16/32 and NTFS file systems. A popular Linux operating system supports Ext2 and ReiserFS (and some other) file systems.

File systems **differ** by the way they organize data on a disk. File systems of different OS families are generally **incompatible** with one another.

Files on disks consist of elementary data storage units – **clusters** that are joined into chains.



For more information about file systems and files storage organization on a hard disk see **Appendix A. «General Information. Hard Disks»**.

### 1.1.2 Hard disk partitions and sectors

It would be extremely inconvenient if only one operating system could access each hard disk, so a mechanism was created that allows access to hard disks from several Operating Systems. This is done by partitioning hard disks into **partitions**.

Partitioning is done by special applications. MS-DOS and Windows do this with a widely-known FDISK program that allows creating partitions, choosing their size and labeling. This program allows designating all the hard disk space to MS-DOS or Windows or just a part of it, while leaving the rest for other operating systems.

A disk partition is used by a single operating system that supports its own file system.

Acronis DiskEditor program provides **low-level** access to information on hard disks. This means:

1. working with DiskEditor program you have direct access to **each byte** on any hard disk sector (or a disk partition);
2. within the limits of the Partition Table, File Allocation Table, Root folder each sector byte (or several bytes) is interpreted differently, therefore it would be extremely difficult and inconvenient to work with a disk editor without a set of **patterns**. Such patterns are view modes of Acronis DiskEditor main window.



For more information about hard disk partitions and their use by operating systems see **Appendix B. «Operating System and Hard Disks»**. For more information about view modes of Acronis DiskEditor main window see: **3.1. «Selecting partitions (disks) for working»**.

## 1.2 Connecting a hard disk to the computer

It is, certainly, better to master Acronis DiskEditor program not with your main hard disk. Connect to the computer any old hard disk of small capacity to experiment fearlessly with. Install this disk into the case of the personal computer and connect it to the power supply unit and the system board.

Before installation and fastening a hard disk you must set a **jumper** on an electronic board of the disk or on a special socket to which power and data cables are connected. The jumper on the main hard disk of your personal computer is, most likely, set to **Master** position. You should set the jumper on an auxiliary disk to **Slave** position.

The overwhelming majority of hard disks for in personal computers use the so-called **IDE-interface**. These disks are installed in the personal computer case to a compartment for devices with 3.5" **form-factor** (see: **Appendix A. «General Information. Hard Disks»**).

A standard power plug is connected to a hard disk along with 40- or 80-wire IDE data cable (its another end should be plugged to system board's Primary IDE socket; if there are hard disks on your computer already you should use another socket – Secondary IDE).



There are also numerous models of hard disks with the so-called SCSI-interface. They are usually plugged to sockets on a special SCSI-controller board. The amount of different cables used to connect such disks is rather great. Such disk models will not be considered here, as the main purpose of the Guide is to describe Acronis DiskEditor program (and file systems).



For more information about installing a hard disk into the personal computer case and its connecting see section **A.2 «Installing Hard Disks on Computers»**.

## 1.3 Setting BIOS

It is not enough to properly position jumpers on the hard disk and connect it to power supply and data cables. You also have to properly configure your hard disk in **BIOS** (Basic Input/Output System).

You can enter the BIOS setup program by pressing the key combination that is displayed right after you turn on your computer.

Hard disks can be configured in the Standard CMOS Setup section of BIOS setup menu. If you have set the hard disk jumper to Slave and have connected the data cable to the Primary IDE plug of the mother board, usually it is enough to set the TYPE and MODE parameters for the Primary Slave disk to Auto (this information is sufficient for BIOS to properly detect and configure modern hard disks.)

After the new disk is configured in BIOS, save the settings.



For more information about BIOS settings and also about possible concomitant errors (and error messages) see section **A.3 «BIOS»** (and especially about errors: **A.3.5 «Hard Disk Initialization Errors»**).

## 1.4 Installing the program

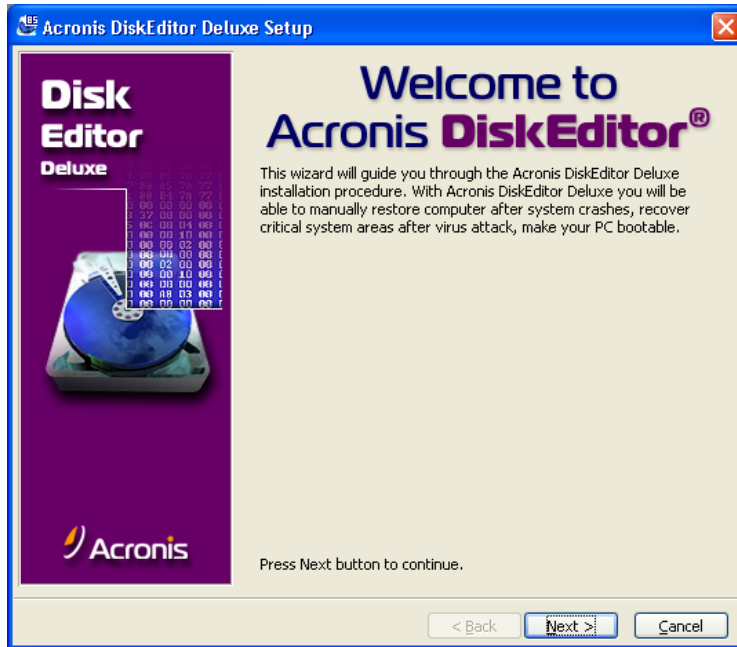
You can purchase the Acronis DiskEditor on a compact disc (CD-ROM) or download it, after purchase, from our secure server Internet site.

The CD-ROM contains an installer that can be run in Windows 9x/NT/2000/XP operating systems.

If you purchased DiskEditor on CD-ROM, to install do the following:

1. Insert the CD-ROM into the CD-ROM drive.
2. Run Windows Explorer, mouse-click the icon of the CD-ROM drive. You will see setup.exe executable on the compact disc. Double-click it to start the installation. (If autorun mode is enabled, the installation will be started automatically.)

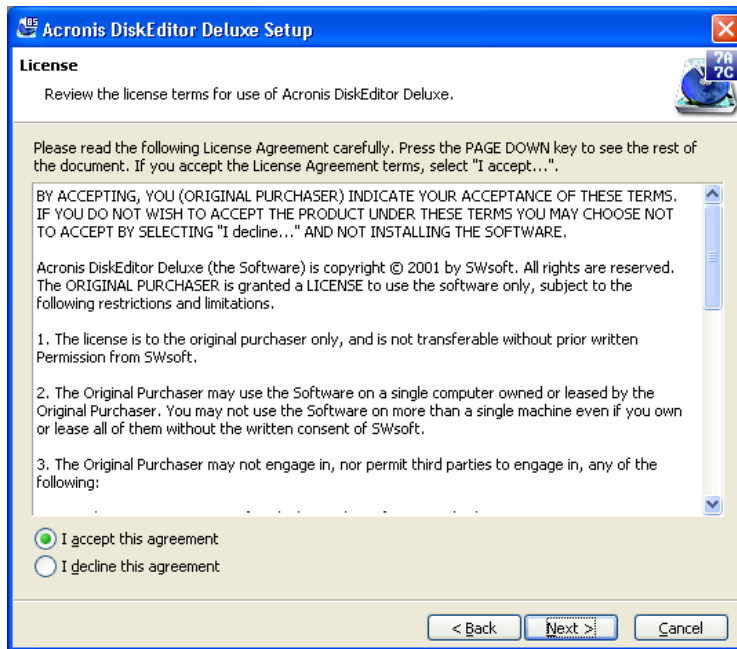
The welcome page of Installation Wizard will appear on the screen after installation is started:



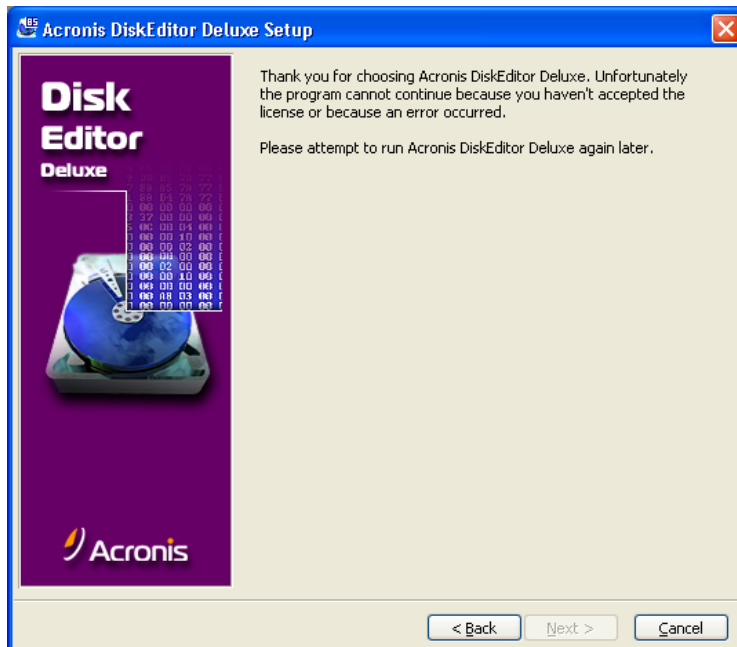
Click **NEXT** button to continue.

To continue working with the program you must accept the license agreement between you (user) and the program developer.

To do this set the switch on the bottom of the next page to **I accept this agreement** position.

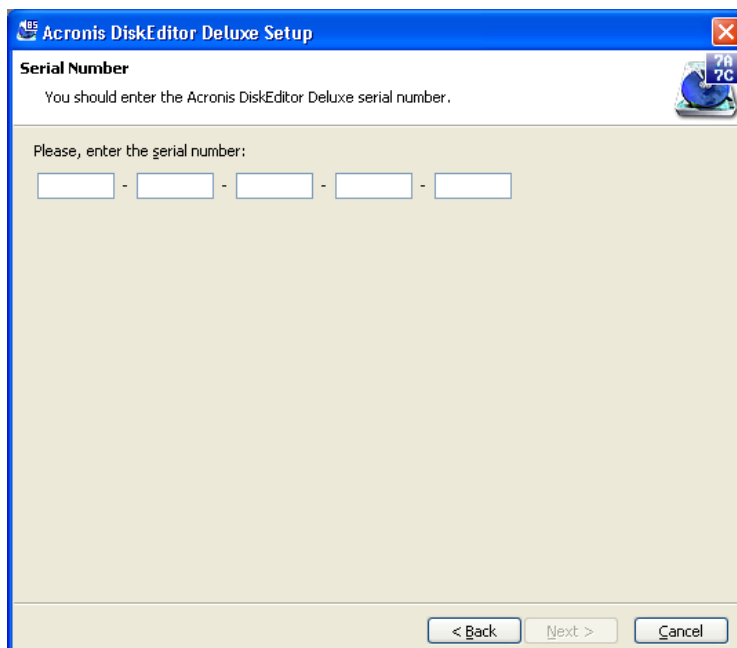


If you disagree with the agreement, set the switch to **I decline this agreement** position and click **NEXT**. In this case you will be thanked for your interest to the program, and the program will stop working and will be unloaded from memory.



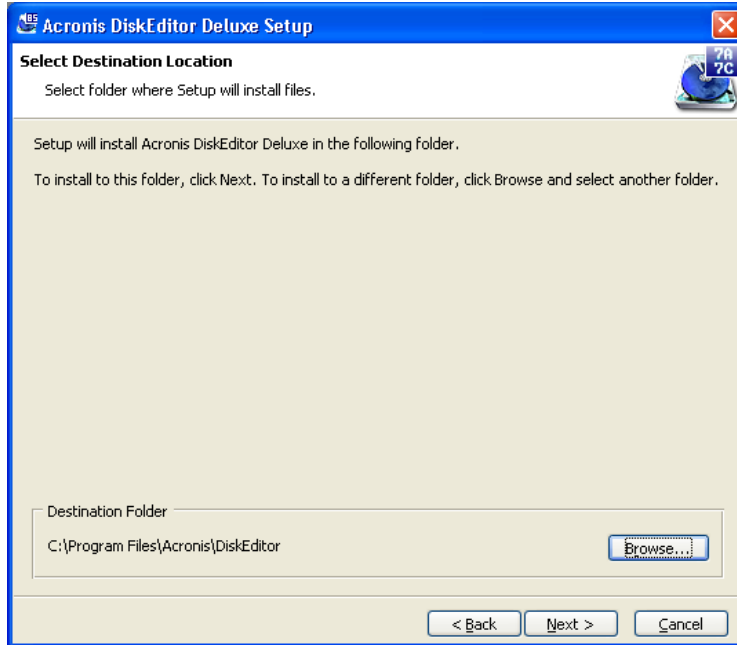
Click **NEXT** button to continue.

In the next window you will need to enter the product **serial number** that you will find on the Acronis DiskEditor registration form – AND DON'T FORGET TO REGISTER YOUR SOFTWARE TOO!

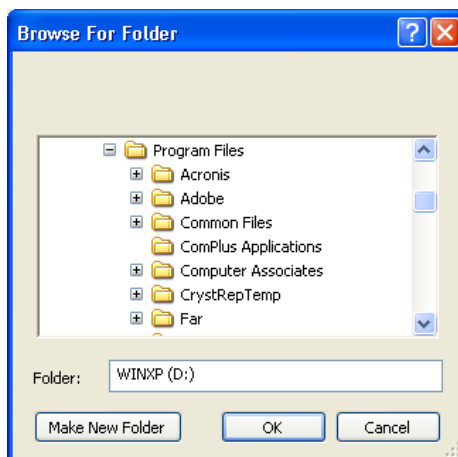


Having entered the serial number, click **NEXT** to continue.

On the next page you should specify a folder to install DiskEditor program into. The default folder is C:\Program Files\Acronis\DiskEditor. If you agree with it, click **NEXT**. If you want to install the program in a different folder, enter its name and path in the appropriate field. If there is no such folder on the disk, it will be created during installation.

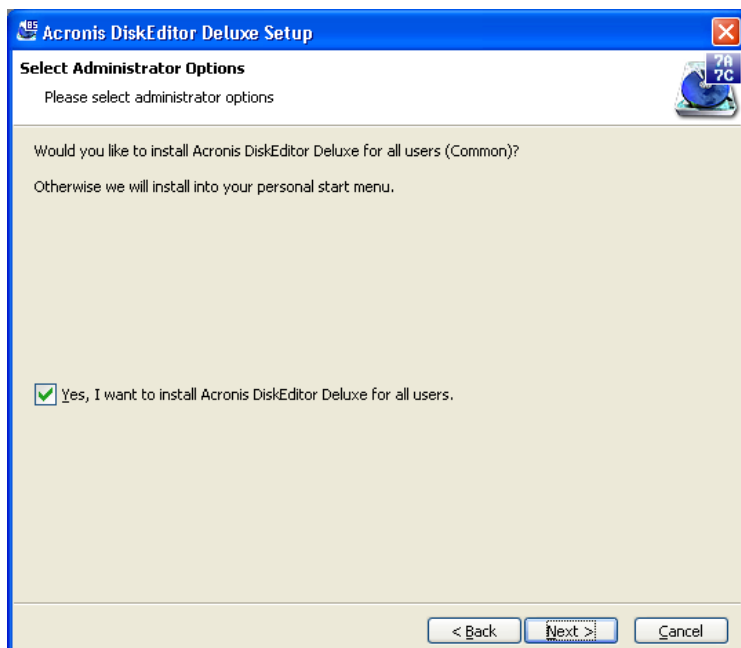


You can choose a folder for program installation by clicking **BROWSE....** You will see the folder tree of your hard disk. Select a folder with mouse and click **ACCEPT**. (You can select any hard disk in your PC for installation.)



After selecting a folder, click **NEXT** to continue.

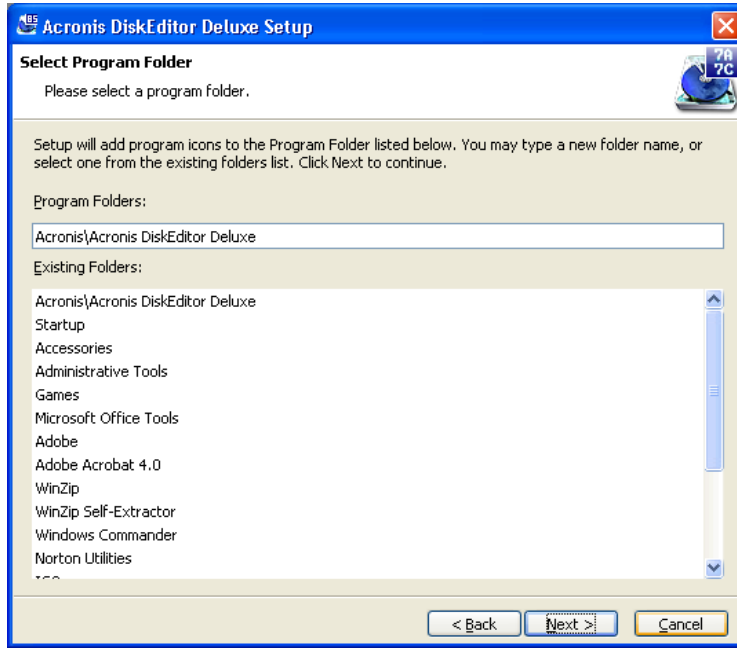
On the next page you will be offered to make DiskEditor available to all users of the PC or only to the current user. If you want to make the program available to all users, check Yes, I want to install Acronis DiskEditor Deluxe for all users box. (This page showing the Installation Wizard is specifically for the Windows NT/2000/XP operating systems.)



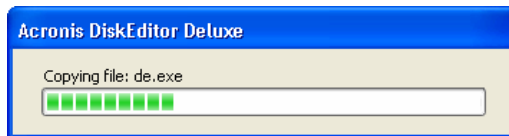
Click **NEXT** to continue.



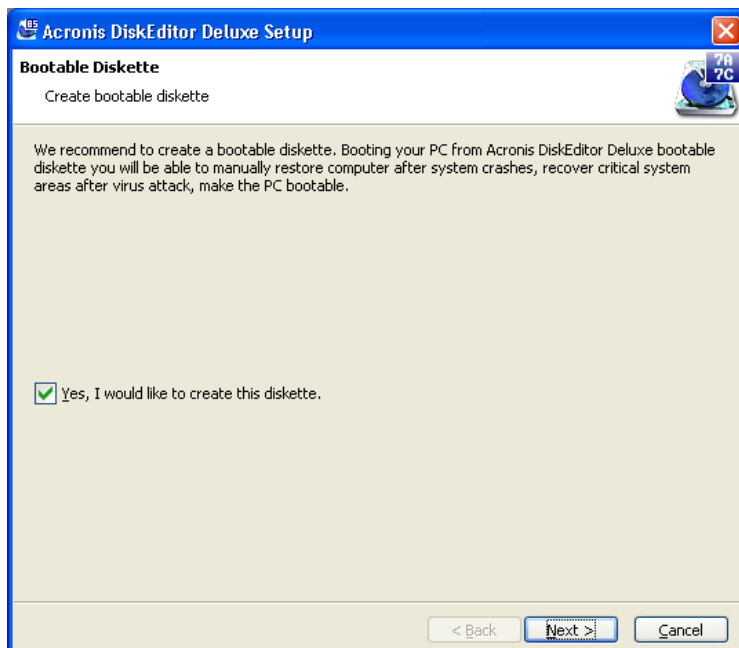
In the next page you will be asked to create a new program group for Acronis DiskEditor program or to select an existing group. If you agree to create a new group click **NEXT**. If you want to use one of the existing program groups, select it by clicking it with the mouse.



After program group is selected the installation of Acronis DiskEditor program will be started and you will see the window showing its progress.



After installation you will be offered to create a **bootable diskette** with DiskEditor. We advise you to follow this recommendation. In case of damage to disk partition bootsectors, you will be able to restore your PC with the help of this diskette.

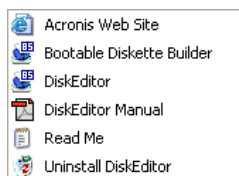


Prepare a clean formatted diskette, insert it into the floppy drive. Set the **Yes, I would like to create this diskette** switch. Click **NEXT** button to continue.

You will also be able to load DiskEditor with this diskette in case you need to make critical changes in the BIOS boot sequence.

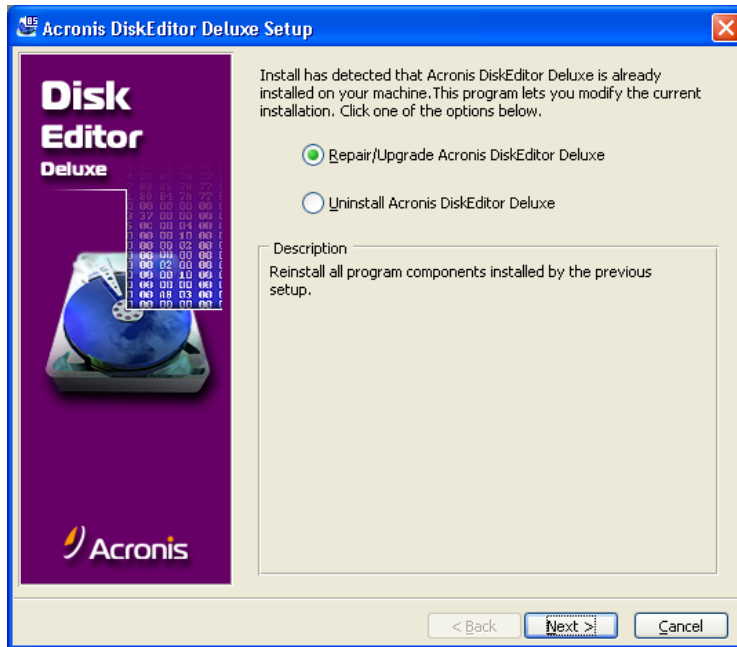
Set up ends with the message about its successful finish.

As a result of the installation process, the **Acronis** group will be added to the Programs submenu of Windows Start menu. Mouse-clicking it you will see the **DiskEditor** submenu containing the following:



Clicking the Acronis DiskEditor Deluxe line starts the disk editor.

If the DiskEditor's work files becomes corrupted, you will be able to restore program by re-running the installation program from the compact disc. The installation program will determine that Acronis DiskEditor was already installed on your hard disk, and you will see the following Installation Wizard page:



By default you will be offered to restore damaged files. Click **NEXT** button to continue.

## 1.5 User interface

The Acronis DiskEditor program has a graphic user interface and is controlled by mouse or **TAB**, **SHIFT+TAB**, **LEFT**, **RIGHT**, **UP**, **DOWN**, **SPACE**, **ENTER** and **ESCAPE** keys.



If you routinely work with applications under Windows, X-Windows, or OS/2, there will be no problems understanding and using the Acronis DiskEditor interface. However, this Guide contains a detailed enough description for users unfamiliar with standard interfaces.

Within the Acronis DiskEditor program, users work with the program's **main window** (or several main windows) which allows access all disk editor functions.

Editor functions are called with the help of «hot-keys» menus and control elements (screen buttons, lists). The necessary switch position (condition) is **selected (set)** with the help of mouse or by pressing keys.

The program's main window, as well as all dialog windows, are supplied with brief help information that appears in a separate window after clicking **HELP** or selecting the **Help -> Contents** line in the main window's menu (or after pressing **F1** key).

Elements of the user interface and its use are described below in detail.

### 1.5.1 Using mouse

The Windows interface of Acronis DiskEditor assumes you are using a mouse.

In this Guide, «mouse-click a button» and «click a button» expressions are used instead of «move the mouse pointer over a button (an object) and click the left mouse button (press and release the left mouse button)» expression.

The Mouse helps to check boxes, press buttons, select/de-select list elements (lines).

You can close the program window by selecting **Disk -> Close** line in the menu.

To exit the program with help of the mouse select **Disk -> Close**.

### 1.5.2 Using keyboard

All Window actions control elements performed with the mouse may also be performed with the help of the keyboard.

Switching between control elements is performed with the **TAB** key and **SHIFT+TAB** key combination. The active or focused element of the screen are marked with a dotted frame around it or by its perimeter (for screen buttons).

Checking boxes (or setting switches) are performed by pressing the **SPACE** key.

Switching between the pages of installation wizard windows is performed by pressing **NEXT** (to the following page) and **BACK** (to the previous page) buttons.

You can close the program window by pressing the **ESCAPE** key.

### 1.5.3 Window

All actions performed with the Acronis DiskEditor are displayed in windows. A window is a rectangular area on the screen that has a title and a frame.

#### Window frame

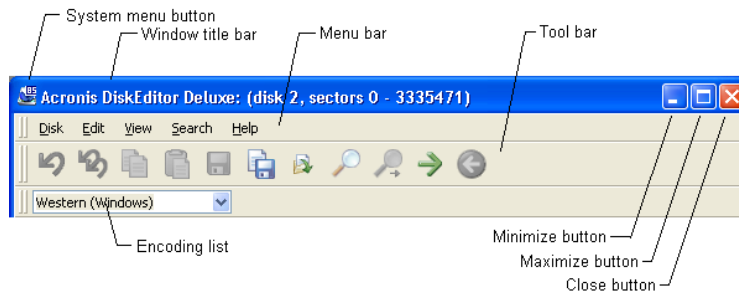
A frame is a line that bounds window rectangle. All window contents are located inside this line.

The frame allows you to change the size of the window. For this purpose move the mouse cursor over the frame until it turns into a double arrow. Press the left mouse button and move the mouse until the window is sized according to your needs, and then release the button. The bottom of a window frame is used to change the height of a window, and left and right frame parts – to change the width.

#### Window title bar

A title bar limits the window from above and contains the text line describing the window (a title) and minimize/maximize and close window buttons.

It is also possible to change the window screen position with the help of the title bar. For this purpose move the mouse cursor over the title bar, press the left mouse button and move the mouse until the window is positioned on the screen according to your needs, and then release the button.




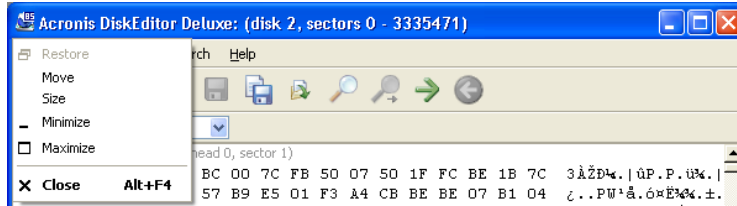
You may maximize a window or return it to the default condition by double clicking its title bar.

The Window menu and tool bars are located under the window title bar.

Window control elements are located under the menu and tool bars.

## System menu

A mouse click on the  button, located in the left top corner of Acronis DiskEditor main window brings the system menu on the screen.



The system menu in Acronis DiskEditor windows contains elements with the following values:

- Restore – restore window’s default settings,
- Move – move the window,
- Size – change window size,
- Minimize – minimize window to a taskbar icon on Windows desktop,

Clicking the  button does the same.

- Maximize – maximize window to the full screen.

Clicking the  button does the same.


Clicking the  button restores window default settings.



- Close – close window.


Clicking the  button does the same.

## Minimize, maximize/restore and close window buttons

On the right side of the window title bar there are minimize, maximize/restore (on sizable windows only) and close window buttons.

A click on the  minimize button minimizes the window to a taskbar icon on Windows desktop.

A click on the  maximize button maximizes the window to the full screen and the button picture becomes: . Mouse-clicking this button restores window’s default settings.

A click on the  close button closes the window.

## 1.5.4 Dialog, and its control elements

A dialog in Acronis DiskEditor is the window to enter disk search parameters, such as parameters for moving disk sectors, for saving a sector to a file, or reading a file.

A user-dialog interaction is performed with the help of **control elements**. Acronis DiskEditor features the following dialog window control elements:

- Checkboxes,
- Input fields,
- Drop-down lists,
- Buttons.

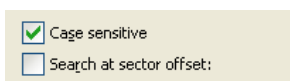
You may access control elements both with the keyboard and the mouse.

All control elements in a dialog are organized into single circular list. One of the control elements is current (has **input focus**) and responds to the keyboard. Input focus can be cyclically moved along the list – forward with **TAB** key and back with the **SHIFT+TAB** combination. Arrow buttons also allow you to move input focus as well, if they are not used in the current control element for other purposes.

Control elements may be disabled. In this case they are grayed-out and cannot be accessed either with mouse or keyboard.

### Checkbox

A checkbox is a control element with two conditions: «on» and «off». In «off» condition it is an empty square, and in the «on» condition it is a marked square. A text description is on the right of the checkbox. Setting a checkbox is performed by mouse-clicking a square or a text description or pressing the **SPACE** key when the checkbox has focus.



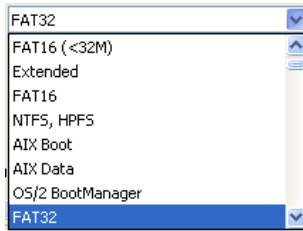
A text description of a checkbox may contain an underlined letter, in this case pressing a key with this letter (if it's not used for other purposes in the given control element), or pressing this key simultaneously with **ALT** button will result in moving input focus to this checkbox and to switching its condition.

## Drop-down list

A drop-down list control element is a field with an arrow button. Clicking allows you to open the list and select a necessary value.

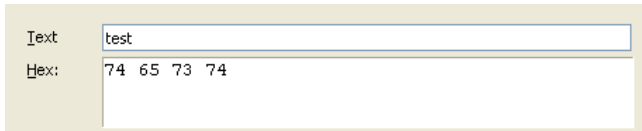


If a list is very long then a scrollbar with arrows appears in its right part. Clicking allows you to move along the list.



## Input field

An input field is intended for viewing and editing information, and is represented as a text line.



You may edit a text line only with the keyboard. Pressing mouse buttons results in moving input focus to the given input field. When the input field gets the input focus and a text line in it becomes selected, you may type something in and, the previous contents of the field will be erased or modified. You may press **LEFT** or **RIGHT** keys to deselect field contents and then edit it. The current editing position is marked by a rectangle cursor. You can use the mouse and the following keys while editing:

- **HOME** – moves the cursor to the beginning of the line;
- **END** – moves the cursor to the end of the line;
- **BACKSPACE** – deletes a symbol to the left of the cursor;
- **DEL** – deletes a symbol behind the cursor.

Input fields in Acronis DiskEditor are used for entering both symbol lines and hexadecimal values. Hexadecimal values are separated by spaces (see above figure with a **Hex** field).

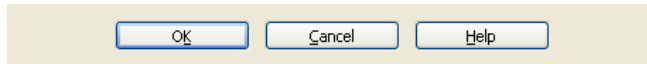


## Button and default button

A button is one of the most frequently used dialog control elements. The main action of a button is to press it.

Pressing the button can be performed both with mouse (simply clicking it), and with keyboard (pressing **ENTER** or **SPACE** when the button has input focus, or press a corresponding letter key in combination with the **ALT** key).

There may also be a default button in a dialog window. It differs from usual buttons in that double-clicking it or pressing the **ENTER** key, when another dialog control element has input focus (if these are not used for other purposes in this control element), results in pressing the default button. A default button is outlined or highlighted unlike other ones.

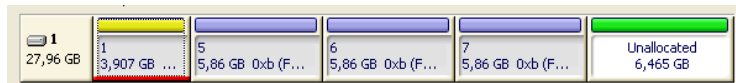


## 1.5.5 Informational elements

Besides its standard control elements, DiskEditor program has elements inform users about separate parameters of hard disks structure and actions to be performed.

### Parameters of hard disks and their partitions

The figure below shows hard disk parameters as you see them working with the DiskEditor program.

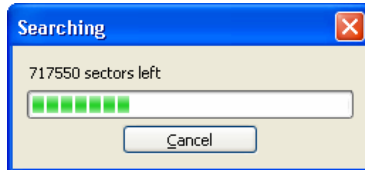


You can see the structure of hard disk #1 that has a 27.96 Gb capacity. The disk has a primary partition (marked yellow; color notations are described in the bottom of the figure) with the FAT32 file system.

The structure representation given is used in the **Open** window that allows you to select a partition or a disk to work with it.

## Progress bar

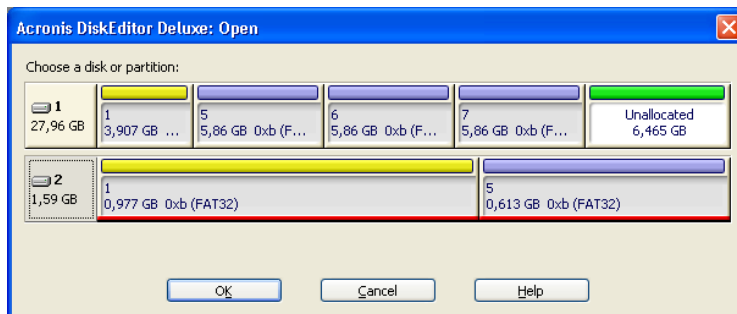
A progress bar is a pure informational element of the dialog (not a control element). It is usually used to display the current progress of any long process (for example, searching the disk for a text line).



## 2. Beginning to work with the program

### 2.1 Partition or disk selection window

Working with the Acronis DiskEditor program begins with the **Open** dialog window (Figure 1) that users see after clicking the Acronis DiskEditor Deluxe line in the program group of Programs desktop start menu.

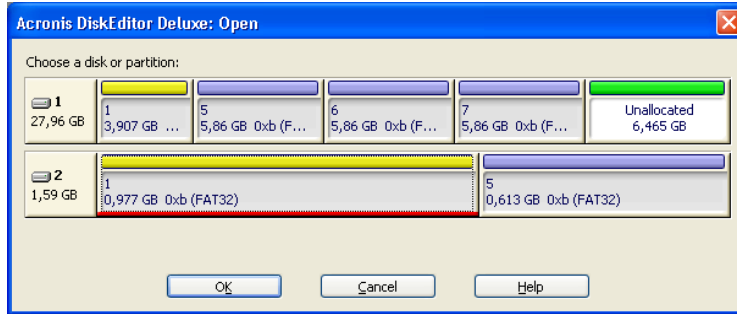


**Figure 1. The window of partition or disk selection: the entire disk**

After loading into memory, the program determines the number of disks attached to the computer and their partitions. The figure above shows that the computer disk subsystem consists of two hard disks: 1 – 27.96 Gb and 2 – 1.59 Gb divided into two partitions – a primary partition of 0.977 Gb capacity and a logical partition of 0.613 Gb capacity. Both disks have FAT32 partitions.

By default the program automatically marks the first hard disk for working. The figure shows the first computer hard disk and all its partitions are marked with a wide red line at the bottom of the rectangle representing the disk and partitions. But you may select any other partition – for this purpose click the left mouse button on a corresponding rectangle.

Below (Figure 2) only the primary partition of the first disk is selected for working. You can select the entire second disk for working in the same way. For this purpose mouse-click the rectangle representing the disk (with a hard disk icon). You may not however select more than one disk partition for working.



**Figure 2. The window of partition or disk selection: selecting the primary partition**

Clicking **OK** button, will take you to the main window of Acronis DiskEditor program and all its functionality.

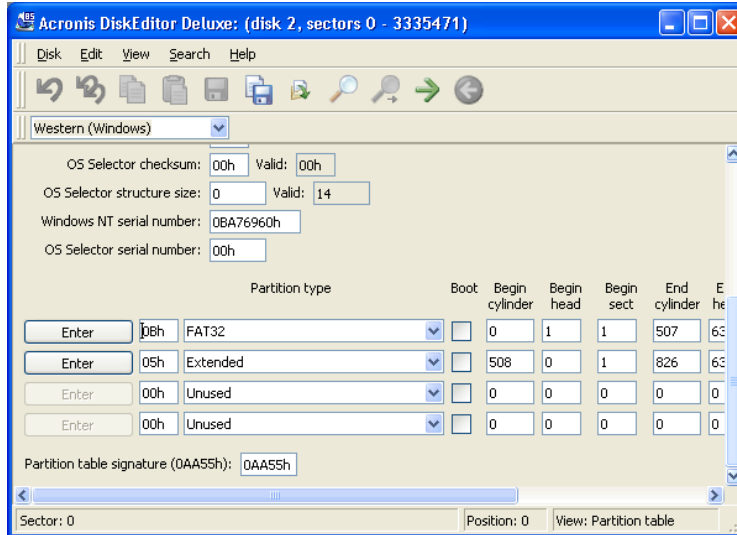


The difference between selecting the entire disk or a partition only is the following: if you select the entire disk you will be able to view and edit data storage structures of the **entire disk** – Partition table, File Allocation Table of all disk partitions, the root folder, data area. If you select a partition only, you will be able to view and edit only data storage structures of **this partition** – File Allocation Table, the root folder and data area of the selected partition.

## 2.2 Main window of the program

The main window shows numbers of disk and sector where internal program cursor is positioned usually contains information and the work of various modes.

Below the main window of the program is the view as Partition table mode (Figure 3).

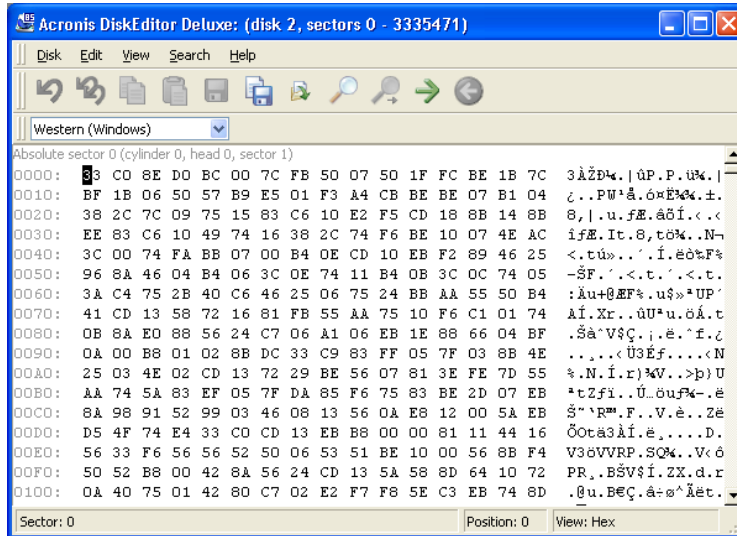


**Figure 3. The main window in the view as Partition table mode**

To make the information contained in this field sensible, the current sector is the MBR or the extended Partition table.

You can move along fields within the limits of a sector with the help of the mouse by selecting the necessary field or with the help of **UP**, **DOWN**, **LEFT**, **RIGHT** keys. Switching to the previous or next sector is performed with the help of **PAGEUP** and **PAGEDOWN** keys.

The next figure shows the Partition table as a single dump (hexadecimal byte values are on the left, and the corresponding characters are on the right; Figure 4).



**Figure 4. The main window in the view as Hex mode**

The current editing position is shown by the cursor that can be either in the left, or right part of the window. You can switch from one part to another with the help of the mouse clicking the necessary position or with the help of **TAB** key. The cursor can be moved with the help of the mouse as well by clicking the necessary position on the scrollbar on the right window border, or with the help of **UP, DOWN, LEFT, RIGHT** keys.



In this section of the Guide all examples of DiskEditor's main window view modes are given for demonstration purposes.

The main menu of Acronis DiskEditor contains the following items:

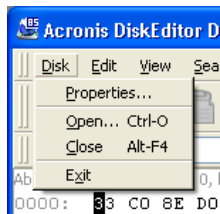
- **Disk** – allows you to open the window of a hard disk partition or to select a hard disk for editing;
- **Edit** – allows you to operate with blocks of selected hard disk partitions; also allows you to save contents of a disk sector after editing;
- **View** – allows you to select a view mode for more convenient presentation of data;
- **Search** – allows you to search a partition (or a hard disk) for any line (or any sequence of characters) and go to a certain disk sector according to its absolute offset;

- **Help** – allows you to get help concerning the editor window or about the program’s developer and version.

Let us now examine the menu of the DiskEditor program in detail.

### 2.3 Disk menu

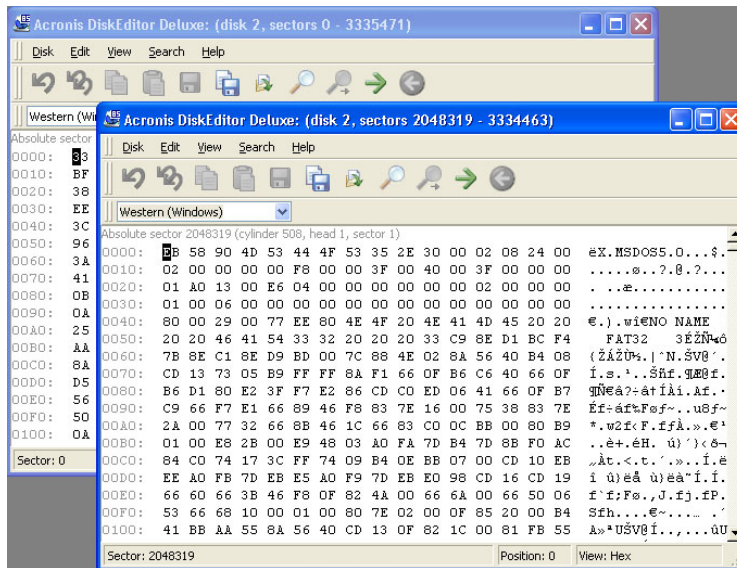
The disk editor features multiple windows: from **Disk** menu of DiskEditor (Figure 5) you can open several main windows, each of them showing various hard disk sectors.



**Figure 5. Disk menu**

You can switch between various Acronis DiskEditor windows with the help of the mouse, clicking the necessary window, or consistently advancing windows with the help of **ALT+TAB** key combination.

After Acronis DiskEditor starts you see a window of partition for hard disk selection (see above). Having selected a partition or a disk and having pressed **OK** button, you are taken the main window of the program. Mouse-clicking **Disk** menu and selecting the **Open** line will take you to the partition selection window again. You may select another disk partition for working. After pressing **OK** button you will see one more main window that may display contents of the same or another partition (Figure 6).

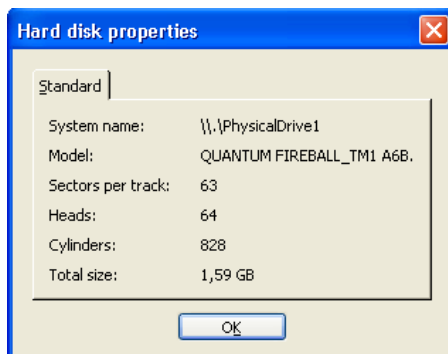


**Figure 6. Acronis DiskEditor as a multiwindow editor**

It is possible to open the same partition in another window. That said, it is possible to work with several windows simultaneously, comparing and editing displayed information as needed.

Selecting **Close** item of **Disk** menu will result in the closure of the current Acronis DiskEditor window, pressing **ALT+F4** key combination leads to the same result. Selecting **Disk** → **Exit** menu line or pressing **ESCAPE** key leads to the same result as well.

Selecting **Properties** item in **Disk** menu allows you to get access to the window with main properties of the computer hard disk selected for working (or of selected partition of the given disk; Figure 7).



**Figure 7. The Hard Disk Properties window**

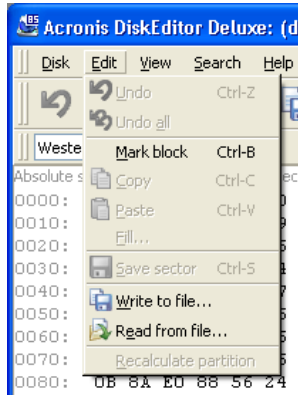
The main hard disk properties displayed in this window are:



- System name – disk’s system name,
- Model – hard disk model,
- Sectors per track – number of sectors per a track,
- Heads – number of heads,
- Cylinders – number of cylinders,
- Total size – hard disk capacity,
- Interface – interface type (ATA, SCSI),
- Controller – controller type.

## 2.4 Edit menu

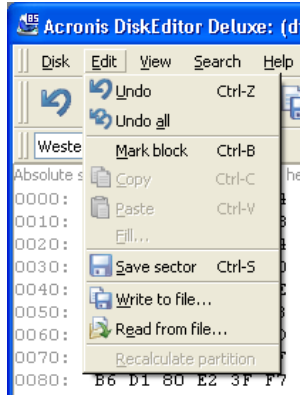
**Edit** menu of DiskEditor’s main window allows you to access the main operations with blocks of given hard disk sectors (Figure 8). You can edit hard disk data directly in the fields of any view mode (see the **View** menu). You can operate with data blocks using **Edit** menu in the view as Hex mode.



**Figure 8. Edit menu**

Any view mode allows you to select blocks with the help of the mouse by clicking and holding its left button, or with the help of the keyboard advancing the window with **SHIFT** key pressed.

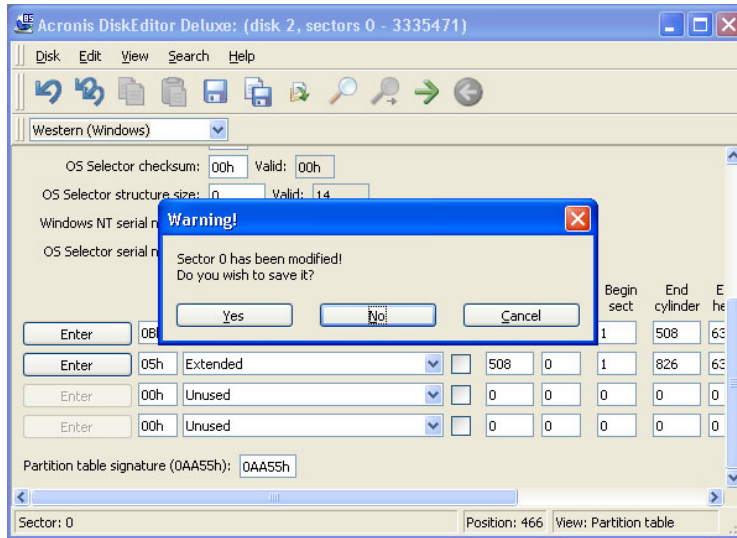
Simply changing data in the hexadecimal or character area, in the view as Hex mode of the main window or in any fields of other view modes, does not lead to changes in the given disk sector. Having performed such changes, you will see that the **Save sector** item (Figure 9) in **Edit** menu will become enabled as well as the appropriate toolbar button.



**Figure 9. Edit menu: Save sector operation**

This feature allows you to save the results of your sector editing or to reject it.

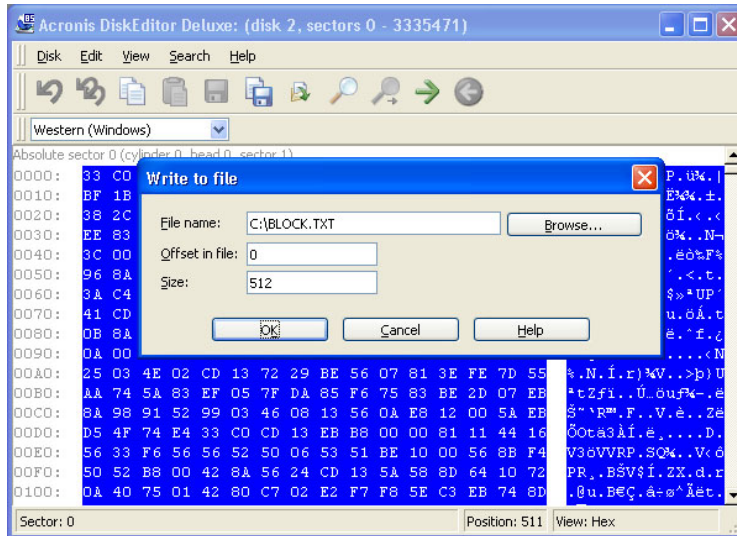
If you have made changes in a hard disk sector and decided to exit the editor without saving them, you will see a warning about saving your changes (Figure 10).



**Figure 10. Saving modified sector**

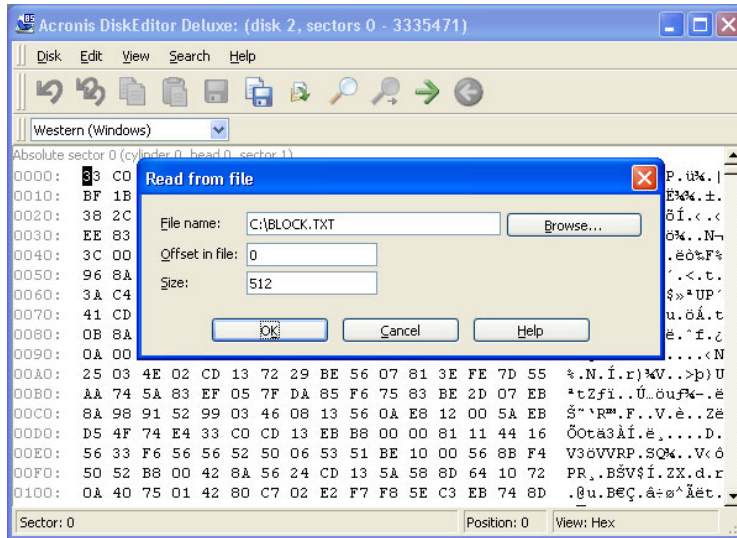
**Write to file...** and **Read from file...** menu items allows you to save the selected block to a file or to read the block from a file and save it to a disk sector.

The selection of **Write to file...** item results in opening the **Write to file** window (Figure 11; the saved block should be preliminary selected). In this window enter the file name and its path, or locate a file by clicking the **BROWSE...** button. The file size will be calculated automatically. To save a file click the **OK** button.



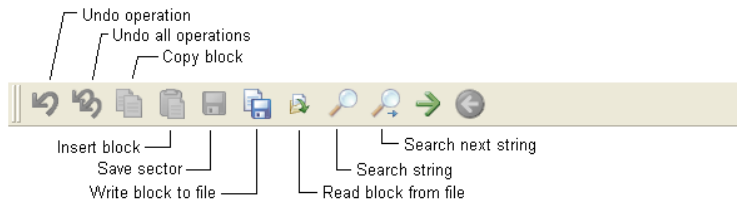
**Figure 11. The Write to file window**

To read a block from a file, and insert it into a disk sector, you must place the cursor on the necessary sector byte and then select **Read from file...** menu item. The selection of this menu item will result in opening the **Read from file** window (Figure 12). In this window enter the file name and its path or locate a file by clicking the **BROWSE...** button. To insert file contents to sector from the current cursor position (or considering offset in the file) click the **OK** button.



**Figure 12. The Read from file window**

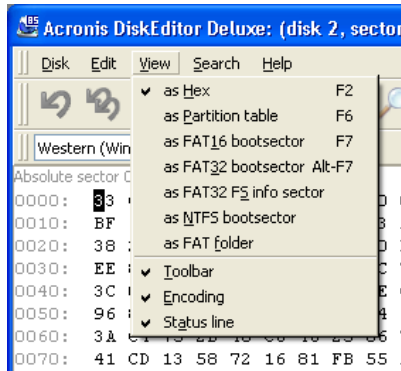
All main operations listed above – copying and inserting blocks, saving sectors, writing to files and reading from them, as well as undoing last or all operations – can be performed by clicking toolbar buttons (Figure 13).



**Figure 13. Toolbar buttons for block operations**

## 2.5 View menu

Information in the Acronis DiskEditor window can be viewed and edited in several different modes. You can select the appropriate view mode with the help of **View** menu (Figure 14).



**Figure 14. View menu**

The editor offers 7 view modes:

- As Hex,
- As Partition table,
- As FAT16 bootsector,
- As FAT32 bootsector,
- As FAT32 FS info sector,
- As NTFS bootsector,
- As FAT folder.

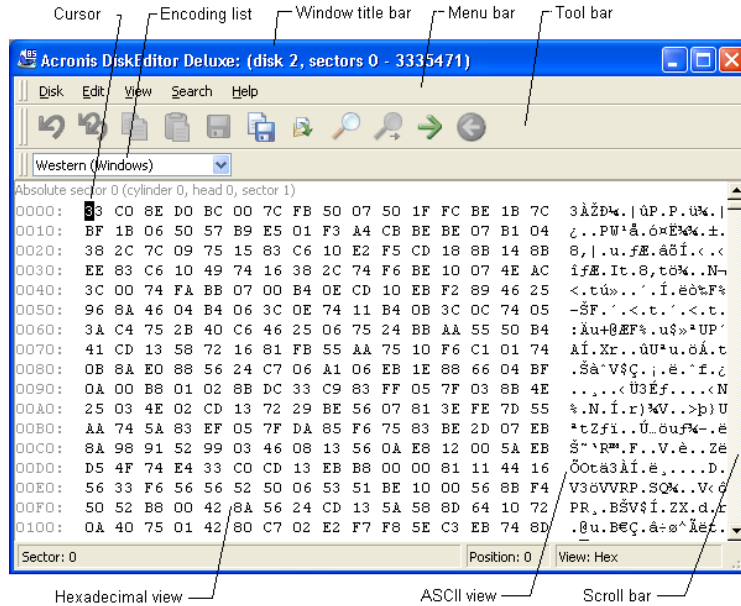
With the help of this menu you can also remove or display separate elements of the main window – toolbar, list of encodings, status line.

As discussed, in the **hexadecimal** mode the editing area is represented as a single **dump** (hexadecimal byte values are shown on the left, and the appropriate characters – on the right; Figure 15).

In the section **2.2 «Main window of the program»** of this Guide, we have described the way to advance sectors and sector bytes in the view as Hex mode. Here we will discuss the view mode as the main operation mode of the DiskEditor main window. This mode is universal, while other modes are suitable for working through important disk structures. You can work only in this mode with the data area that occupies the major part of a disk. If you are sure about

the values of each record byte, for example of the Root folder, you can edit it without the help of the view as Folder mode.

The figure below shows the elements of the main window and the representation of hard disk sector contents in the hexadecimal and ASCII formats (Figure 15). Note that the status line contains the absolute offset of the current disk sector; the same offset (with cylinder, head and sector numbers) is shown in the line dividing contents of adjacent sectors.

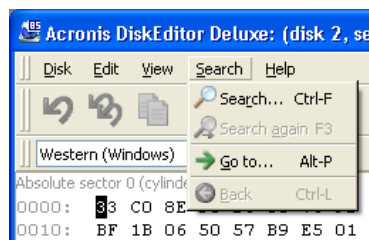


**Figure 15. Elements of the main window in the view as Hex mode**

The **3. «Selecting partitions (disks) for working»** is entirely devoted to features of main window view modes.

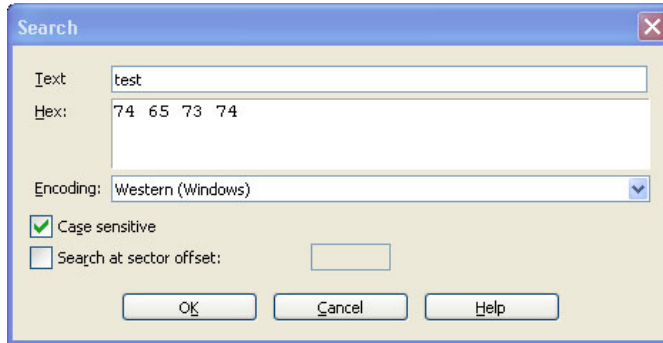
## 2.6 Search menu

**Search** menu allows you to search a hard disk for some line and to go to a disk sector according to its absolute offset (Figure 16).



**Figure 16. Search Menu**

Selecting **Search** item in the same menu will give you access to functions of searching lines in the disk being edited. (You can do the same by pressing **CTRL+F** key combination.) Search parameters can be set in the **Search** dialog window (Figure 17).



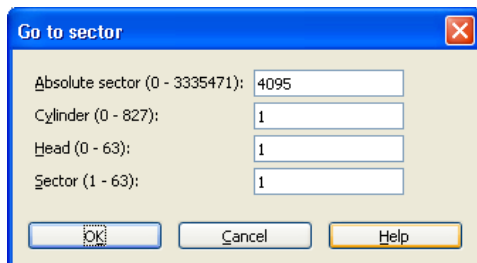
**Figure 17. The Search dialog window**

A search line can be set both as char, and numeric (hexadecimal) value. During a search you can ignore letter case, as well as search for a given line at a given offset inside the sector.

During search, disk data is interpreted according to the encoding selected. If you selected a search mode without letter case matching, not only case but also elements above characters will be ignored for Roman character sets.

After the search process is finished, the current position will be moved to where a line was found, or will remain the same if no lines were found. You can search for the next line from the current position by selecting **Find next** item in the **Search** menu or by pressing **F3** key.

You can go to the necessary sector according to its absolute offset by selecting the **Go to...** line the **Search** menu (or by pressing **ALT+P** key combination). Selecting this line opens the **Go to...** dialog window (Figure 18).



**Figure 18. The Go to sector dialog window**

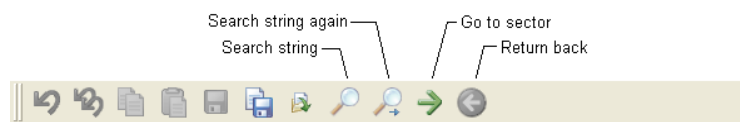
The transition is performed by entering absolute sector offset, or cylinder, head, and sector numbers. The listed parameters are bound by this expression:

$$(CYL \times HDS + HD) \times SPT + SEC - 1,$$

Where CYL, HD, SEC are numbers of cylinder, head, sector in the CHS coordinates (Cylinder – Head – Sector); HDS is the number of heads per disk, SPT is the number of heads per track.

You can return to sector from another one by selecting the **Back** item in the **Search** menu (or by pressing **CTRL+BACKSPACE** key combination).

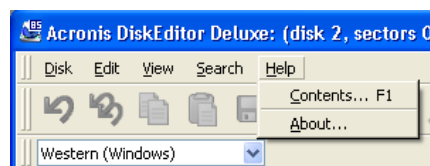
All operations listed above – searching for lines, going to sectors according to their absolute offsets, returning to starting sectors – can be performed by clicking toolbar buttons (Figure 19).



**Figure 19. Search line and Go to sector toolbar buttons**

## 2.7 Help menu

Selecting **Help** menu item will give you help about each program window as well as information about the program version and developer (Figure 20).

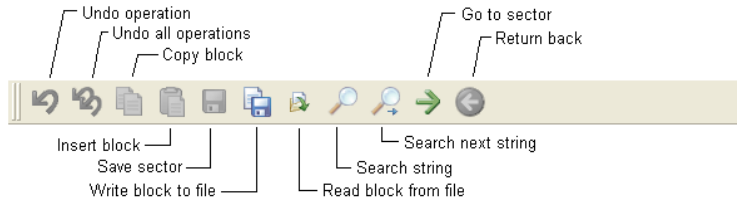


**Figure 20. Help menu**



## 2.8 Toolbar

You can also access main functions of the program with the help of the toolbar shown below (Figure 21).



**Figure 21. Toolbar buttons**

Some toolbar buttons may be disabled (such buttons are usually grayed-out). You can access most buttons to perform operations on disk sectors.

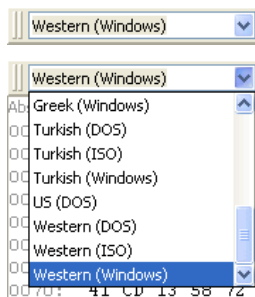
## 2.9 Controlling the program with keys

You can access almost all main functions of the program (or menu items) with the help of **F1, F2, ... F10** «hotkeys» and combinations like **CTRL+O, CTRL+F, ALT+P**, etc.

For example, the dialog window for searching partitions and/or hard disks for lines is done by pressing **CTRL+F** key combination, going to sector according its absolute offset – by **ALT+P**, switching between various main window view modes is performed by pressing **F2** (view as hex), **F6** (view as Partition table), **F7** (view as bootsector), and other keys.

## 2.10 The list of encodings

The main window of the program features a list of encodings available in DiskEditor. This list is shown closed and dropped down below (Figure 22).



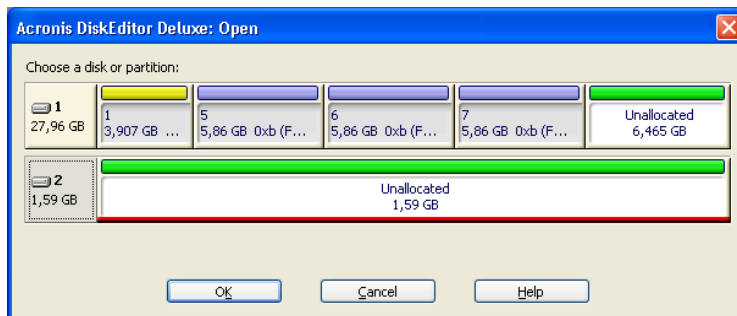
**Figure 22. The closed and dropped down list of encodings**

The list of encodings is intended for correct interpretation of hard disk sector contents. Selecting the necessary encoding, you will be able to view sector contents correctly. This is interpreted in the right part of program's main window in the hex mode.

## 3. Main window view modes

### 3.1 Selecting partitions (disks) for working

Work with Acronis DiskEditor program begins with the hard disk partition or entire disk selection window (Figure 23). The selection is performed by clicking the left mouse button on a rectangle representing a disk partition, or on a rectangle representing the entire hard disk (with an icon). After clicking the bottom of a rectangle it appears underlined red. If you clicked a hard disk rectangle then all disk partitions and a disk will be underlined.



**Figure 23. The partition (hard disk) selection window**

Having selected a disk partition (or a disk), click **OK** button to continue. You will see program's main window. (To exit the program click **CANCEL** button.)

The top part of this window features information about Acronis OS Selector program, the bottom part actually features information about the Partition table.

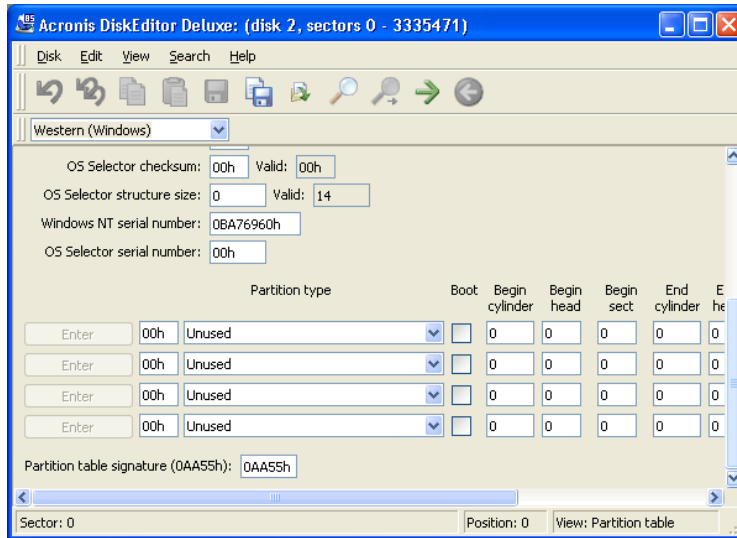


The Acronis OS Selector is a boot manager/partitioner developed by Acronis. The Acronis OS Selector allows you to install many operating systems on a hard disk (up to 100!), and to select systems at boot-up. These specific disk editor fields will not be described here.

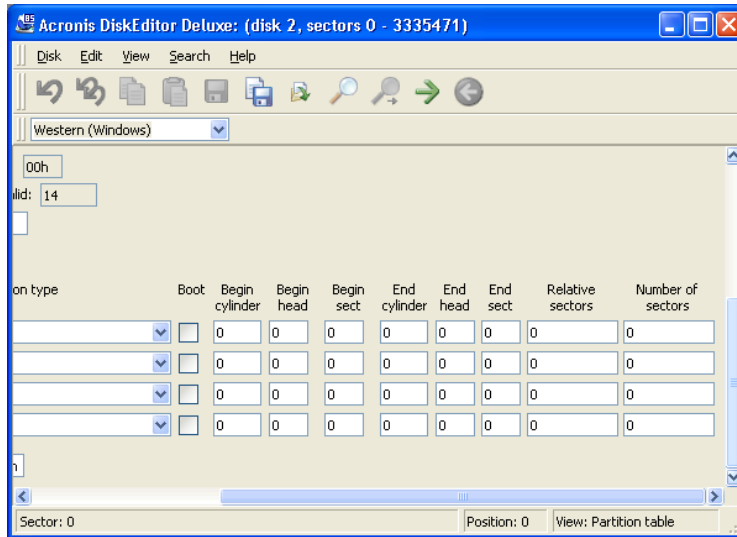
A hard disk of 1.79 Gb capacity is used as an example of how DiskEditor displays information. All disk partitions were removed and the disk was completely cleared of any information with the help of Acronis ProofEraser which zeroed all hard disk sector bytes. This is done to demonstrate how data storage structures are created on a hard disk by standard means -- for example MS Windows (or MS DOS) operating system using FDISK and/or the FORMAT utility.

### 3.2 Master Boot Record (MBR)

The main window of the program in the view as Partition table mode is shown below. As you can see the disk is empty and contains no partitions (Figure 24 – Figure 25).



**Figure 24. DiskEditor main window in the view as Partition table mode (part 1)**



**Figure 25. DiskEditor main window in the view as Partition table mode (part 2)**

On the screens above you can see that all fields for displaying information about disk partitions are empty and contain only zeroes. No hard disk sectors are allo-

cated for partitions (Figure 23) and hard disk space is completely free (Unallocated).

As a hard disk can contain only 4 primary partitions, 4 lines on the screen are allocated for information about partitions. A hard disk partition is characterized by the following parameters:

- Type, for example FAT16/32, NTFS, etc.,
- Boot patch flag – whether partition is bootable or not,
- Begin partition cylinder,
- Begin partition head,
- Begin partition sector,
- End partition cylinder,
- End partition head,
- End partition sector,
- Number of hard disk sectors allocated for partition.

Hard disk partitions are described in the Partition table in two ways simultaneously: (1) in CHS co-ordinates (Cylinder – Head – Sector; limited and obsolete variant) – coordinates of initial and end partition sectors, and (2) partition heading and its length in sectors.

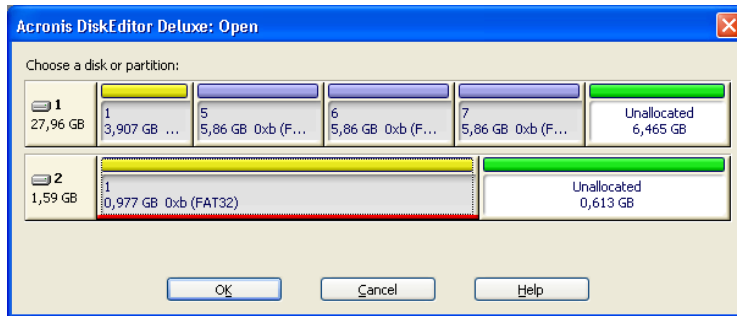
Each record of the Partition table contains a partition boot patch flag indicating to the main loader what partition to boot. The boot patch flag can be assigned to one partition only.

At the lowest (byte) level MBR structure containing the master boot record and the Partition table can be described as follows:

- Bytes 000h – 1BDh – load code of active partition's bootsector,
- Bytes 1BEh – 1CDh, 1CEh – 1DDh, 1DEh – 1EDh, 1EEh – 1FDh – partition records,
- Bytes 1FEh – 1FF – Partition Table signature equal to AA55h.

### 3.3 Creating primary partition

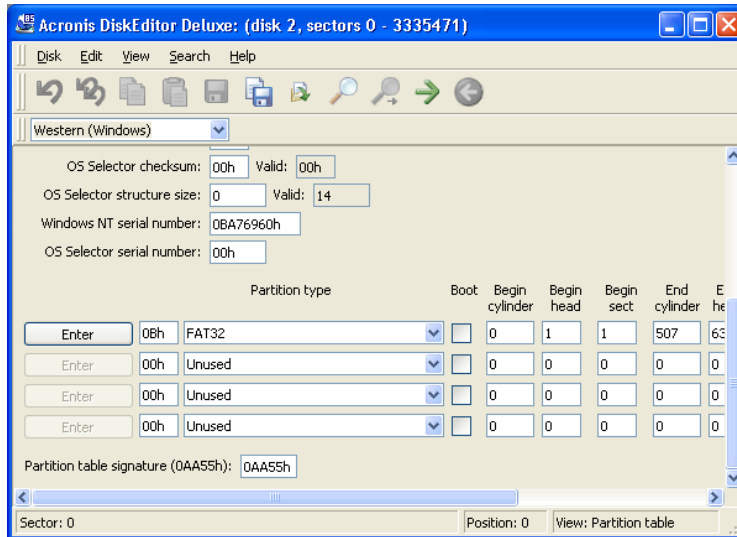
Using the FDISK utility supplied with your Windows operating system (enabling large disks support), we will create the Primary partition at a 0.97 Gb capacity (or 1000 Mb) on the selected hard disk. This will be reflected immediately in the partition selection window (Figure 26).



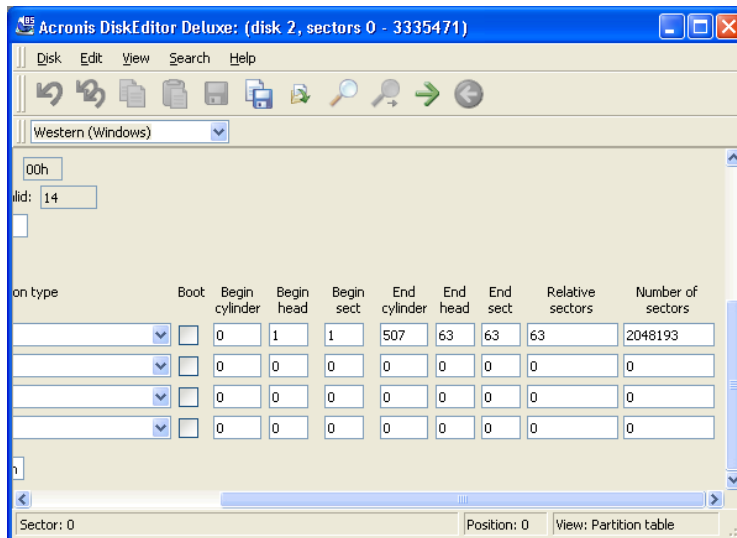
**Figure 26. The partition selection window with the primary partition created**

To see changes in the Partition table contents, select the hard disk with the newly created partition for viewing. If we select the created partition for working, the Partition table will be unavailable for viewing. The Partition table contains information about the entire hard disk. If you select a disk partition for working, DiskEditor will allow you to browse only the space allocated for that partition, and accordingly, to look at its information only.

The creation of a hard disk primary partition will immediately update the Partition table (Figure 27 – Figure 28).



**Figure 27. View as Partition table mode: information about the primary partition**



**Figure 28. View as Partition table mode: information about the primary partition (continued)**

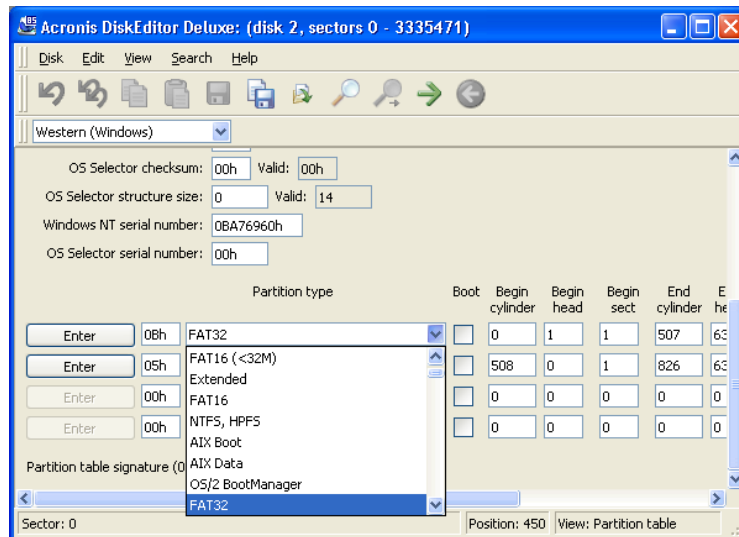
Creating 1000 Mb primary partition on the disk results in filling the 1<sup>st</sup> line of the Partition table. You can see partition parameters on the figure. Partition contains 2048193 sectors total and is of FAT32 type (or 0Bh in hex).

Please note that **ENTER** button to the left of **Partition type** field is enabled; clicking it you can quickly proceed to partition specified by the Partition table record.



You must be extremely careful when changing partition parameters. One can easily completely destroy information stored on the disk. The best way to master DiskEditor capabilities is with an old disk, without important information, that can be used for creating and destroying partitions, formatting and performing any experiments. Please do not experiment with your work disk!

To work with a hard disk you must know where and what data is stored on it. As a complete disk editing program, DiskEditor allows you to create a complete partition on a disk without the use of the FDISK utility (or any other utilities). DiskEditor is also capable of much more: while the FDISK utility allows you to create only FAT16/32 partitions, DiskEditor allows you to create partitions for almost any types known. Look closely at the **Partition type** drop-down list (Figure 29).



**Figure 29. The list of partition types supported by Acronis DiskEditor**

This list also contains a substantial group of partition types, that may seem exotic to some users.

### 3.4 Formatting a partition

We now use the FORMAT utility, also supplied with Windows operating system, to format the created primary partition with /S option that allows to transfer system files to partition:

```
FORMAT C: /S
```

(The disk can have another letter assigned, for example, D:.)



Formatting a partition with the FORMAT utility results in the creation of partition bootsector contents.

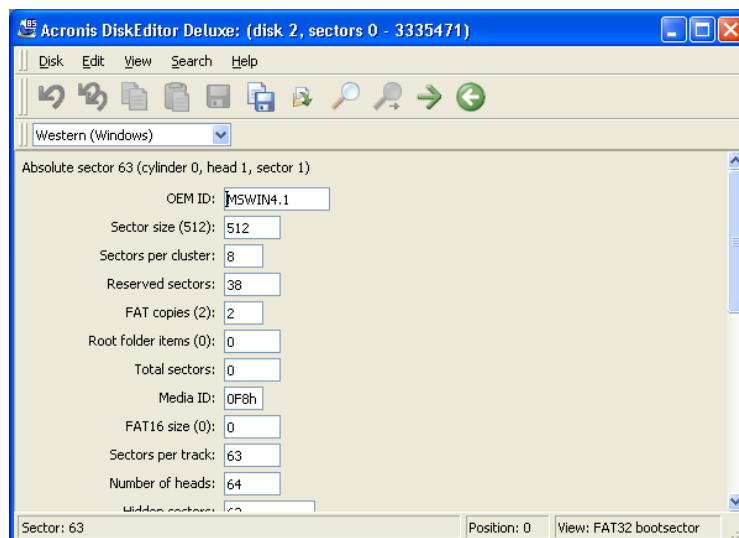
The following modes allow you to view a bootsector: as FAT16 bootsector, as FAT32 bootsector, and as NTFS bootsector. You should keep in mind that simple selection of a view mode does not automatically result in positioning DiskEditor to a partition bootsector. You have to do it manually. For this purpose you can use **Search → Go To...** menu while working with the entire hard disk and enter the sector offset from the Partition table into **Absolute sector** field in the dialog window.

Or to use **Disk → Open** menu to select a partition in the **Open** dialog window (a partition, not a disk) which bootsector you need. In this case DiskEditor will position on the 1<sup>st</sup> sector of that partition – its bootsector.

If you are in the view as Partition table mode, you can find a bootsector of any partition simply clicking **ENTER** button to the left of partition record.

Keeping in mind that all hard disk sectors have been zeroed with the help of program Acronis ProofEraser, if you want to look at a hard disk partition bootsector in the view as FAT32 bootsector mode before this partition is formatted, you would, see some random data without any relation to actual partition characteristics.

Let us assume you positioned DiskEditor on a partition bootsector and selected the view as FAT32 bootsector mode. In this case you will see a picture similar to that shown below (Figure 30; actual data entirely depends on your hard disk).



**Figure 30. The main window in the view as bootsector mode**

The main window in the view as bootsector mode has many fields. Not all fields are of equal importance: file systems are created and developed, grow old, or become obsolete. Therefore some fields used earlier may become obsolete in the future. The most important fields are:

- OEM ID – the line formed at partition formatting;
- Sector size – sector size in bytes;
- Sector per cluster – the number of sectors per cluster;
- Reserved sectors – the number of sectors occupied by the loader and reserved;
- FAT copies – the number of FAT copies; theoretically a disk can contain up to 4 FAT copies, but usually there are 2;
- Root folder item, Total sectors, FAT16 size – fields used by FAT16 file system;
- Media ID – Media (disk) identifier;
- Sectors per track – the number of sectors per track;
- Number of heads – the number of heads;
- Big total sectors – the number of disk sectors available to partition file system (for partitions > 32 Mb);
- FAT size – the size of File Allocation Table in disk sectors;
- Active FAT – four low bits of this field contain the number of active FAT copy used for reading and writing if copy duplication is switched off;
- File System Version – always 0 for FAT32;
- First Cluster of Root – the first cluster of the root folder (FAT32 stores the root folder as all other folders and files; FAT16 used a special disk area for it);
- FAT32 FS Info sector – the relative number of information sectors of FAT32 file system;
- Backup sector – the sector containing a copy of bootsector;
- Hard Drive – disk's BIOS number;
- Extension signature – the signature of extended bootsector;
- Serial number – volume serial number (set at formatting);
- Volume label – a char line;
- File system name – file system char identifier (FAT32);
- Bootsector signature – a bootsector signature.

Some bootsector fields are reserved for further use.

In the above section **3.2 «Master Boot Record (MBR)»** we have described this structure containing, in particular, primary loader program. A hard disk partition also begins with the sector containing the operating system loader program. Primary loader and partition loader are bound as follows: the purpose of primary loader is to locate the first (boot) sector of active partition, to load its code to memory, and to pass controls to it. Further actions concerning operating system loading are performed by the active partition loader.

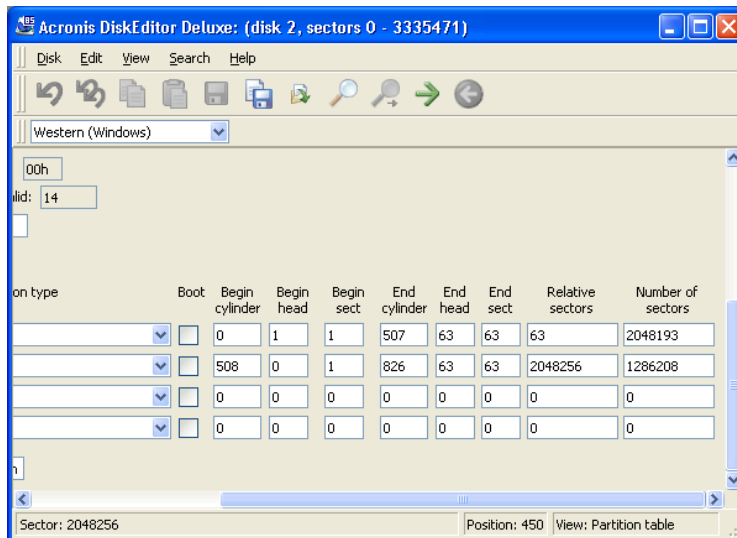


To view instructions contained in the primary loader (partition loader) code, you can use DEBUG utility supplied both with DOS and Windows. DEBUG allows to read any disk sector and disassemble its code, i.e. to translate the machine code to assembler language. Your understanding of these instructions will depend on your knowledge of assembler language.

You can navigate fields within the limits of a sector with the help of the mouse by simply clicking the necessary field, or with the help of **UP**, **DOWN**, **LEFT**, **RIGHT** keys. You can switch to previous or next sector with the help of **PAGEUP** and **PAGEDOWN** keys.

### 3.5 Creating a logical partition

Creation of logical partition results in filling the second line of the Partition table (Figure 31).



**Figure 31. DiskEditor main window in the view as Partition table mode with the created logical partition**

The partition is marked Extended LBA or 0Fh in hex. Externally it is characterized by the same parameters, as the Primary disk partition, but its organization and purpose differ. The primary partition contains a single logical disk. Extended DOS/Windows partition allows you to organize any number of logical disks.



Again, this User's Guide describes DiskEditor capabilities used on a FAT32 file system.

The first sector of the Extended partition is similar to the first sector of the Primary hard disk partition or MBR, but unlike the latter it does not include loaders and contains the Extended Partition Record, or EPR.

The Primary partition loader loads the operating system located on this disk (if any). However, it is impossible to load Windows 9x/Me from a logical disk of the Extended partition without special set-up (see: section B.3 «Primary and Logical Partitions»).

The first record of the Extended Partition Record specifies the first logical disk of the partition.

If the first logical disk does not occupy an entire extended partition, the second EPR record specifies position of sector with the next Extended Partition Record. Third and fourth EPRs are not used.

While MBR Partition table (of a physical disk) can address to 4 partitions at once, the Extended Partition Record addresses to a chain of partitions that ends with a partition, which the second record of EPR does not address.



Please note that the Extended Partition Record addresses not from the beginning of a physical hard disk (MBR sector), but from the beginning of the Extended partition (EPR sector).

The chain of extended records is continuous and non-branching. From the angle of a physical disk, all logical disks are located in the area described in the Partition table as the Extended partition.

Each logical disk from the Extended partition has the same structure as the Primary disk partition: it begins with a bootsector (but the logical disk loader is never used), and has the Partition table (Extended).

### **3.6 File Allocation Table (FAT) and Root folder (Root)**

The structure of a partition with the FAT file system was described above. A partition begins with a bootsector containing a loader and a table describing file system parameters.

Bootsector is followed by one or several File Allocation Tables (FAT), the Root folder, and the Data area divided into clusters and used for storing folders and files. (There may be a number of reserved sectors between a bootsector and the first FAT copy.)

### **A file as a chain of clusters**

As above (see **1.1.1 «Files, folders, file systems»**), a file on a disk is stored as a chain of **clusters**. A cluster consists of one or more sectors, the number of sectors per cluster depends on partition size and equals a degree of 2 (1, 2, 4, 8, 16, etc.). Each cluster has its number with numeration beginning with 2.

The File Allocation Table (FAT) describes the order of folders and files in clusters. Each cluster corresponds to a FAT **element** (a table cell).

FAT elements can indicate of the following values:

- 0 – free cluster,
- 2 – the number of the next element in a cluster chain (0FEFh – for FAT12, 0FFEFh – for FAT16, 0FFFFFFEh – for FAT32),
- 0FFF0h – 0FFF6h – reserved values,
- 0FFF7h – bad cluster,
- FFF8h – FFFFh – last cluster in a chain.

To **read** a file it is necessary to find its record in a folder by its name and to read the number of the 1<sup>st</sup> file cluster. The first cluster represents the beginning of a file. Then it is necessary to read a **FAT element** corresponding to the first cluster of a file. If this element contains a «last in chain» label, there's no need to continue searching: this file is simply located in a single cluster. If a cluster is not the last, it contains the number of the next cluster. Contents of the next cluster should be read after the first. When the last cluster in a chain is found, it is necessary to cut off the remaining bytes of the cluster if this file does not occupy all of it. Extra bytes are cut off according to the file length stored in the folder record.

To **write** a file, the operating system has to execute the following sequence of actions: create a file description in a free folder element, search for a free FAT element, and place the reference to it to a folder record; fill the first cluster described by the FAT element found; and place the number of the next cluster or an attribute of the last cluster in a chain to this FAT element.

The operating system operates in this way to add next clusters to chains according to the increasing number. It's clear that consistently located clusters can be addressed much faster than if they were randomly spread along the disk. Clus-

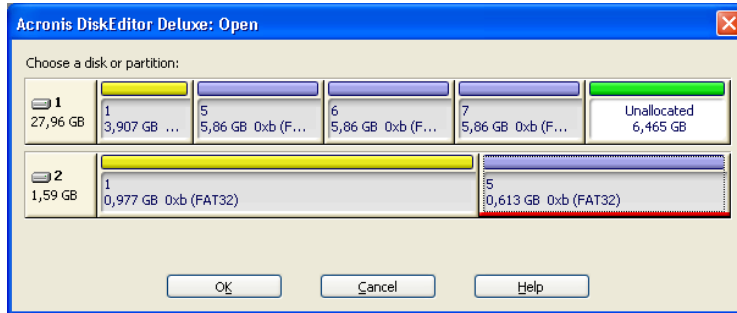
ters that are already occupied or clusters marked in FAT as defective are left out.

File **deletion** (from a Recycle bin that appeared in Windows 95) declares all file clusters free (along with zeroing of corresponding FAT elements) and replacing the first symbol of file name in the folder record by 0E5h value. Operating system ignores such elements during a usual folder search.

### A folder in DiskEditor main window

Let us now create a very simple file on the logical partition of our hard disk. It can be done with the help of Windows Notebook application. Name this file test.txt, fill it with a single word «test», and save it the partition root folder.

Now run DiskEditor and mouse-click the logical disk partition in the partition selection window (Figure 32).

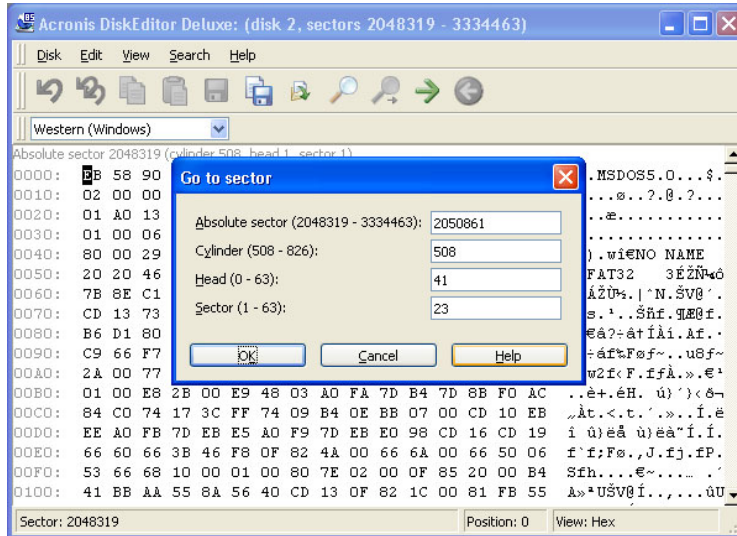


**Figure 32. Selecting the logical partition for working**

DiskEditor will automatically position on the bootsector of this partition and the main window will open in the view as FAT32 bootsector mode by default.

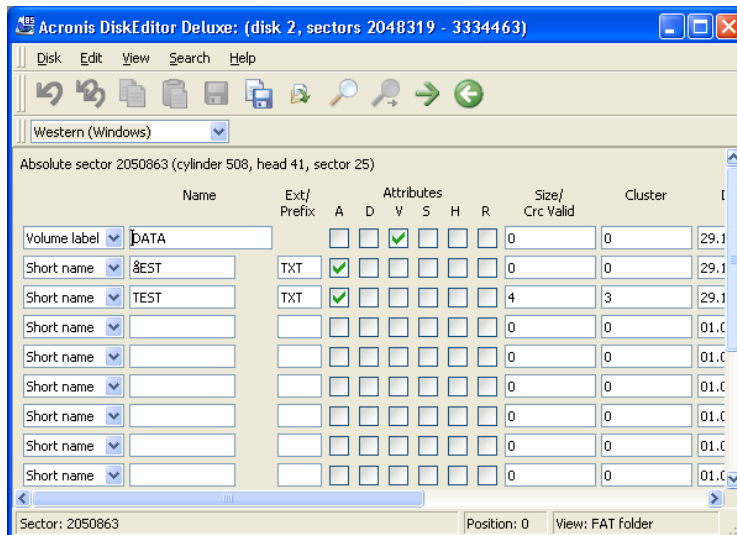
A bootsector of the logical partition is the sector with 2048319 absolute offset (your values may be different). The File Allocation Table of the logical partition has is 1255 sectors long. 32 sectors are reserved so,  $2048319 + 2 * 1255 + 32 = 2050861$ , i.e. the root folder is, most likely, located at sector 2050861.

Select **Go to sector...** item in the **Search** menu (or press **ALT+P** key combination) and in the **Absolute sector** field of the **Go to sector...** dialog window enter 2050861 (Figure 33) and click **OK** button.



**Figure 33. Transition to the root folder of the logical disk**

DiskEditor will position on the root folder area. Switch the main window to the view as FAT folder mode as shown below (Figure 34).



**Figure 34. The root folder area of the logical disk: the main window in the view as FAT folder mode**

### Files with short names

As discussed, a file name and other attributes are registered in a folder record.

A folder record has the following structure, or main bytes of a folder record indicate the following values:

Offset bytes	Length bytes	Purpose
0	8	File name up to 8 symbols.
8	3	File extension (type) up to 3 symbols.
0Bh	1	File Attributes: Bit 0 – Read-Only (can't update or delete); Bit 1 – Hidden (H), hidden file; Bit 2 – System (Sys), system file; Bit 3 – Volume label; Bit 4 – folder attribute (Directory Entry); Bit 5 – Archive (A), file was not archived.
0Ch–15h	10	Reserved.
16h	2	Last changed (created) time: Bit 0-4 – pairs of seconds (0-29), Bit 5-10 – minutes (0-59), Bit 11-15 – hours (0-23).
18h	2	Last changed (created) date: Bit 0-4 – day (0-31), Bit 5-8 – month (1-12), Bit 9-15 – year from 1980 (0-119).
1Ah	2	Primary cluster number.
1C	4	File size in bytes.



The figure above shows values already interpreted by the DiskEditor program, these fields have the following values:

Name	File name.
Ext/Prefix	File name extension or a number of a long name fragment.
A, D, V, S, H, R	Represent Archive, Directory Entry Volume Label, System, Hidden, Read-Only file attributes.
Size/Crc Valid	File size or a long name checksum.
Cluster	Number of the first file cluster.
Time	File creation time.
Create date	File creation date.
Reserved	Reserved field.



Root folders in FAT16 file system have fixed sizes, locations, and contained limited number of elements. FAT32 represents root folders like usual (sub) folders by a cluster chain of any length. The number of entries in it – files and subfolders – is unlimited. The number of folder entries is unlimited as well, though it's indirectly limited by the maximum file (folder) name length counted off from the root folder.

### Files with long names

Everything described above represents the classic way of describing files with names in «8.3» format in FAT folder. But all Windows version after Windows 95, support long file names (up to 255 chars), allowing almost any chars in Unicode encoding. This is handled by simply allocating not one, but several adjacent 32-byte elements for each file in a folder. The number of elements used for a long name depends on its length. The first several elements contain a long name, followed by a usual element, with the structure provided for a short name containing an uppercase file name. Elements with parts in this long name are numbered one after another. The 01h element is located just before the element with a short name. If one element with a long name fragment is insufficient, another 02h element is used, etc. The number of the last element is increased by 40h. If a long name takes one element, its number will be 41h.

A folder record with a fragment of a long name has the following structure:

Offset, bytes	Length, bytes	Purpose
0	1	Element's number.
1	10	Name (or its fragment).
0Bh	1	File attributes (always 0Fh).
0Ch	1	Reserved.

0Dh	1	Check code (calculated from short name).
0Eh	12	Name (continued).
1A	2	Number of the first cluster (0)
1Ch	4	Name (continued).

The above scheme can be viewed in Acronis DiskEditor program in the view as FAT folder mode. As you can see above, test.txt corresponds to two folder elements. The **lower line** indicating Short Name value in the drop-down list on the left and the uppercase name (TEST.TXT), and the **upper** one indicating Long Name value in the list and 041h in the Ext/Prefix field indicating that a file's long name took one additional folder record.

### 3.7 Folders and subfolders

Section **3.6 «File Allocation Table (FAT) and Root folder (Root)»** described structure of the root folder in FAT32 file system that contains file information: names, attributes, and some other.

A root folder may contain records corresponding to **subfolders**. A root folder together with subfolders form the folder tree of the file system.

FAT represents a subfolder similar to a file described by a **parent** folder record, and consists of a set of the same folder elements as a root folder.

The very first element of a subfolder is a self-reference under «.» pseudonym. This reference provides the number of the first cluster of the given folder. It is followed by an element referring to the parent folder under «..» pseudonym. This reference allows to find the beginning of the parent folder. At that cluster, with zero number, is specified the root folder.

## Conclusion

Acronis DiskEditor is a professional-grade program, that allows you to access all data storage structures on your computer's hard disk – Partition table, bootsectors, and File Allocation Tables of partitions, folders and data areas (files).

This User's Guide outlines Acronis DiskEditor's representation of data storage structures and contains a description of the program's main interface elements. All menus, toolbar buttons, dialog windows and control elements were described. Capabilities of DiskEditor were described on the example of FAT32 file system.

DiskEditor has special patterns for each hard disk structure – main window view modes that allow you to correctly interpret not only separate bytes of hard disk partitions, but in many cases even separate bits, facilitating work in these structures.

All fields of the DiskEditor program were shown with byte structure within hard disk partitions – all details of DiskEditor were examined.

Sound knowledge of disk data storage features is necessary for effective work with the program. User error can cause structure of partitions, folders, and files to be scrambled leading to a condition that the operating system or applications will stop loading, or programs will stop working with files created with them.

However, it is impossible to describe all features of file systems in just one User's Guide. Therefore many subjects have been omitted, such as the obsolete but still used FAT12 and FAT16 file systems, as well as much more complex NTFS file system.

The potential application range of DiskEditor is correction of file structure imperfections: bootsector recovery, location of lost clusters and cross-linked cluster chains, folder structure defects and fighting computer viruses.

The most important thing is to master Acronis DiskEditor capabilities and file system organization details. You will then be able to deal with any imperfections in your hard disk's operation and restore them to a usable state should they become corrupted.

# Appendix A. General Information.

## Hard Disks

The Appendices below provide you with extra information on the hard disk organization, how information is stored on disks, how disks should be installed in the computer and plugged into motherboard, configuring disks with BIOS, partitions, file systems, and how operating systems interact with disks.

### A.1 Hard Disk Organization

All hard disks, or hard disk drives, have basically the same structure, however diverse they are in size. Inside the case there are several disks with magnetic coating set on a single axis (spindle). A special motor provides the necessary rotation speed to the spindle, e.g. 5400 rpm, 7200 rpm, or 10000 rpm.

Information on disks resides on concentric **tracks**. Each track has its number. The outermost track is number 0, and the numbers grow inwards.

Each of the tracks is divided into **sectors** that contain minimal information blocks that can be written to disk or read from it. Sectors also have numbers. On every disk there is a marker that indicates the beginning of sector enumeration. The sector that is the closest to this marker is number 1.

Usually sector size is 571 bytes. At the beginning of a sector there is a header (prefix portion) that marks the beginning of the sector and its number. At the end of a sector, in the suffix portion, there is the checksum that is used to check data integrity. Data area between the prefix and suffix portions is 512 bytes large.

Both upper and lower sides of each disk on the spindle are used to store data. All tracks that have the same number on all the surfaces of all disks comprise a **cylinder**. For each work surface of a disk in the drive there is a **head** that enables reading and writing data to/from the disk. Heads are assembled into a block and are enumerated, starting with 0.

To perform an elementary read or write operation the head block must be positioned at the necessary cylinder. When the necessary sector (with the necessary number in the service area) of the rotating disks approaches the head, data is exchanged between the head and the electronic board of the drive.

Sector structure of a hard disk is created via **low-level formatting** during which each of the tracks of the disk is marked up.

Modern disk drives usually contain relatively few magnetic disks (1-2) to make the head block lighter and speed up access to sectors (a drive like this has 2-4 heads respectively).

There can be up to several tens of thousands of cylinders per disk. The higher the write density on the disk, the more cylinders can be created on it and the larger the capacity of the disk.

This design has many technical implementation peculiarities, but we will not discussing those here.

## A.2 Installing Hard Disks on Computers

### A.2.1 Installing a Hard Disk. General Scheme

To install a new IDE hard disk you should do the following – ***we will assume you have powered OFF your PC before you begin!***

1. Configure the new hard disk as **Slave** by properly installing jumpers on the board of its controller.
2. Open your computer and insert the new hard disk into a 3.5” or 5.25” slot with special holders. Fasten the disk with screws.
3. Plug power cable into the hard disk (4-threaded: two black, yellow, and red; there is only one way you can plug in this cable).
4. Plug the 40- or 80-thread flat data cable into sockets on the hard disk and on the motherboard (plugging rules are described below).
5. Turn your computer on and enter BIOS setup by pressing the keys that are displayed on the screen while the computer is booting.
6. Configure the installed hard disk by setting the parameters **Type, Cylinder, Heads, Sectors, and Mode** (or **Translation Mode**; these parameters are written on the hard disk case) or by using the IDE Autodetection BIOS utility to configure the disk automatically.
7. Set the boot sequence to A:; C:; CD-ROM or some other, depending on where your copy of MigrateEasy is located. If you have a boot diskette, set the diskette to be first; if it is on a CD, make the boot sequence start with CD-ROM.
8. Quit BIOS setup and save changes. MigrateEasy will automatically start after reboot.

9. Use MigrateEasy to configure hard disks by answering the Wizard's questions.
10. After finishing the work, turn off the computer, set the jumper on the disk to the **Master** position if you want to make the disk bootable (or leave it in **Slave** position if the disk is installed as additional data storage).

## A.2.2 Mounting Hard Disks in Cases

In most mini-tower PC cases hard disks are mounted in special slots and are fastened with screws.

Size of slots (their width) is defined by the size of hard disks (and other drives that can be mounted). Physical sizes of hard disks are standardized by their **form factor**. There are only a few standard form factors. For example, hard disks with form factor 3.5" are 41.6 x 101 x 146 mm large. Almost all desktop PCs are designed for hard disks with 3.5" form factor.

If there is no slot for a 3.5" hard disk in the case, it can be installed in a 5.25" slot with help of special holders.

Hard disks are secured in the slot by screws. Usually hard disks are mounted horizontally so that the electronic board faces down, but sometimes there are slots where hard disks can be mounted vertically inside the PC.

## A.2.3 Motherboard Sockets. IDE Cable. Power Cable

There are two slots on the motherboard the hard disks can be connected to: **Primary IDE** and **Secondary IDE**.

Hard disks with IDE (Integrated Drive Electronics) interface are connected to the motherboard via a 40- or 80-thread flat marked cable: one of the threads of the cable is red.

Two IDE hard disks can be connected to each of the sockets, i.e. there can be up to 4 hard disks of this type installed in the PC. (There are three plugs on each IDE cable: 2 for hard disks and 1 for the motherboard socket.)

IDE cable plugs are usually designed so that there is only one way to connect them to the sockets.

Usually one of the pinholes is filled on the cable plug, and one of the pins facing the filled hole is removed from the motherboard socket, so it becomes impossible to plug the cable in the wrong way.

In other cases there is a jut on the plug on the cable, and an indentation in the socket of the hard disk and of the motherboard. This also ensures that there is a single way to connect the hard disk and the motherboard.

In past eras this design of plugs did not exist, so there was an empirical rule: *IDE cable is connected to the hard disk socket so that the marked thread was the closest to the power cable*, i.e. the marked thread connected to pin #1 of the socket. Similar rule was used for connecting cables with the motherboard.

Incorrect connection of the cable with either the hard disk or the motherboard does not necessarily damage the electronics of the disk or the motherboard. The hard disk is simply not detected or initialized by BIOS. However, it is better to connect the IDE cable according to the rules.



There existed some models of hard disks, especially the older ones, for which incorrect connection damaged the electronics of the drive.



Here we will not attempt to describe in detail all the existing types of hard disks. Currently the most widespread hard disks are those with IDE or SCSI interfaces. Unlike IDE hard disks, there can be from 6 to 14 SCSI hard disks installed in your PC, however you need a special SCSI controller to connect them. SCSI hard disks are not usually used in personal computers (workstations), but are found mostly in servers.

Aside from an IDE cable, a 4-thread power cable is to be connected to the hard disks. There is only one way to plug in this cable, so there cannot be any mistake.

#### A.2.4 Installation Modes. Jumpers

A hard disk can be installed in a computer as **Master** or as **Slave**. The mode of hard disk installation is defined by the position of special connectors (called jumpers) on the hard disk.

The jumpers are either located on the electronic board of the hard disk or are taken out to a special socket that provides for connection of the hard disk and the motherboard.

Jumpers are usually marked. On the case of the hard disk there is usually a sticker that explains the markings. Typical markings are **DS**, **SP**, **CS**, and **PK**.

Each jumper position corresponds to one hard disk(s) installation mode, e.g.:

- **DS – Master/Factory default;**
- **SP – Slave (or No Jumper Required);**
- **CS – Cable Select for Master/Slave:** the purpose of the hard disk is determined by its physical position with respect to the motherboard;

- **PK – Jumper Parking Position:** the position where one can put the jumper if it is not necessary in the existing configuration.

The hard disk with the jumper in **Master** position is treated by the basic input/output system (BIOS) as bootable.

The jumpers on hard disks that are connected to the same cable can be in the **Cable Select for Master/Slave** position. In this case BIOS will deem as «Master» the disk that is connected to the IDE cable closer to the motherboard than the other one.



Unfortunately, hard disk markings were never standardized. You can easily find out that markings on your hard disk differ from the ones described above. Moreover, for the old types of hard disks their purpose could be defined by two jumpers instead of one. You should study the markings carefully before installing your hard disk in the computer.

It is not enough to physically connect the hard disk to the motherboard and set the jumpers properly for the hard disk to function – hard disks have to be properly configured with the motherboard BIOS.

## A.3 BIOS

Many times you have seen what happens on the screen of your computer when you turn it on. Before you see the well-known splash screen of your operating system, you see a large number of short text messages.

These messages are from the POST (Power-On Self Test) program that belongs to BIOS and is executed by the processor.

BIOS, or the Base Input/Output System, is a program that resides in the permanent memory chip (ROM or Flash BIOS) on the motherboard of your computer and is its key element. The version of BIOS that you use «knows» all the peculiarities of all the components of the motherboard: processor, memory, integrated devices. BIOS versions are provided by the manufacturers of motherboards.

Main BIOS functions are:

- testing (POST sequence) of the processor, memory, and input/output devices;
- initial configuring of all software-manageable parts of the motherboard;
- initialization of operating system (OS) booting process.

Among numerous components of the computer, initial configuring is necessary for external memory subsystem that control hard disk drives, floppy disk drives, CD-ROM drives, DVDs, and others.



### A.3.1 Setup Utility

BIOS has a built-in Setup utility for initial computer configuring. To enter the Setup utility you have to press a certain combination of keys (**DEL**, **CTRL+ALT+ESC**, **Ctrl+Esc**, or some other, depending on your BIOS) during the POST sequence that starts right after you turn your computer on. Usually the message with the required combination of keys is displayed during the startup testing. Pressing this combination of keys takes you to the menu of the Setup utility that is included in your BIOS.

The menu can differ in appearance and sets of items and their names, depending on the BIOS manufacturer. The most widely known BIOS manufacturers for PC motherboards are Award/Phoenix and AMI companies. Moreover, while items in the standard Setup menu are mostly the same for various BIOSes, items of the extended Setup heavily depend on computer and BIOS version.

Below we describe the general principles of initial hard disk configuring, and then configuring (sample) in AwardBIOS.



Large PC manufacturers, like Compaq, Dell, and Hewlett-Packard, produce motherboards themselves, and develop their own original BIOS versions. If your PC is of this kind, you have to refer to the documentation that came with your computer for instructions on proper BIOS configuring.

### A.3.2 Standard CMOS Setup Menu

Parameters in the Standard CMOS Setup menu usually define the geometry of hard disk. The following parameters (and values) are available for each hard disk installed in your PC:

Parameter	Value	Purpose
Type	1-47, Not Installed, Auto	Type 0 or Not Installed is used when there is no hard disk installed (to un-install it). Type 47 is reserved for user-defined parameters or for parameters detected by the IDE Autodetection utility.  Auto value allows for automatic detection of IDE disk parameters during the boot sequence.
Cylinder (Cyl)	1-65535	The number of cylinders on a hard disk. For IDE disks a logical number of cylinders is specified.
Heads (Hd)	1-16	The number of heads on a hard disk. For IDE disks a logical number of heads is specified.
Sectors (Sec)	1-63	The number of sectors per track of a hard disk. For IDE disks a logical number of sectors is specified.
Size (Capacity)	MBytes	The capacity of the disk in megabytes. It is calculated according to the following formula:  Size=(Cyl x Hds x Sct x 512) / 1024 / 1024.
Mode (Translation Method)	Normal/LBA/ Large/Auto	Method of translation of sector addresses.

For example, to demonstrate main features of MigrateEasy we used a Quantum Fireball TM1700A hard disk as one of the disks. Its parameters have the following values:

Parameter	Value
Type	Auto
Cylinder (Cyl)	827
Heads (Hd)	64
Sectors (Sec)	63
Mode	Auto
CHS	1707 MB
Maximum LBA Capacity	1707 MB

In BIOS setup you can set the Type parameter to User Type HDD (user-defined type). In this case you have to additionally specify the value of the Translation Method parameter, which can be Auto/Normal/LBA/Large.



Translation Method is the method that is used to translate sector addresses. This parameter appeared because in BIOS versions there were limitations to the maximum address capacity of disks, which is 504 MB (1024 cylinders x 16 heads x 63 sectors x 512 bytes). There are two ways to bypass this limitation: (1) switch from physical to logical sector addresses (LBA), (2) use mathematics to reduce the number of addressed sectors (cylinders) and increasing the number of heads; this way is called Large Disk (Large). The simplest decision is to set the value of this parameter to Auto.

If there are several hard disks connected to your motherboard, but you do not want to use some of them at the moment, you have to set the Type of these disks to Not Installed.

Parameters of hard disks can be set manually with help of information provided by the hard disk manufacturer on its case, but it is easier to use the IDE Autodetection utility that is usually included in modern BIOS versions.

IDE Autodetection utility is sometimes a separate BIOS menu item, and sometimes is included into the Standart CMOS Setup menu.



Please note that in Appendix A. «General Information. Hard Disks» we have described the general details of **physical** hard disk structure. Built-in IDE hard disk controls mask the physical disk structure. As a result, the BIOS of the motherboard «sees» **logical** cylinders, heads, and sectors. We are not going to elaborate on this issue here, but knowing about this fact can sometimes be useful.

### A.3.3 Arranging Boot Sequence. Advanced CMOS Setup Menu

Aside from Standard CMOS Setup, BIOS menu usually has an **Advanced CMOS Setup** item. Here you can adjust the **boot sequence**: C:: A:: CD-ROM:.



Please note that **boot sequence** management differs for various BIOS versions, e.g. for AMI BIOS, AWARD BIOS, and brand-name hardware manufacturers'.

Several years ago operating system boot sequence was hard-coded in BIOSes. An operating system could be booted either from a diskette (drive A:), or from the hard disk C:. That was the sequence in which BIOS queried external drives: if drive A: was ready, BIOS attempted to boot an operating system from a diskette. If the drive was not ready or there was no system area on diskette, BIOS tried to boot an operating system from hard disk C:.

At present, BIOS allows booting operating systems not only from diskettes or hard disks, but also from CD-ROMs, DVDs, and other devices. If there are several hard disks installed in your computer labeled as C:, D:, E:, and F:, you can adjust the boot sequence so that an operating system is booted from, for example, disk E:. In this case you have to set the boot sequence to look like E:, CD-ROM:, A:, C:, D:.



This does not mean that booting is done from the first disk in this list; it only means that the **first attempt** to boot an operating system is to boot it from this disk. There may be no operating system on disk E:, or it can be inactive. In this case BIOS queries the next drive in the list. Errors can happen during the booting, see A.3.5 «Hard Disk Initialization Errors».

BIOS enumerates disks according to the order in which they are connected to IDE controllers (Primary Master, Primary Slave, Secondary Master, Secondary Slave); next go the SCSI hard disks.

This order is broken if you change the boot sequence in BIOS Setup. If, for example, you specify that booting has to be done from hard disk E:, enumeration starts with the hard disk that would be the 3<sup>rd</sup> in usual circumstances (it is usually the Secondary Master).

After you have installed the hard disk in your computer and have configured it in BIOS, one can say that the PC (or the motherboard) «knows» about its existence and its main parameters, however, it is still not enough for an operating system to work with the hard disk.

### A.3.4 AwardBIOS

Below we provide an example of configuring hard disks in Award BIOS.

The **Main** AwardBIOS menu, which is responsible for configuring hard disks on motherboards for Celeron, Pentium II-III, AMD-K6/K7 processors usually looks like this:

AwardBIOS Setup Utility				
Main	Advanced	Power	Boot	Exit
System Time		[16:16:35]		Item Specific Help:
System Date		[01/10/2001]		
Legacy Diskette A		[1.44M, 3.5in]		<Enter> to go to sub-
Legacy Diskette B		[None]		menu.
Floppy 3 Mode Support		[Disabled]		
Primary Master		[Auto]		
Primary Slave		[Auto]		
Secondary Master		[Auto]		
Secondary Slave		[Auto]		
Language		[English]		
Supervisor Password		[Disabled]		
User Password		[Disabled]		
Halt On		[All but Disk/Keyboard]		
Installed Memory		[Disabled]		
F1 Help ↑↓ Select Item -/+ Change Values F5 Setup Defaults				
Esc Exit ←→ Select Menu Enter Select Sub-Menu F10 Save and Exit				

Place the marker to the Primary Master line and press the **ENTER** key to get to the screen that looks like this:

AwardBIOS Setup Utility		
Main	Primary Master [Auto]	Item Specific Help:
Type	[Auto]	<Enter> to select the type of the IDE drive.  [User Type HDD] allows you to set each entry on your own.  Warning: Ultra DMA Mode 3/4/5 can be enabled only when BIOS detects Shielded 80-pin cable.
F1 Help ↑↓ Select Item -/+ Change Values F5 Setup Defaults Esc Exit ←→ Select Menu Enter Select Sub-Menu F10 Save and Exit		

Place the marker to the value of the Type parameter ([Auto] in this case) and press the **ENTER** key to open the list of possible values of this parameter, for example:

- |                                                                                          |
|------------------------------------------------------------------------------------------|
| None<br>Auto<br>User Type HDD<br>CD-ROM<br>LS-120<br>ZIP-100<br>MO<br>Other ATAPI Device |
|------------------------------------------------------------------------------------------|

Use the ↑, ↓ keys to select the User Type HDD value and press the **ENTER** key to get to the following screen:

AwardBIOS Setup Utility		Item Specific Help:
Main		
Primary Master [User Type HDD]		
Type	[User Type HDD]	<p>&lt;Enter&gt; to select the type of the IDE drive.</p> <p>[User Type HDD] allows you to set each entry on your own.</p> <p>Warning: Ultra DMA Mode 3/4/5 can be enabled only when BIOS detects Shielded 80-pin cable.</p>
Translation Mode	[LBA]	
Cylinders	[.....0]	
Head	[.....0]	
Sector	[...0]	
CHS Capacity	0MB	
Maximum LBA Capacity	0MB	
Multi Sector Transfer	[Maximum]	
SMART Monitoring	[Disabled]	
PIO Mode	[4]	
Ultra DMA Mode	[5]	
F1 Help ↑↓ Select Item -/+ Change Values F5 Setup Defaults Esc Exit ←→ Select Menu Enter Select Sub-Menu F10 Save and Exit		

Enter the values of Translation Mode, Cylinders, Head, and Sector parameters (you can find them on the case of the hard disk) to finish the initial configuring of the hard disk with BIOS.

Translation Mode parameter can have the following values:

LBA
Large
Normal
Match Partition Table
Manual

New versions of AwardBIOS menu have the **Boot** item. The screen of the **Boot** menu of Award BIOS, which is responsible for adjusting the boot sequence, looks like this:

AwardBIOS Setup Utility				
Main	Advanced	Power	Boot	Exit
1. ATAPI CD-ROM		[None]		Boot Sequence: <Enter> to select the device To select the boot sequence, use up or down Arrow.
2. Removable Device		[Legacy Floppy]		
3. IDE Hard Drive		[IBM-DTLA-307020]		
4. Other Boot Drive		[Disabled]		
Plug & Play O/S		[No]		Press <+> to move the Device up the list, or <-> to move it down the list
Boot Virus Detection		[Enabled]		
Quick Power On Self Test		[Enabled]		
Boot up Floppy Seek		[Enabled]		
Full Screen Logo		[Enabled]		
F1 Help ↑↓ Select Item -/+ Change Values F5 Setup Defaults Esc Exit ←→ Select Menu Enter Select Sub-Menu F10 Save and Exit				

Place the marker to the field to the right of the 3. IDE Hard Drive item and press the **ENTER** key to open the list of all hard disks connected to your computer and detected by BIOS, together with the Disabled line, for example:

Disabled
Quantum FireBALL_1700A
IBM DTLA-307020
Quantum FireBALL_1700A

This device sequence in the upper left corner of the screen means that when looking for a boot disk BIOS will first query the CD-ROM (1. ATAPI CD-ROM), but there is no CD-ROM in the current configuration. Next it queries the floppy disk drive (2. Removable Device). Only if none of these disks contains an operating system, BIOS queries the hard disk (3. IDE Hard Drive) that has been selected from the list.



Select an item of the list with the ↑, ↓ keys and move the device up the list or down it with help of + or - keys. This will change the boot sequence. For example, you can select the boot sequence as the one shown below:

AwardBIOS Setup Utility				
Main	Advanced	Power	Boot	Exit
1. Removable Device		[Legacy Floppy]		Boot Sequence:
2. ATAPI CD-ROM		[None]		
3. IDE Hard Drive		[IBM-DTLA-307020]		<Enter> to select the device To select the boot sequence, use up or down Arrow.
4. Other Boot Drive		[Disabled]		
Plug & Play O/S		[No]		
Boot Virus Detection		[Enabled]		
Quick Power On Self Test		[Enabled]		Press <+> to move the Device up the list, or
Boot up Floppy Seek		[Enabled]		<-> to move it down the list
Full Screen Logo		[Enabled]		
F1 Help ↑↓ Select Item -/+ Change Values F5 Setup Defaults				
Esc Exit ←→ Select Menu Enter Select Sub-Menu F10 Save and Exit				

After setting the specified parameters, use the ←, → keys to select the **Exit** menu item. Reply positive to the prompt on saving the specified parameters.

AwardBIOS for motherboards for Pentium, Pentium Pro processors (i.e the ones older than those described above), and AMD-K5/K6 the Standard CMOS Setup screen looks like this:

ROM PCI/ISA BIOS (2A69JA2A)  
STANDARD CMOS SETUP  
AWARD SOFTWARE, INC.

Date (mm:dd:yy) : Fri, May 15 1998								
Time (hh:mm:ss) : 11: 33: 53								
HARD DISKS	: TYPE	SIZE	CYLS	HEAD	PRECOMP	LANDZ	SECTOR	MODE
Primary Master	: Auto	0	0	0	0	0	0	Auto
Primary Slave	: Auto	0	0	0	0	0	0	Auto
Secondary Master	: None	0	0	0	0	0	0	0
Secondary Slave	: None	0	0	0	0	0	0	0
Drive A : 1.44 M								
Drive B : None								
Video : EGA/VG					Base Memory: 0K			
Halt On: All Errors					Extended Memory: 0K			
					Other Memory: 512K			
					Total Memory: 512K			
ESC : Quit			↑↓→←: Select Item			PU/PD/+/- : Modify		
F1 : Help			(Shift) F2 Change Color					

TYPE parameter can have Auto, Not Installed, or User Defined values. In the latter case you have to manually specify the values of other parameters (except MODE) according to the information on the case of the hard disk.



The purpose of the MODE, LANDZ parameter is described in the note above. PRECOMP parameter is the so-called precompensation parameter and is important only for the older hard disks (MFM and RLL). IDE disks ignore this parameter.

In earlier BIOS versions the boot sequence was set in the BIOS FEATURES Setup section in the Boot Sequence line where you could select the most typical boot sequences, for example C:; A:;CD-ROM: or CD-ROM:; C:; A:, and other.

### A.3.5 Hard Disk Initialization Errors

Devices are usually initialized successfully, but sometimes errors may happen. Typical errors related to hard disks are reported by the following messages:

PRESS A KEY TO REBOOT

This error message is not directly related to errors during hard disk initialization. However, it appears for example, when the boot program finds no operating system on the hard disk, or when the primary partition of the hard disk is not set as active.

```
DISK BOOT FAILURE,  
INSERT SYSTEM DISK AND  
PRESS ENTER
```

This message appears when the boot program finds no available boot device, be that a floppy or a hard disk, or a CD-ROM.

```
C: DRIVE ERROR  
C: DRIVE FAILURE  
ERROR ENCOUNTERED  
INITIALIZATION HARD DRIVE
```

This message appears when it is impossible to access the disk C:. If the disk is positively functional, the reason for this error message is probably incorrect settings/connections of:

- hard disk parameters in BIOS Setup;
- jumpers on the controller (Master/Slave);
- interface cables.

It is also possible that the device is out of order, or the hard disk is not formatted.

### A.3.6 Booting Operating Systems

After the computer is turned on or rebooted, control gets to BIOS (Base Input/Output System). As it was said above, BIOS initializes and tests the hardware, and then reads the code from the **first sector** of the boot device (it is usually the first hard disk; the sector is called Master Boot Record, MBR) and passes control to it. All further actions depend on the code stored in this sector.

Usually the MBR of a hard disk contains the code that was written there by standard partitioning software (FDISK). This code does the following:

- Searches the partition table for the first partition marked as active;
- Tries loading the first sector of this partition to memory. Such sector of a partition is called boot sector;
- Passes control to the loaded sector.

Boot sector usually contains code that tries booting an operating system from the partition. Each operating system has its own boot code.

## Appendix B. Operating System and Hard Disks

You know that hard disks are used to store information. You also know (see A.1 «Hard Disk Organization») that information is stored in sectors, and the sectors appear on disks by way of low-level formatting. Each sector contains the smallest possible 512-byte data block.

Data blocks of larger size are stored in chains of such sectors. Each sector of a chain contains the address of the next sector in the chain or the marker telling that this sector is the last one in its service area. Chains of sectors make a **file**. The address (number) of the first sector of a file is stored in the file allocation table (FAT).

All applications you are familiar with – office applications (text and graphical editors, spreadsheet processors) – and documents, tables, images, e-mail clients and Internet access software, and games – are stored on hard disks as files. Files are grouped into folders to make working with them more convenient.

Among various software the most important one is the operating system.

An operating system provides all other applications running under its control with basic input/output access to all the resources of the PC, be that the CPU, memory, or external devices (monitor, hard disks, floppy disks, CD-ROM, DVD, printer). All applications are loaded into memory from a hard disk and are executed under the control of the operating system.

It would have been very inconvenient for a user to use data on disks if he/she had to deal only with addresses (numbers) of sectors comprising a file. That is why every file in the file system has its **name**. In DOS and Windows 3.x operating systems the name of the file contained 8 characters followed by a dot (.), which, in its turn, was followed by three more characters of the extension (or the type of the file). There could be fewer characters in the name and/or extension, and there were limitations as to what characters could be included in names and extensions.

Windows NT and newer OS versions allow files names that are up to 255 characters long. Full file name (including the disk letter and the name of the folder containing the file) can be up to 260 characters long, however it is not recommended using file names longer than 50-70 characters.

Other operating systems, like Linux, have never had such strict limitations on file names as did DOS or Windows. The name of the file in Linux can be up to 256 characters long and have not one, but several extensions (or none at all). The path to the file in Linux can be up to 4096 characters long.

To provide access to data organized in files and folders each operating system, for example, MS-DOS, Windows 95/98/Me/NT/2000, Linux, has its own characteristic way to create and manage hard disk space, which is generally non-compatible with other OSes. However, it would be inefficient if one hard disk could be used by only one operating system. That is why a mechanism was created allowing several operating systems to use the hard disk. This mechanism breaks the hard disk space into **partitions**.

## **B.1 Hard Disk Partitions**

### **Partitioning Hard Disks in DOS/Windows**

Partitioning hard disk is done by special applications. In MS-DOS and Windows the widely known FDISK program is used for this purpose. It allows creating partitions, setting their size and labels. FDISK allows allocating either all hard disk space or part of it for MS-DOS or Windows, and leave some part of it for other operating systems.

FDISK can perform the following functions:

- create a primary DOS partition containing one logical disk;
- create an Extended Partition that can be broken into any number of logical disks;
- mark a partition as active (only one primary partition can be active).

The structure of partitions on the hard disk can be shown as follows:

MBR.
Primary partition 1-1.
System logical disk C:.
Extended partition 1-2.
Logical partition 1-5.
Logical disk D:.
Logical disk E:.
Logical disk F:.
.....

Additional partitions can be created in the unallocated disk space with tools from **other** operating systems (OS/2, Linux).



Information about partitions on a hard disk is stored in a special disk area – in the 1<sup>st</sup> sector of the 0<sup>th</sup> cylinder, header 0, which is called a Partition Table. This sector is called the Master Boot Record, MBR.



A physical hard disk can contain up to 4 **primary** partitions. This limitation is due only to the capacity of the Partition Table, which itself consists of only four partitions. However, this does not mean that you can install **only 4** different operating systems. Modern software disk managers allows the installation of many more operating systems. For example, the **Acronis OS Selector 5.0 Deluxe** disk manager, allows a user to install up to 100 operating systems on one hard disk!

Partitioning hard disks is a feature that is available in all operating systems.

## B.2 Creating Partitions for Other OSes

In this part of the Guide we have described the way to partition a hard disk by the FDISK program for the DOS/Windows operating system. As it has been said above, the FDISK program creates primary and possibly (but necessarily so) extended DOS partitions and logical partitions (disks) on them, and can also leave disk space unallocated to be used by other operating systems. However, the Windows operating system, while being a very widely distributed operating system for PCs, is not the only operating system.

You can configure the hard disk of your computer to used with the Linux operating system. In this case you do not need to create either primary nor extended DOS partitions (if you are not going to use DOS/Windows).

For example, the installation program from the ASPLinux operating system includes a disk manager that allows partitioning a disk into partitions that can be used by this OS.

The **Acronis DiskEditor** (or more precisely, the disk manager that comes with it) also has all the capabilities of creating partitions on hard disks.

### **B.3 Primary and Logical Partitions**

Many operating systems, including DOS and Windows 95, can only be booted from primary partitions.

Some operating systems do not see primary partitions except those they were booted from (OS/2).

Knowing these limitations allows you to decide what partitions are better to use and for what purposes. Primary partitions are better used for booting operating systems and storing system folders and files.

Logical partitions can keep all the other information, since they are accessible from most operating systems.

If you are going to keep several operating systems on your computer, it is better to install those that can be booted from logical partitions onto logical partitions, so that they do not use up extra primary partition space.

### **B.4 Formatting Hard Disks**

Within each of the partitions information has to be organized in a way understandable for the operating system that uses the partition. This organization is the **file system**. Format and location of information (the logical structure of a partition) for MS-DOS and Windows is created by the FORMAT program.

FORMAT program performs the following functions:

- It creates the boot record;
- It creates the file allocation table (FAT);
- It marks bad clusters so they are not used.

After formatting, logical disks created by the FDISK program are organized as follows:

- Logical disks start with the boot sector;
- One or several copies of file allocation table (FAT) are placed after the boot sector;
- Next goes the root folder;
- Then the data area.

Each logical disk has to be formatted with FORMAT separately.

Formatting a partition is done by issuing a DOS command:

```
FORMAT C :
```

To create a system (bootable) logical disk, it has to be formatted with FORMAT with /S parameter:

```
FORMAT C: /S
```

Another way is to transfer the system to the partition that has been formatted with /B parameter is booting the computer from a system diskette and issuing the command

```
SYS C :
```

IO.SYS, MSDOS.SYS, COMMAND.COM files that are necessary for the operating system to be booted are copied from the diskette to the disk. Initial loader code is written to the boot sector.



Other operating systems provide hard disk formatting means to create file systems that are recognizable by these OSes.

## B.5 File Systems

The logical structure that has been created on the hard disk is supported by means of operating system. The file system itself presents the information on the disk as an ensemble of **files** and **folders**.

From the user's point of view a file is a unit of storage of logically connected information: text, graphics, and sound. As for data storage organization, a file is a chain of connected sectors or **clusters**. A cluster is a unit of several sectors. (Sectors are characteristic for file systems supported by various versions of Windows.)



Operating systems support file systems on hard disks (or disk partitions) by allowing to create, copy, and delete files and folders.

At present, the most widely spread file systems for PCs are the following two:

- **FAT** (File Allocation Table) for DOS, OS/2, Windows 95/ 98/Me/NT/2000;
- **NTFS** (Windows NT File System) for Windows NT/2000.

However, there are many more other file systems. The Linux operating system, for example, uses three different file systems:

- **Ext2** is a common file system used on desktop PCs running Linux;
- **Ext3** is the default file system used by Red Hat Linux;
- **ReiserFS** is a more secure (with respect to data integrity) file system that is used on many Linux data servers.

For details on file systems see Appendix C. «File Systems».

# Appendix C. File Systems

All file systems are made of structures that are necessary to store and manage data. These structures are usually comprised of operating system boot sectors, folders, and files. File systems perform the following basic functions:

1. Track occupied and free disk space (and bad sectors, if any);
2. Support folders and file names;
3. Track physical location of files on disks.

Different operating systems use different file systems. Some operating systems are able to work with only one file system while other can use several of them. Let us describe some of widely used file systems in detail.

## C.1 FAT16

FAT16 file system is widely used by DOS (DR-DOS, MS-DOS, PC-DOS, PTS-DOS и др.), Windows 95/98/ME, Windows NT/2000/XP operating systems and is supported by most other systems.

Main features of FAT16 are the file allocation table (FAT) and clusters. FAT is the core of the file system. To increase data safety it is possible to have several instances of FAT (there are usually 2 of them). A cluster is a minimum data storage unit in FAT16 file system. One cluster contains a fixed number (some power of 2) of sectors. FAT stores information about what clusters are free, what clusters are bad, and also defines in what clusters files are stored.

Maximum size of a FAT16 file system is 4 gigabytes, and the maximum number of clusters is 65525, the largest cluster being 128 sectors. Usually cluster size is selected as minimum as possible so that the resulting number of clusters is less than 65526. The larger the partition size, the larger the cluster has to be. Most operating system incorrectly perform with 128-sector clusters, thus reducing the maximum FAT16 partition size to 2 gigabytes.



Usually the larger the cluster size the more disk space is wasted.

Like many others, FAT16 file system has a root folder. Unlike others however, its root folder is stored in a special place and is limited in size (standard formatting produces a 512-item root folder).

Initially, FAT16 had limitations to file names that could only be 8 characters long, plus a dot, plus 3 characters of name extension. However, long name support in Windows 95 and Windows NT bypasses this limitation. OS/2 operating system also supports long names, but does it in another way.

## **C.2 FAT32**

FAT32 file system appeared in Windows 95 OSR2 and is also supported by Windows 98/ME and Windows 2000/XP. FAT32 grew out of FAT16. The main differences between FAT32 and FAT16 are 28-bit cluster numbers and more flexible root folder implementation, which is not limited in size. The reason for FAT32 is the necessity to support large (larger than 8 gigabytes) hard disks and the inability to build any more complex file system into MS-DOS, which is still in the core of Windows 95/98/ME.

Maximum FAT32 file system size is 2 terabytes.

## **C.3 NTFS**

NTFS file system is the main file system for Windows NT/2000/XP. Its structure is closed, so no other operating system fully supports. Main structure of NTFS is the MFT (Master File Table). NTFS stores a copy of the critical part of the MFT to reduce the possibility of data damage and loss. All other NTFS data structures are special files.

Like FAT, NTFS uses clusters to store files, but cluster size does not depend on partition size. NTFS is a 64-bit file system, it uses Unicode to store file names. It is also a journaling (failure-protected) file system, and supports compression and encryption.

Files in folders are indexed to speed up file search.

## **C.4 Linux Ext2**

Ext2 is one of the main file system for the Linux operating system. Ext2 is a 32-bit file system. Its maximum size is 16 terabytes. Main data structure that describes a file is an i-node. A place to store the table of all i-nodes has to be allocated in advance (during formatting).

## **C.5 Linux Ext3**

Officially introduced with their version 7.2 of the Linux operating system, Ext3 is the Red Hat Linux journaling file system. It is forward and backward compatible with Linux ext2. It has multiple journaling modes and broad cross-platform compatibility in both 32- and 64-bit architectures.

## **C.6 Linux ReiserFS**

ReiserFS was officially introduced to Linux in 2001. ReiserFS frees many Ext2 disadvantages. It is a 64-bit journaling file system that dynamically allocates space for data substructures.

## Appendix D. Glossary

**Active partition.** One of the primary partitions of a hard disk is usually active. Default MBR code tries to boot an operating system from the active partition of the first hard disk. Letter assignment in Microsoft operating systems depends on what partitions are active.

**Bad cluster.** A cluster that contains bad sectors. Such cluster cannot store useful information.

**Bad sector.** A sector that cannot store the information written, for instance due to defects or aging of the magnetic surface.

**BIOS extension.** BIOSes that were released before 1994 could only support hard disks that were less than 8 gigabytes. Extended hard disk management functions were added to BIOSes to solve this problem, and now the maximum supported hard disk capacity is  $2^{75}$  bytes.

**Boot partition of an operating system** is a partition from which the initial stage of operating system booting is done (boot sector is read and executed, first file of the operating system is read and executed).

**Boot record.** The initial part of a partition that contains code and data necessary for booting an operating system. May consist of one or several sectors. First sector of a boot record must end with the boot sector signature (0AA55h).

**Boot sector** is the first sector of a disk or a partition that contains the initial code for the operation system booting. Boot sector must end with 0AA55h signature.

**Bootable disk** is a disk from which an operating system may be booted. A bootable disk must contain a boot sector of an operating system and the necessary system and configuration files. The «Bootable disk» term usually refers to diskettes and CD-ROMs.

**Bootable partition.** A partition that can host an operating system. In the beginning of such a partition there should be a boot record.

**Booting an operating system** is initialized by reading its boot sector to the memory at 0:7C00h and passing control to it. Since each operating system has its own boot sector, it is able to perform the required actions on loading and initializing system and configuration files.

**Booting the computer** is a procedure that is executed every time a computer is turned on or an operating system finishes its work or when the reset button is pressed. Booting consists of the following stages:

- Hardware diagnostics;
- Memory check;
- Built-in BIOS initialization;
- Initialization of additional hardware components and their BIOSes (video, SCSI etc.);
- Booting an operation system.

**Cluster.** Information storage unit in such file systems as FAT and NTFS. Every file occupies a certain number of whole clusters, so the more the cluster size the higher the losses are that are due to file size adjustment, but the smaller the cluster the larger the cluster distribution tables.

**Cylinder.** A group of all the tracks on all the magnetic platters of a hard disk that can be accessed without moving the magnetic head. Access to the data inside one cylinder is much faster than moving the head from one cylinder to the other.

**Disk.** It is a general term that can mean both a data storage media (floppy disk, compact disk), and a data-reading device (hard disk), and a partition that is accessible from some operating system (logical disk).

**Diskette.** Floppy disk. A removable storage media that consists of a flexible magnetic platter enclosed in a protective envelope. Nowadays the most common are 3.5" diskettes with 1.44 megabyte capacity.

**File.** A file is named information storage in the file system. In different file systems files can be stored in different ways, different may also be ways to store file names and to write the full path to the file in the folder tree.

**File system.** Data structure that is necessary to store and manage files. File system does the following functions: tracks free and occupied space, supports folders and file names, tracks the physical positions of files on the disk. Each partition may be formatted with its own file system.

**Floppy disk drive.** A device that reads and writes data to diskettes.

**Folder.** A table in the file system that contains description of files and other folders. Such structure allows creating folder tree that begins with the root folder.

**Formatting.** The process of creating service structure on the disk. There are three levels of hard disk formatting: low-level (marking the magnetic surface with tracks and sectors), partitioning and high-level (creation of file system on a partition).

**Hard disk.** Fixed storage media along with integrated electronics that consists of several magnetic platters that rotate synchronously on one spindle. Hard disks have relatively high capacity and high read/write speed.

**Hard disk geometry.** A set of hard disk parameters that usually includes the number of cylinders, heads and sectors per track.

**Head (magnetic head, read/write head).** A hard disk consists of several magnetic platters, for each side of each platter there is a head that is used to read and write information on it.

**Hidden partition.** A partition that is somehow made invisible to the operating system. Usually partitions are hidden by changing their type.

**Label.** An optional name that can be assigned to a partition to simplify its identification. Usually has the same limitation as file names. For example, FAT partitions have labels up to 11 characters long, but may contain spaces.

**Large hard disk support software.** Some BIOSes have troubles supporting large (larger than 8 gigabytes) hard disks. That is why some hard disk manufacturers provide special software that is installed in the beginning of the hard disk, is loaded before any operating systems, and substitutes hard disk access BIOS functions.

**Letter (of a drive, partition).** All operating systems that are somehow DOS-compatible use Latin letters to identify drives and partitions. Letters A: and B: are usually reserved for floppy drives. Starting with C: letters are assigned to hard disk partitions that can be recognized by the operating system. Separate letters may be assigned to CD-ROMs, to other disk drives and to network drives.

**Logical disk** is a partition whose file system is recognized by the operating system. Usually each logical disk is assigned with a letter that uniquely identifies it.

**Logical partition.** A partition information about which is located not in MBR, but in the extended partition table. The number of logical partitions on a disk is unlimited.

**Master Boot Record (MBR)** is a special place in the very first sector of the hard disk to store information about the hard disk partitioning and code to be loaded with BIOS. All the actions that follow depend on the contents of this code.

**Operating system** is a set of programs that usually includes kernel, drivers, shell, and system programs that is used for centralized hardware management and hiding the details of hardware management from the user and applications.

**Operating system installation** is a process during which its system folders are created, system files are copied and boot sector is created.

**Partition.** An independent area on a hard disk where a file system can be located. A partition can be either primary or logical, depending on its position in the partition structure. One of the primary partitions of a hard disk may be active. A partition has the following attributes: type, beginning and size. Besides, some partition managing software and boot managers allow hiding partitions. Information about partitions is stored in the partition table.

**Partition structure.** All the partitions on a hard disk make a tree with the root in the MBR partition table. Many operating systems and programs assume that any partition table but MBR may contain not more than one partition entry and one table entry, and it simplifies the partition structure greatly – all the logical partitions form one chain.

**Partition table.** It is the table that contains the information about partitions and links to other partition tables. A partition table cannot have more than four entries. Main partition table is located in the hard disk MBR, and the rest partition tables are called extended. Partition tables are usually stored in the first sector of a cylinder.

**Partitioning.** The process of creating the partition logical structure on a hard disk. Partitioning is usually done with programs like FDISK. Disk administrator completely replaces FDISK where functionality is concerned and allows performing many more useful operations.

**Primary partition.** The partition, information about which is contained in the MBR partition table. Majority of operating systems can be booted only from the primary partition of the first hard disk, but the number of primary partitions is limited.

**Physical disk.** A disk that is physically a separate device. Thus, a floppy disk, hard disk, CD-ROM are physical disks.

**Root folder.** The folder where the folder tree of a file system begins. Starting from the root folder one can uniquely describe the file position on the folder tree by sequentially naming all the intermediate nested folders, e.g.: \WINDOWS\SYSTEM\VMM32.VXD. Here the WINDOWS folder is a subfolder of the root folder, SYSTEM folder – of the WINDOWS folder, and the VMM32.VXD file is located in the SYSTEM folder.



**Sector.** It is the minimal information unit on a disk that is transferred in single read or write operation. Usually a sector is 512 bytes large. A sector on a disk can be addressed two ways: via the absolute number or via cylinder, head, and sector number on a track.

**Status.** A flag that shows if a partition is active. This flag is stored in the partition table and has no meaning for logical partitions.

**System disk/partition** is a disk/partition from which an operating system may be booted. Such disk usually contains the boot sector and system files of this operating system.

**System file** is a file that contains the code and constant data for an operating system. Each operating system has its own system file set.

**System folder.** Some operating systems keep most of their files in a special folder on a partition that may be different from the system one. For example for Windows 95/98/ME operating systems IO.SYS system file resides on the system partition, while other system files are located in the system folder which is usually called WINDOWS. Program Files folder can also be treated as system since it resides on the same partition as WINDOWS and also contains files that are relevant to the operating system.

**Track.** Disks are divided into concentric circles called tracks. Information from one track can be accessed without moving the head.

**User interface** is a set of principles, concepts, and means by which programs interact with the user. For example in window interface all input and output is done in windows and the mouse is widely used.