PARAGON
Partition Manager

User Manual

Paragon Technology GmbH, System Programmierung

Copyright © Paragon Technology GmbH

Published by:
Paragon Technology GmbH
System Programming
Pearl-Str. 1
D-79426 Buggingen
Germany

1. Introduction, installation, notes etc..

Summary – please read

Important: The DOS and Windows versions are documented separately.

Chapter one: Installation and program execution.
Chapter two: The Windows version.
Chapter three: The DOS version.
Chapter four: Tutorial/reference to chapters 2 and 3 and to the examples in chapter 5.
Chapter five: Examples.

1.1 Important information

Last minute changes are documented in README.TXT

Support:
By e-mail: support@partition-manager.com; support@penreader.com

System requirements:
These requirements apply to the operation of DOS as well as Windows. For the successful work with FAT-16 partitions, there must be a minimum available RAM of 16 megabytes. For the work with FAT-32, you should have at least 32 MB of RAM. In either case, you should have approximately 7 MB available hard drive space as well as a CD ROM drive and floppy diskette drive.

Important: Naturally, the Windows installation program will work only on systems that have Win 95/98/ME installed (no WIN 3.x). Systems that have only DOS will require manual installation, by either copying the installable files directly from the CD or by creating a bootable diskette using the DOS executable PREPDISK.BAT

In order to better understand partitions, MBRs, SBRs, operating systems, etc., please refer to the **fundamentals** section of this document. This section will explain to you (among other things) that reformatting your drive will not remove any boot manager.

1.2 Abbreviations and Explanations

PTS = Paragon Technology Systems
DB = Drive Backup
PM = Partition Manager
BM = Boot Manager
DR = Any drive, either physical or logical
Physical hard drive = For our purposes, always a hard drive (the hardware)
Physical hard drive ID #: (ie 128, 129) [aka 80H and 81H] (sometimes also identified as 0 and 1)

OS =	Operating System
Logical Drive =	A single logical drive either primary or within an extended partition.
Logical drive identifier	The letters A thru Z.
Partition =	A sequential portion of the hard drive, from track to track.
Primary Partition =	A bootable partition, which, when set "active", the partition table will recognize as "bootable". Actually becomes bootable with an operating system installed.
Extended Partition =	Not bootable, a framework (sub-section of the hard drive) within which we can create (multiple) logical drives. This area is recorded/tracked just like a primary partition within the partition table
BR or SBR =	Boot Record or System Boot Record, the first sector in a primary partition if an OS is installed.

1. Raw, non partitioned disk space will be indentified by PM as **primary <free>**. Raw space in extended partitions will be identified as **logical <free>**.
2. Primary partitions are directly bootable through the MBR, as long as operating systems have been installed on those areas (the number one sector has an SBR).
3. Logical drives are partitions visible to the operating system. But an Operating systems see these only when:

a) The partition is formatted.
b) The formatting is familiar to the OS (DOS sees only FAT-16, Win 95 (B)/98 see both FAT-16 and FAT-32).
c) The (logical) partition in question isn't hidden.

4. Active/Inactive is actually a flag that's used by the partition table to identify the partition as bootable or not. Only one primary partition can be activated at a time.
5. Hidden/Not hidden: If it's hidden, the partition ID is not recognized at all and hence the partition is totally invisible to the OS.
6. Partitions are closed physical units (from track to track). When a partition is formatted using a file system for a particular OS, the partition becomes a new addressable logical unit.
7. Adjacent deleted partitions free up the entire area as a unit: [IE (raw) disk space to a primary **<free>** or within an extended partition to logical **<free>**].

1.3 About the program versions

It's our philosophy to cater to all users, especially and including those that are limited to DOS only. Therefore, the basic kernel of operations revolving around these programs operate at the basic DOS level. Windows users will gain the full benefit of Win 95/98 in their usage, taking full advantage of the GUI interface, toolbars, right-clickability, etc.. DOS users on the other hand, have the advantage of being able to boot from a floppy at any time and being able to work with a completely unbootable drive, and/or one without any valid OS whatsoever.

1.3.1 Functions / scope

Partition Manager must be run from either the DOS or Windows operating systems.

All versions can perform the following tasks on a hard drive.

- Modify the size of partitions with data.
  - ⇒ Increase the size of the partition up to the available free disk space.
  - ⇒ Decrease the partition size to fit the data only.
- Copy (clone):
  - ⇒ Entire hard drives or individual partitions.
    - ∗ Perform in 1:1 mode (sector-by-sector copy), including foreign or even defective file formats.
    - ∗ Adjust to the size of the target.
- Move partitions.
- Create new partitions.
  - ⇒ Create primary bootable partitions.
  - ⇒ Create extended partitions.
  - ⇒ Create new logical drives in extended partitions.
- Format new and existing partitions in a variety of file systems.
- Delete partitions.
- Activate/inactivate (make bootable/unbootable) primary partitions.
- Hide/unhide partitions.

Furthermore, the DOS version can install a boot manager which can boot up to four operating systems (one per primary partition on hard drive 0, 80 Hex). It can not boot multiple operating systems from one partition or from several hard drives.

The Windows version works under all WIN 9x versions.

The DOS version works under:

- Windows 9x only in the DOS "single task" mode.
- All native DOS, 5.0 and above.
- From a bootable DOS floppy, DOS 5.0 and above (even if the hard drive isn't bootable).

Common features

- Clone (copy) partitions in the 1:1 mode, also known as **image copy** or **sector-by-sector** copy. Operates not on the logical level but on the sector level (copies sector-by-sector without regard to contents). This mode is extremely efficient because it interacts directly with the physical sectors. (works with all known, unknown, copy-protected or even defective file systems).
- Clone (copy) in autoresize mode (automatically adjusting to the size of the target, manual or *automatic* mode).

- Use of fast copy algorithm (proprietary read and write routines) which bypass the I/O routines of the operating system to speedup the copy process.
- Detection and avoidance of defective sectors on source and target.

1.3.2 Partition Manager functions and capabilities

When partitioning a hard drive, what actually happens is that data are written to the partition table that define the beginning and end of a partition. It's important to note that when using (the standard) FDISK program to partition a drive, the whole area within the partition is actually not accessed at all. That is, the area within the partition remains exactly as it was. In order to completely re-allocate already available partitions of a hard drive, programs such as FDISK completely delete the old entries in the partition table and then write new entries which contain the new boundaries of the partition. When this happens, of course the logical addressability of all existing files (from before) is completely lost.

Conversely, partitioning programs like Partition Manager can extend or reduce partitions without any data-loss. Here also, new partitions are created, but the main purpose is the resizing of existing partitions. In this case the main work takes place directly within the partition, where all files are rearranged to fit the new size of the partition and therefore all file and directory entries have to be rewritten (FAT-table in the case of FAT-systems). Also, it should be noted that this is a very complex process and must be executed with great care to avoid data loss. Extra caution must be taken if the cluster size is to be changed in the process of repartitioning FAT-16 partitions.

Regarding changing the size of partitions containing data:
Repartitioning of any drive should always be given a great deal of forethought, since considerable data loss could ensue if any error were to be made. Your partition contains your data, and its loss means the loss of all work contained therein.

A repartitioning with data essentially entails one of the following processes:
a) Shrinking of only one partition (as small as the data in it) so (a minimum of) one more partition can be created.
b) Increasing the size of a partition to include adjacent free space (to make room for more data and programs).
c) Optimizing or converting a large FAT-16 partition by division into smaller sub-portions.
d) Installation of a variety of different data and operating systems.

Cloning of hard drives and partitions
Transferring the contents of either a hard drive (or a partition within a drive), to a new drive or partition (including in the 1:1 mode), while adjusting to the size of the destination drive or partition.

Facilities and services for partitions
The following tasks can be performed:
a) Creation of new partitions.
b) Formatting of partitions for various types of file systems.
c) Deleting partitions.
d) Moving partitions.

Creating a multi-operating system PC, using Boot Manager Easy
In case you want to be able to utilize multiple operating systems, after creating/resizing the necessary primary partitions and installing these operating systems, the option of booting off these various primary partitions can be achieved with Boot Manager Easy.

1.3.3 Benefits of using Partition Manager

- Quick repartitioning of partitions with data and programs.
  - ⇒ Easily enlarging a partition if it has become too small.
  - ⇒ Reducing the size of a partition to make room for another partition.
- Converting the file system of partitions as follows:
  - ⇒ From FAT-16 to a FAT-32 type file system.
  - ⇒ From FAT-32 to FAT-16.
- Move partitions on the hard drive.
- Hiding/unhiding of any partition.
  - ⇒ Determination of drive lining up sequences
- Enabling/disabling the bootability of systems (partitions).

- Allowing access to user-dependent partitions; Creation of completely separate partitions that are invisible to one another.
  - ⇒ Games only partition (for the kids).
  - ⇒ Work only partition (for safe normal work)
  - ⇒ Test partition (testing new programs).
- With the purchase of a new hard drive:
  - ⇒ Quick cloning of the old drive to the new one.
- Setup and administration of a multi-operating system PC.
  - ⇒ Creating/resizing partitions for installing new OS.
  - ⇒ Booting the operating systems from an easy boot "menu".

In all these cases:

- All the partitions maintain all their original data and programs
- The operating system (if any) contained in the partition remains completely bootable.
- Reinstallation of operating system and/or programs is not necessary.
- All services for partitions
  - ⇒ Create new partitions.
  - ⇒ Format Partitions for the following file systems:
    - ∗ FAT-16
    - ∗ FAT-32
    - ∗ NTFS
    - ∗ HPFS
    - ∗ EXT2FS

  Delete partitions.

1.3.4 The Windows version

PC requirements:

- Workspace of at least 16 MB. For a FAT-32 partition, at least 32 MB.
- Operating system: WIN 9x (95/98/ME/NT 4.0 Windows 2000, -XP)

The Windows version of Partition Manager must of course be installed exactly like a standard Windows program or it won't be executable. Other than that, the execution of PM behaves precisely like any other Windows program, as a multitasking program with a GUI interface, menubar, toolbar and right-click functionality of other Windows programs.

1.3.5 The DOS version

PC requirements

- Workspace of at least 16 MB. For a FAT-32 partition, at least 32 MB.
- Operating system:
  - ⇒ The DOS single task text mode of Windows 9.x
  - ⇒ Any DOS, version 5.0 or above.

The DOS version of Partition Manager and the Windows version are both automatically installed under WIN-95/98. If no Windows 9x is available, the programs must be manually transferred to the hard drive (see chapter 1.4.2 and 1.4.3). When starting under Windows, the DOS version of PM runs in the single task text mode (DOS mode of Windows 95/98) or it can be executed from a natural DOS (MS/DOS 6.x or under our provided DOS 7, see also chapter 1.5.2).

1.3.6 Similarities and differences in the different versions

Some differences exist in the DOS and Windows versions of Partition Manager, but they are essentially the same other than a few variations in the interface. All main functions exist in both.

1.3.7 Partition Manager functions and particular file systems

OS: **file system:**	**DOS/Win** **FAT-16/-32**	**WIN-NT** **NTFS**	**Linux** **EXT2FS**	**OS/2** **HPFS**	**Other** **systems**
copying and moving	YES	YES	YES	YES	1:1
resizing	YES	YES	YES	YES	NO

formatting	YES	YES	YES	YES	NO
Copy with autoresize	YES	YES	YES	NO	NO
File system conversion	YES	YES	NO	NO	NO
cluster resize	YES	NO	NO	NO	NO

1.4 Installation of the programs

The installation of PM varies significantly for each operating system it is being installed under. If a Windows 9x is available, both the DOS and Windows versions will automatically be installed. With DOS naturally only installation of the DOS version is relevant (and this has to be done completely manually).

1.4.1 Under Windows 9x, ME, WIN NT, 2000 and XP

This is one of those standard, well-known Windows installations. You just call the Setup.exe from the CD using Start, Run. The setup of Partition Manager then installs both the DOS and Windows versions for all Win 9x versions, as well as the Floppy Build Wizard program. PM setup is a very basic, dialog-driven program and requires no further explanation.

Under Windows ME, NT, 2000 and XP there will be no DOS version installed because in this versions there is no DOS available.

The automatically co-installed Floppy Build Wizard program allows creation of bootable diskettes, either during the installation, or at any later time. **(See section 1.6, Floppy Build Wizard)**.

1.4.2 Under DOS (or Windows 3.x)

If there is no Windows 9x available, the DOS version of Partition Manager must more or less be transferred manually from the CD to the hard drive (and the diskettes). We assume of course, under DOS (or Win 3.x), you have access to your CD-ROM drive, or no installation will be possible.
In the DOS version, change directory into \PMDOS on the CD-ROM and run PMCOPY.BAT. This program creates a directory called PMDOS on the hard drive, and copies all files for the DOS version from the CD into this hard drive directory. After copying the files, it removes the R/O attribute of the copied files. The installation is then complete and the program is addressable as PM.EXE in the PMDOS directory.

If you‘re a DOS expert you can of course manually create your own custom PMCOPY.BAT

1.4.3 Creating bootable diskettes

Creating bootable diskette under DOS.

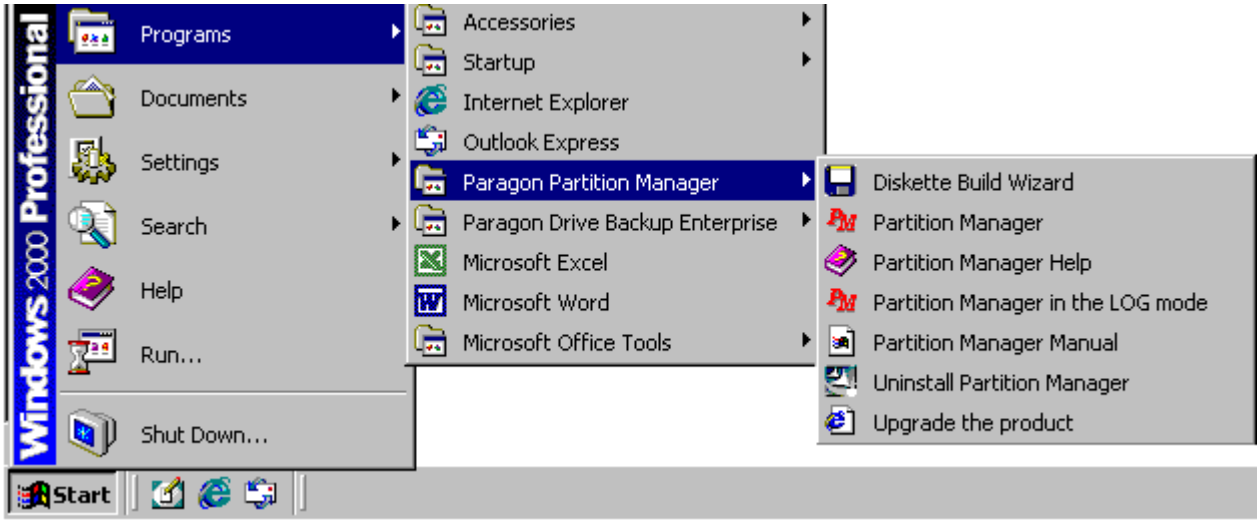
1. Boot under a DOS or run MS-DOS prompt from Windows.
2. Insert a formatted diskette (1.44 MB) into the A: drive.
3. Clean the diskette and make is bootable. The latter can be performed by means of standard DOS command ‘sys.com’ (sys.com A:).
4. Copy also following file onto the diskette: himem.sys, ramdrive.sys.
5. Change the currently-logged drive (if needed) and change directory to the PM directory (PMDOS).
6. Copy the contents of the subdirectory ‘PACKED’ onto the diskette.
7. The diskette is ready, you can boot computer with it.

Creating bootable diskette under Windows

Bootable diskette can be created during installation of Partition Manager (see pic.2), or later by means of Diskette Build Wizard utility, which is included in Partition Manager installation package.

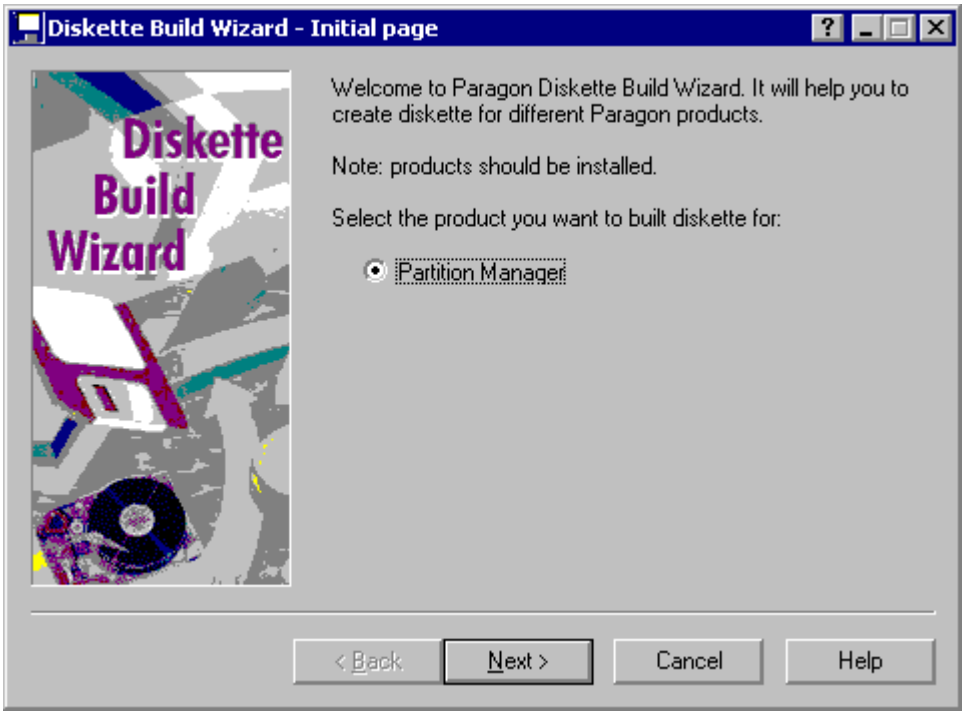
Creating bootable diskette under Windows 95/98

You can create bootable diskette by pressing ‘Yes’ button in dialog window (pic.2), or by having launched Diskette Build Wizard (from the product program group).



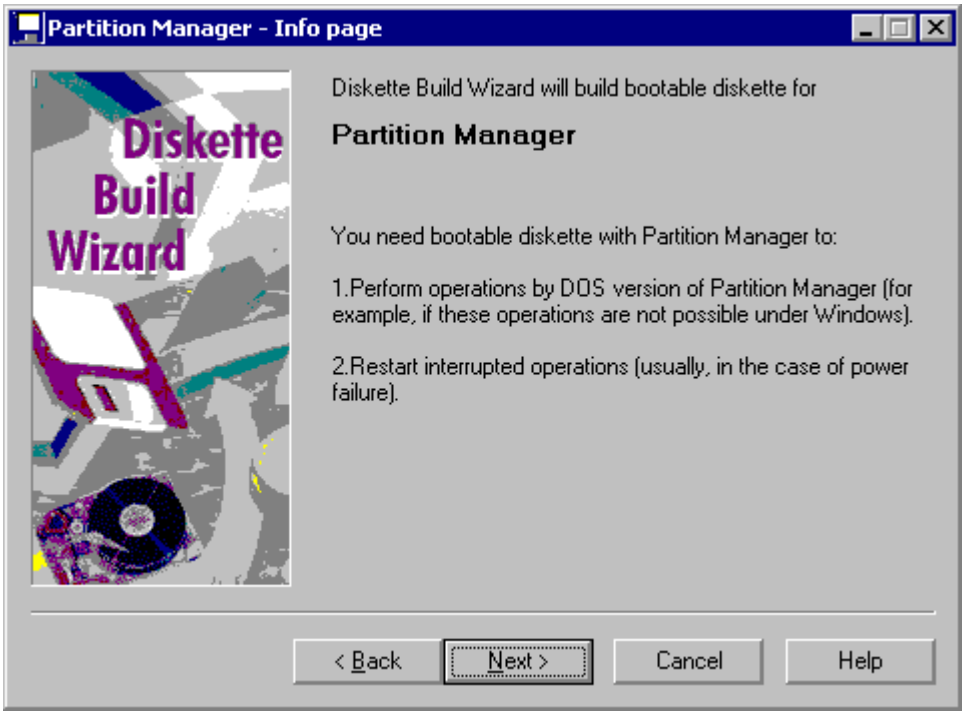
Pic. 1. Launching Diskette Build Wizard

Then Diskette Build Wizard will start and the first dialog will appear:



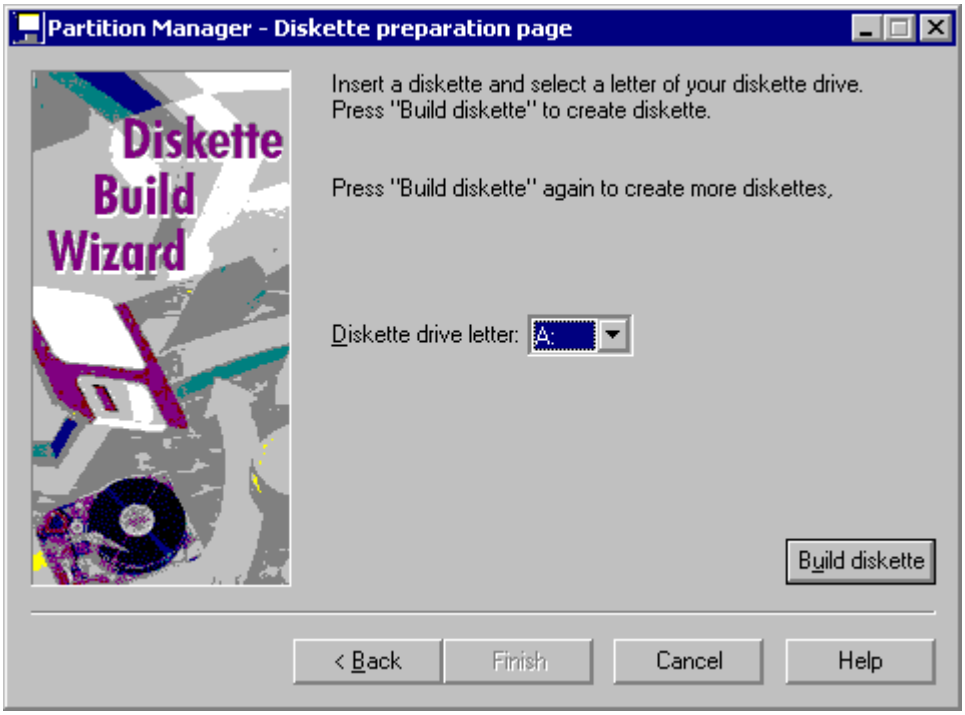
Pic. 2

You can choose separate component to create diskette with. That is, if you choose BootManager, then the diskette will start BootManager, or if you choose Partition Manager, then the diskette will start DOS Partition Manager. You can read about working of DOS programs in further chapters.
Having chosen the component and pressed 'Next' button you will see information dialog window with the component name and some corresponding hints. You can continue creating bootable diskette by pressing 'Next', or return to previous dialog of selecting components (pic. 4).



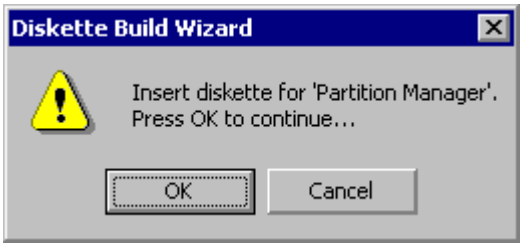
Pic. 3

Having pressed 'Next' button, you will see the next dialog, where you will be prompted to insert formatted diskette in floppy drive and create bootable diskette. Creating of bootable diskette will be started after pressing 'Create' button. You can create ad lib bootable diskettes, as after creating first diskette Diskette Build Wizard will return to this dialog.



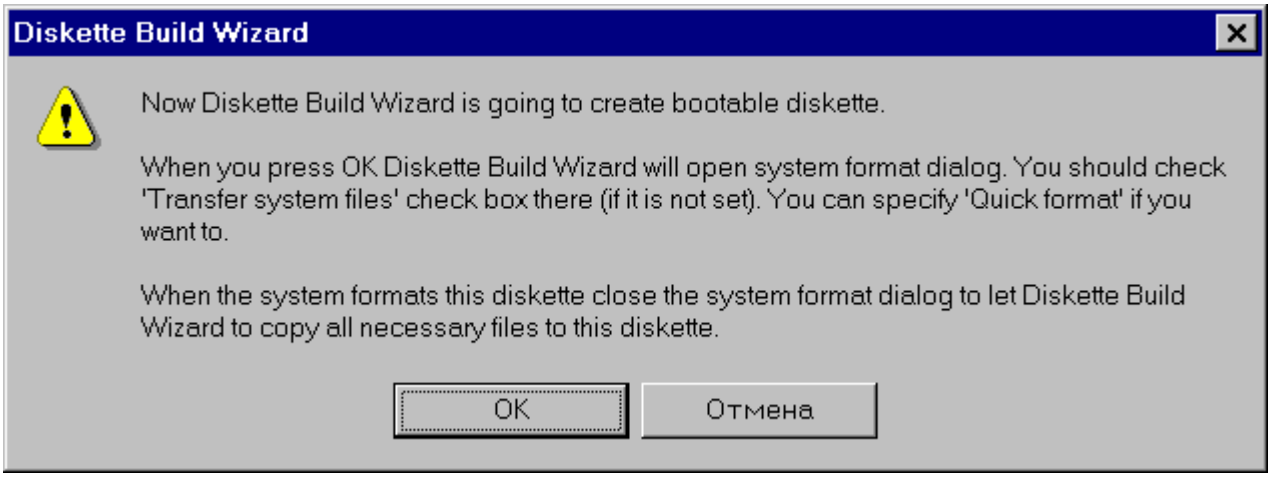
Pic. 4

Diskette Build Wizard will prompt you to confirm that the diskette being inserted is intended for creating diskette with selected component (Partition Manager).



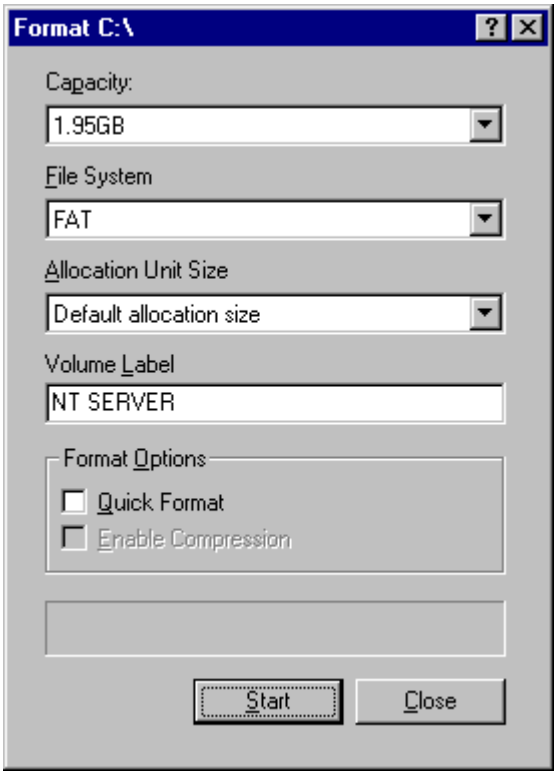
Pic. 5

In the case of not formatted diskette, the dialog (pic.6) will appear till there will be acceptable diskette in the floppy drive.



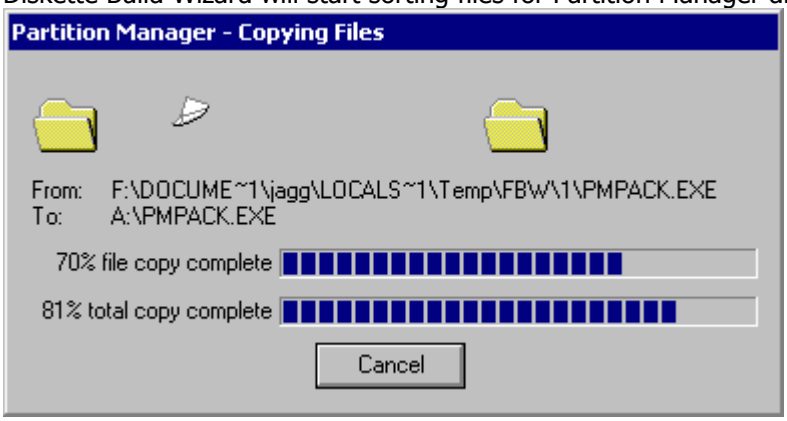
Pic. 6

Bootable diskette should contain all files necessary for booting. Hence, the first stage of creating diskette is copying of system files by means of standard Windows formatting procedure. If you use new formatted diskette, then it will be sufficient just to copy system files (pic.7). In the case the diskette has had any information on it, you can use the procedure of quick formatting (in this case the contents of diskette will just being erased) or the procedure of full formatting (in this case all the file system structure will be created again). All these operations are possible in the system format dialog.



Pic. 7

After formatting or copying system files is completed close system format dialog by pressing 'Close' button, and Diskette Build Wizard will start sorting files for Partition Manager diskette.



Pic. 8

After completing copying file of Partition Manager, Diskette Build Wizard will prompt to remove the diskette and label it as 'Partition Manager bootable diskette'.

Pic. 9

Creating bootable diskettes under Windows ME/NT/2000

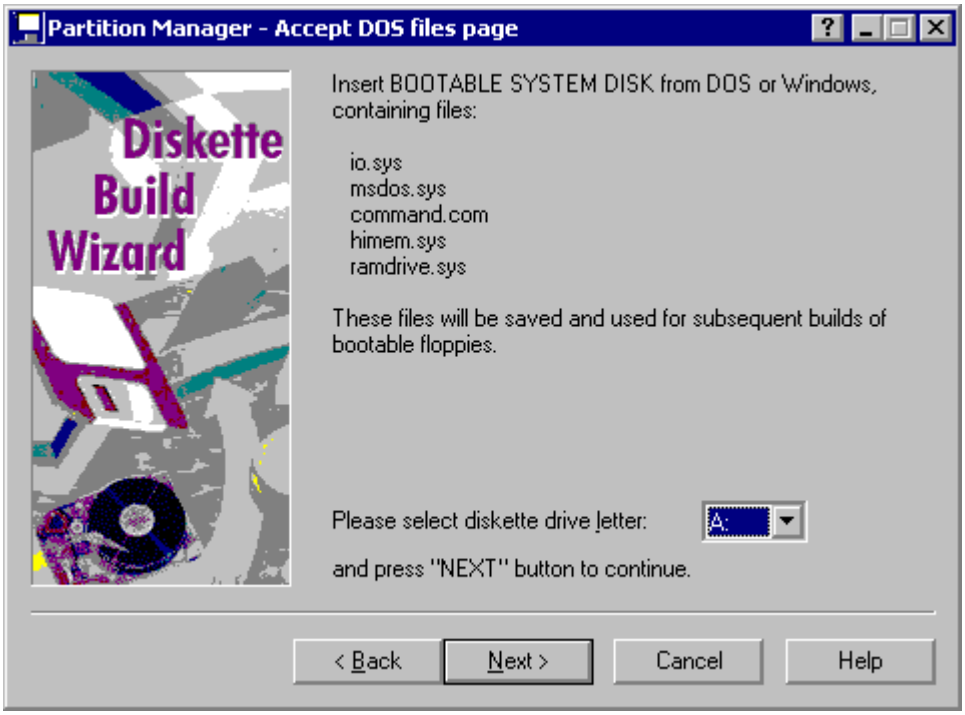
Under these operating systems bootable diskette can be created only during Windows installation as 'Rescue Disk'. Also, you can use system diskettes created under DOS and Windows 3.x for creating bootable diskette with Partition Manager.

Such a diskette should contain following files:

2 Io.sys

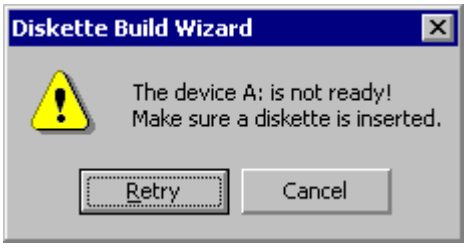
3 Msdos.sys
4 Command.com
5 Himem.sys
6 Mscdex.exe
7 Ramdrive.exe

Information concerning this matter will be shown after you have chosen the component, with which you will create bootable diskette.



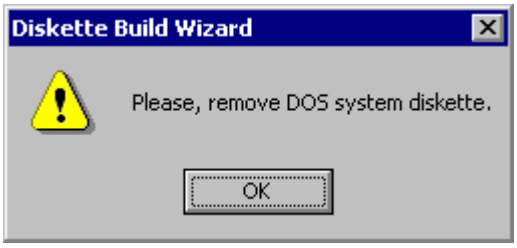
Pic. 10

Then you should insert the diskette with above mentioned files in the floppy drive and press 'Next'. In the case there will be no diskette in the floppy drive or the diskette will be invalid, Diskette Build Wizard will show warning message.



Pic. 11

Check the diskette is inserted into the floppy drive and press 'Retry'.
After copying system files from the diskette will be completed, Diskette Build Wizard will show warning message:



Pic. 12

As in the case of Windows 9.x, you can create ad lib bootable diskettes, as Diskette Build Wizard will return to the dialog of creating the diskette (see pic.4).

1.5 Program Execution

Dependent on the program version of PM; whether DOS or Windows and of the OS version.

1.5.1 Running the Windows version

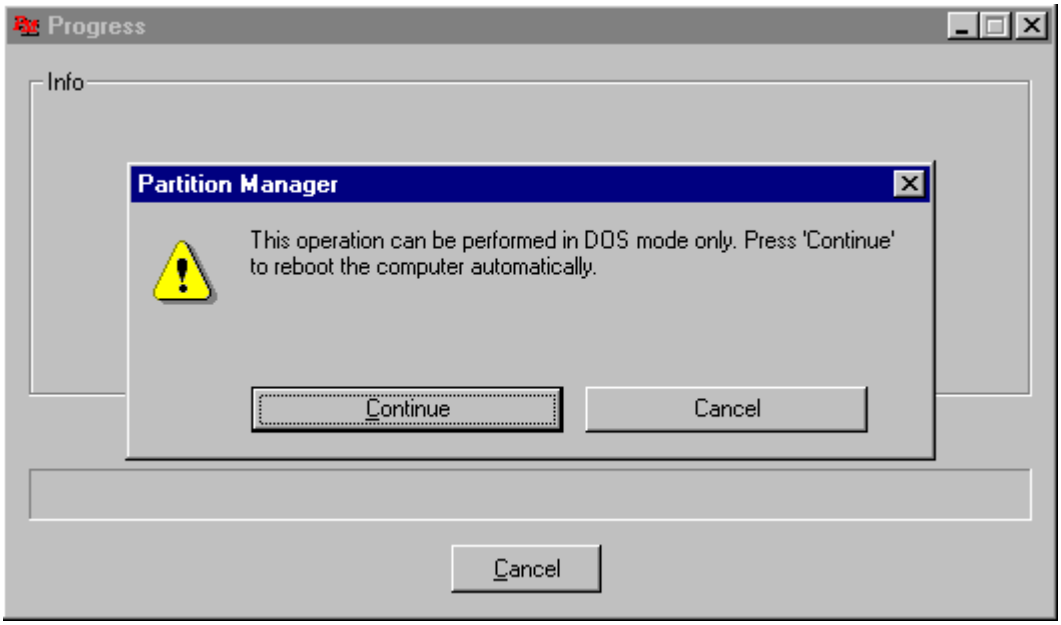
The windows version must of course be run from Windows 95, Windows 98 or Windows ME NT or 2000 resp. XP. It's a totally standard Windows application in all respects. If installed, you should find a shortcut icon in Start | Programs, then scroll down to the last entry (most recently installed program group).

You can find all the installed programs in a submenu, including the DOS version (not in WIN ME, XP, NT and 2000) as well as Floppy Build Wizard.
Note: When running the Windows version there should be no other applications active/open, otherwise some functions like resizing may not work. The only applications that should really be active at this time are the basic Windows system functions and Partition Manager.
See the following chapter 1.5.2 Windows – Operation on boot partition

1.5.2 Windows versions – Operations on active boot partition

All operations that require exclusive access to the whole partition can not be done under Windows if the boot partition is the operation target. On this partition the multi task Windows OS has open files which can not be closed. The functions resize, copy, move and converting can therefore never be executed under window for the partition from which the system was booted. If you select such an operation under windows for this partition (or any other partition with an open file) the program offers to do this operation after a reboot only.

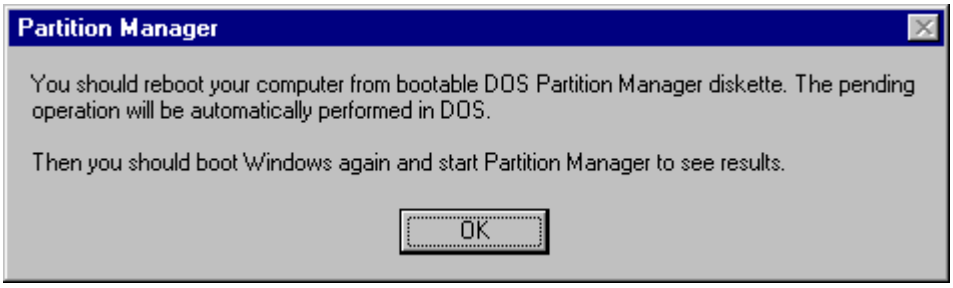
Windows 95/98
These 2 versions come with an internal DOS and windows may therefor be rebooted in its single task DOS mode.



Windows ME
Because this Window version does not have any DOS component at all there is no menu item "DOS version" in the PM Window submenu.

Under Windows ME, selecting an operation in PM that requires a reboot under DOS will display the message that this has be done in DOS mode. Contrary to Windows 95/98 you are then asked to insert the bootable DOS Diskette with PM to proceed.

Insert that bootable Diskette with PM before you press OK



Windows NT, 2000 and Windows XP
Because these Windows version do not have a DOS component at all there is no menu item "DOS version" in the PM Windows submenu. But they offer a boot time service in which the required task will be executed.

Under Windows NT, 2000, XP, selecting any operation in PM that requires single task mode (normally DOS mode) will display a message that this operation requires a Reboot.

After the reboot the selected task will be executed by PM as a boot time service (blue screen mode) and when finnished PM will reboot and show following window:

After you select a Resart the changes will become active under this windows version.

1.5.3 Runnning the DOS version

The DOS version of Partition Manager is a (single-task) program and thus executable directly from DOS as well as from the DOS mode of all Windows 9.x but not from WIN ME/NT/2000/XP. The following are the three ways PM DOS can be invoked:

1. Start from a bootable DOS floppy. In this case all functions can be executed without restriction since no file on the hard drive is open. This is the recommended procedure for any type of repartitioning on any partition, also boot partition.

2. Run Partition Manager from a DOS booted from the hard drive. All functions may be executed.

3. When executed from Windows 9x, Windows itself and all Windows applications are automatically shut down and the computer is restarted in basic single-task DOS mode. All functions may be executed.

1.6 Diskette Build Wizard

This program is a small utility (run under Windows OS), which allows you to create bootable diskettes with the DOS version of Partition Manager. This program is completely menu-driven, according to the Windows standard (you can go back to the prior installation dialog at any time).

- Run the program from Windows calling it from the Partition Manager submenu.
- Select the program(s) for which a bootable diskette is to be created, using the radio buttons. Press **Next**.
- Select the floppy drive(s) to be used (if there are more then one) Press the **Build diskette** button.
- DBW prompts you to insert a floppy. Insert the diskette and press OK
- There comes an another message which describes the next steps. Please read it and press OK.
- For Windows 9x only:

In the next step, the options for formatting and system creation appear, Make your choice and press **Start**.
Note: If you already have a formatted disk, select "copy system files only", otherwise select one of the format options (the 2 format options include the system file transfer).
After the formatting (and the system transfer), press **Close** at the statistics results dialog, then press **Close** again to end the format.

- For Windows ME/NT/2000/XP only

Under these operating systems bootable diskette can be created only during Windows installation as 'Rescue Disk'.
You can also use system diskettes created under DOS or Windows 9x to create bootable diskette with Partition Manager.
Such a diskette should contain following files:

Io.sys
Msdos.sys
Command.com
Himem.sys
Mscdex.exe
Ramdrive.exe

Information concerning this matter will be shown by Diskette Build Wizard after you have chosen the component you will create bootable diskette with.
First, DBW will copy necessary files from bootable diskette and prompt you to remove diskette. Then, DBW will prompt to insert the diskette for PM and will copy all necessary files onto it.

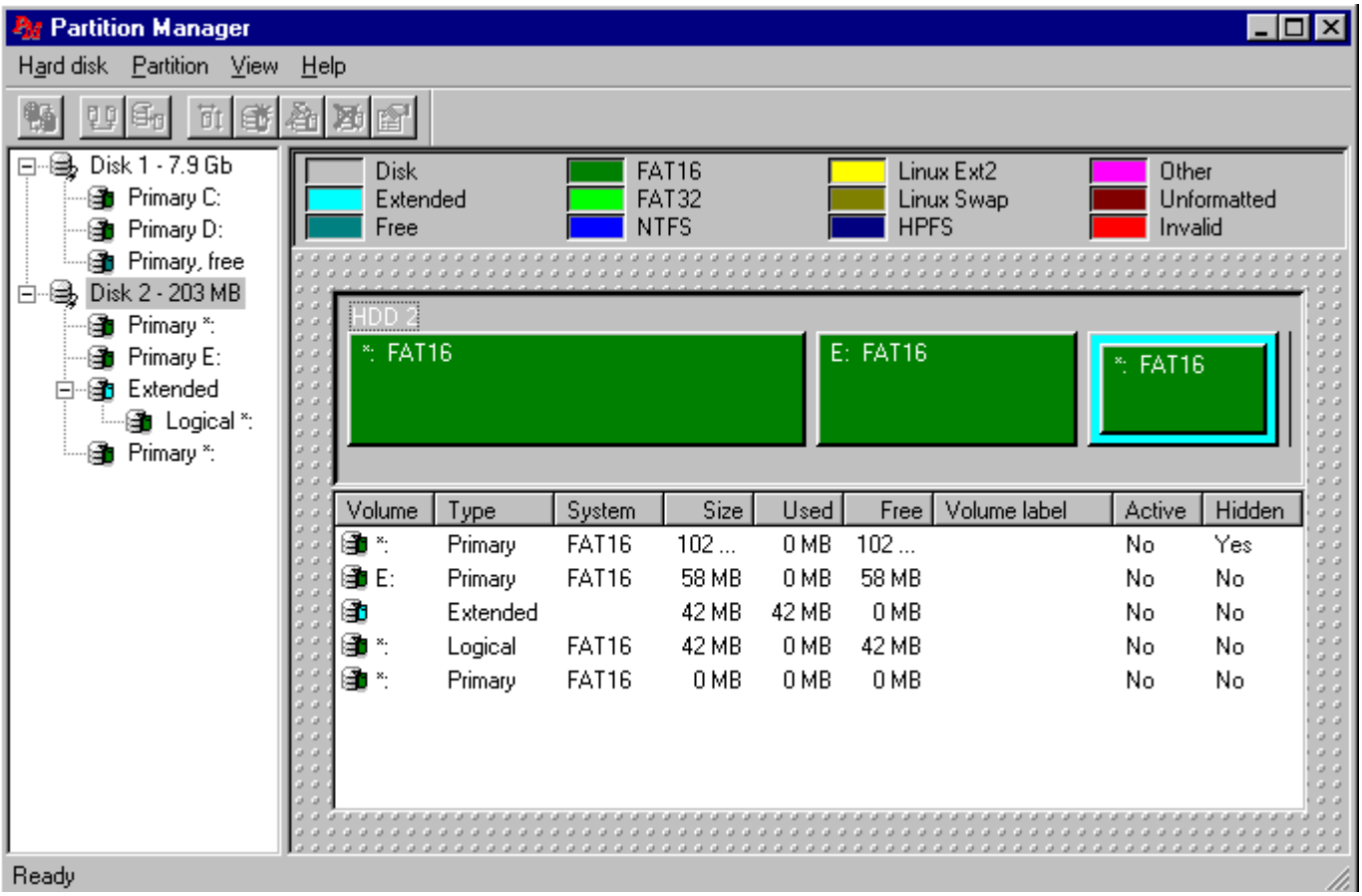
- The selected program will now be transferred. Press **Cancel** if you need to stop the copy.
- You now have the option of again selecting **Create floppy**, to create a second identical copy, or you can select **Finish** to end the program.

2. The Windows version of Partition Manager

Important Note for all Windows versions:
You can never do a resize, copy, converting or move of the boot partition directly under Windows. On Windows 9x the program will ask for a reboot in DOS, in WIN 2000, NT and XP the task will be done in a boot time service (blue screen mode) and in Windows ME you can do this operations only by booting from the PM Diskette (DOS) version.
Please check chapter 1.5.2 Windows – Operation on Boot partition

2.1 The main display

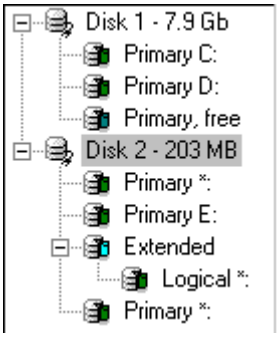
After PM has scanned and detected your hard drives and partitions ***(this may take a while)*** the main screen is displayed.



As you can see, the main display shows all the detected hard drives and partitions, four main pulldowns and eight buttons on the toolbar. These will be explained in detail later. The colored key codes represent the various known types of partitions and statuses they can have.

2.1.1 The left window – hard drives and partitions

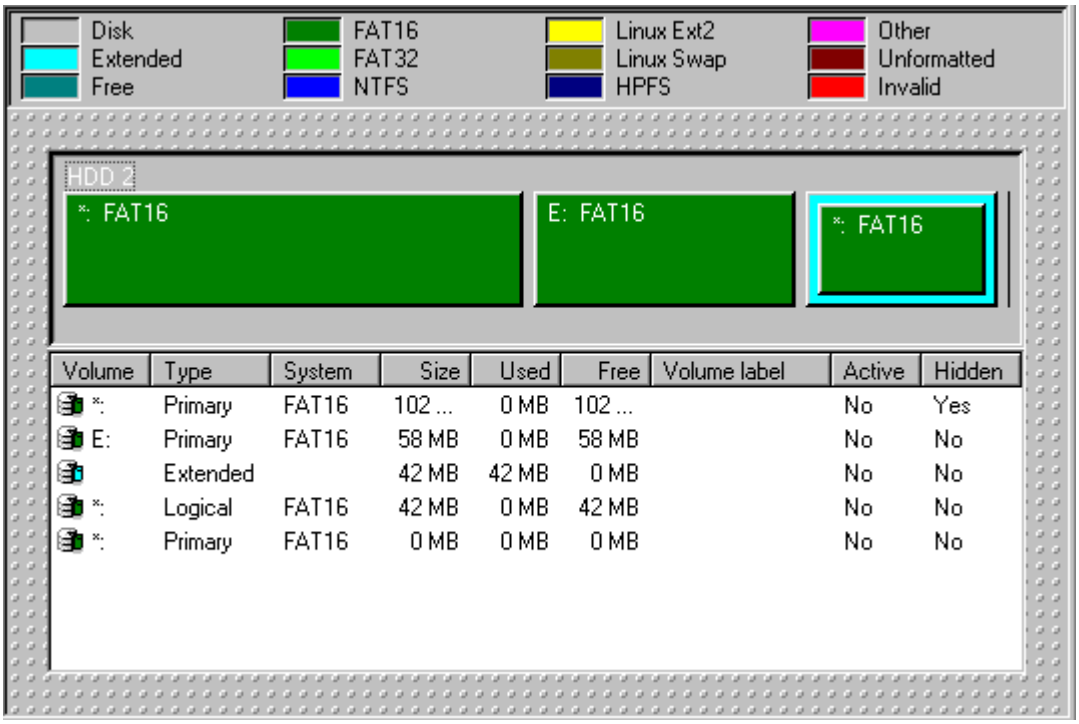
Hard drives, i.e. the physical drives are displayed as a symbol with the labels HDD1 to HDDn, followed by the respective size of the hard drive.



To the left of the hard drive symbol, (pressing of) the zoom box ("+/-") enables one to display the allocation of the hard drive in partitions (+), or only the hard drives (-). This works in the standard expand/collapse tree fashion that is standard in Windows Explorer and other Windows applications. By expanding, the partitions are displayed, by collapsing, only the physical hard drives in the system. In this example, the two hard drives are displayed, along with all their partitions (expanded view).

2.1.2 The right window – partition detail

As noted above, partitions are shown in the left window, underneath the drives in which they reside (expanded tree). In this, the right window, the partitions are displayed as a list, along with detailed specifications about them, and directly above the list, you'll note a graphical representation of the various partitions in the selected drive. The color of the partition represents its type or status as defined by the keys directly above that.



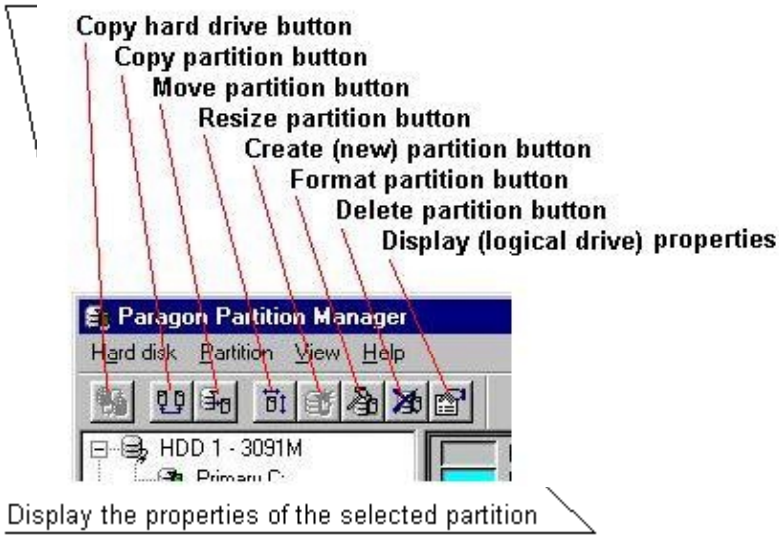
A partition is selected by one of the following:

- clicking the partition in the list in the right window
- clicking the graphical representation of the partition above the list
- clicking the partition in the left window

Also, right-clicking any of these presents all functions that apply to the selected partition, same as you would get from the partition pulldown off the main menu bar.

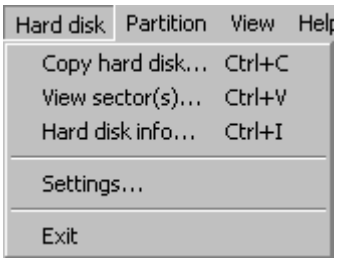
2.1.3 The toolbar

The toolbar has the following buttons:



2.2 The *Hard disk* pulldown

This pulldown menu allows you to copy entire drives, change *settings* and exit the program.



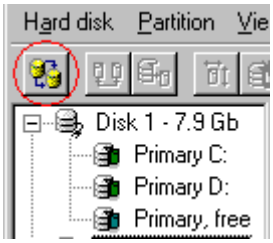
Since in this case there are two hard drives in the system, the copy option is valid and executable. If there were only one drive, this option would be greyed-out. Throughout this document, the terms valid and active may be used interchangeably, when referring to these kinds of situations, where the option may be greyed-out or not valid in some other sense. Furthermore, active in this sense should not be confused with the term active as it relates to a bootable drive.

2.2.1 COPY HARD DRIVE (Hard disk | Copy hard disk ...)

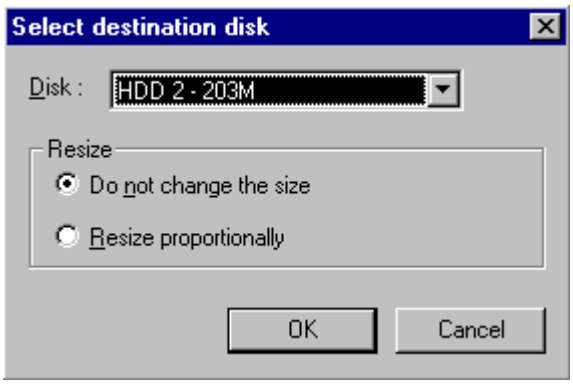
Shortcut for copying whole drives: Button number one on the toolbar.

This menu option is only active (and relevant) if at least two hard drives are available. The drive to be copied must in any case always be selected by clicking on it (in the left window), or as a shortcut, you can right-click the drive and then PM will ask you for the target drive. Or, after first selecting the drive:

a) Select the copy function from the hard drive pulldown, as mentioned in the previous paragraph.
b) Press the copy button on the toolbar (leftmost button).



If there's a second drive available to copy to, PM asks you to select the target drive using the following dialog:

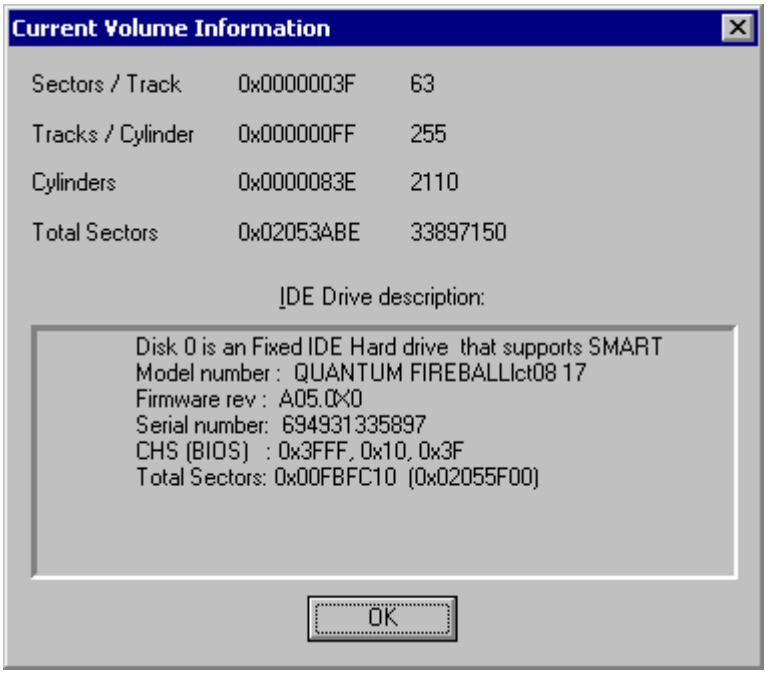


Select the target drive from the pulldown menu. If the target is not displayed, press the downarrow on the rightmost of the drive selection window for a list of all the available drives.

Note: The option "Resize proportionately" means that the source drive (with all its partitions) is adapted to the size of the target drive in a way that resizes each partition in an equal way. If the target drive is 2.3 times the size of the source, each partition will be enlarged by a factor of 2.3.

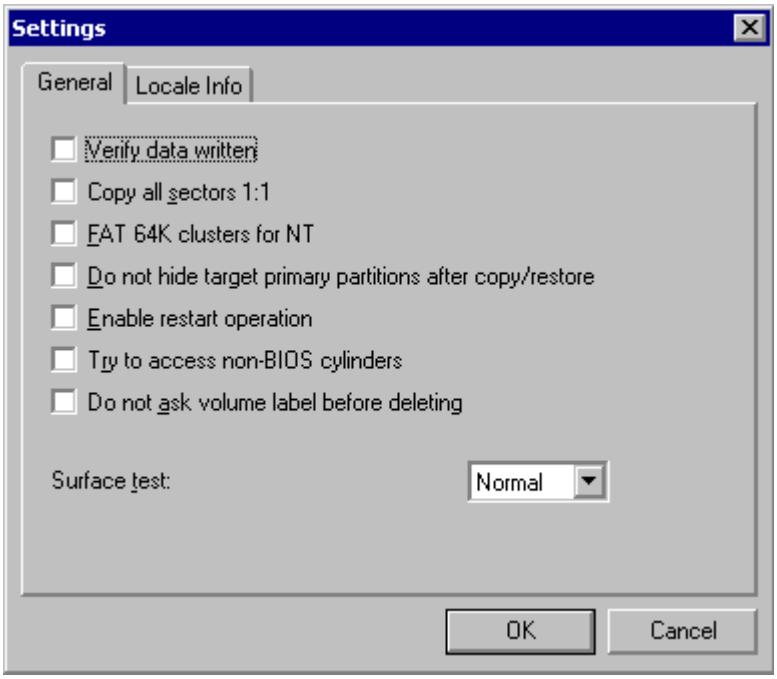
2.2.2 HARD DISK INFO (Hard disk | Hard disk Info)

This shows the low level formating information about the selected hard disk as there are: number of sectors per track, number of tracks per cylinder, the number of cylinders and the total number of sectors.



2.2.3 CHANGE PM SETTINGS (Hard disk | Settings ...)

These are general parameters that apply to many program functions. Explained in detail below.



Verify data written

Regulates the verification status of any PM process which does any writing. Whenever this parameter is set on (active), a verification (comparison) is done between the data read and the data written to ensure they are the same. This is done *at the sector level.* Although this slows these processes down considerably, it ensures positively that no errors will occur during the copy.

Copy all sectors 1:1 (sector-by-sector copy)

If the box is checked, the copy process bypasses the normal (logical) copy process which goes through the operating system and performs instead a basic physical, (sector-by-sector) copy which transfers bytes exactly, without regard to the content.

FAT 64K clusters for NT

This option allows the creation of a 64 KB cluster size for FAT 16 partitions. This achieves the doubling of the upper sector limit from 64 sectors (512 bytes) to 128 sectors. Partitions converted/copied in such a way will subsequently be accessible only by Windows NT, however these will then no longer be recognizable by DOS. Partition manager detects both cluster sizes and creates (in the standard mode) a 32 KB cluster size. Only if this parameter is set, is a 64 KB cluster size created.

Do not hide target primary partition after copy/restore

In the standard installation the new copy of a partition (the target partition) is automatically hidden. The reason for this is that the logical drive designators are the same as before the copy process. When the parameter is set, it makes the target partition immediately visible from the start, which subjects the drive designator sequence to change and can lead to confusion.

Try to access non-BIOS cylinders

This function is relevant only if there are operating systems such as Windows NT, Windows 2000 and some newer Linux versions installed. These operating systems bypass the PC BIOS and use their own internal BIOS to access cylinders (invisible to the PC-BIOS). If this parameter is set (switched on), Partition Manager will attempt to access these cylinders.

Do not ask volume label before deleting

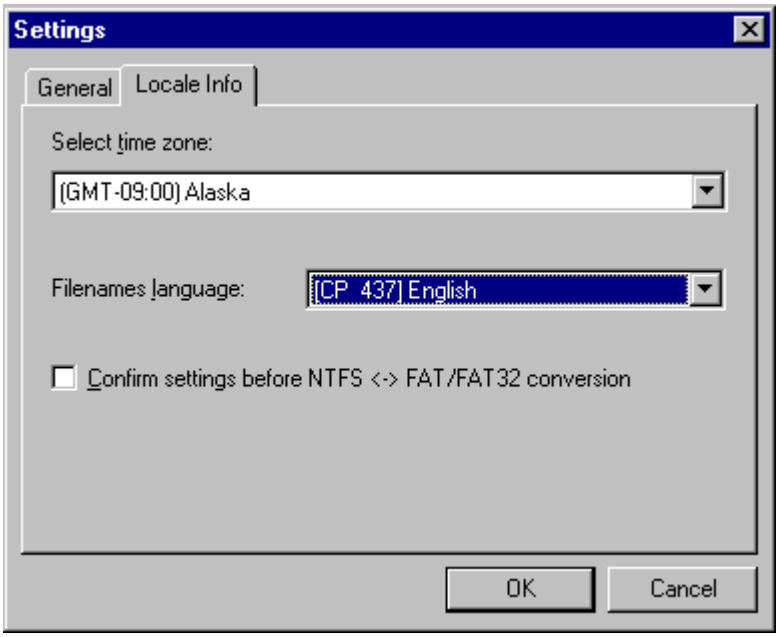
If selecting a partition for deleting you will be asked for the name of that partition(if the volume has not been named, **NO NAME** should be supplied as the volume label). If switched on you do not need to enter the name to proceed in deleting a pratution.

Surface test

By setting of this parameter to one of the various options, the usability (physical state) of the disk space involved in a partition **create, copy, move** or **format** operation will be verified.

Available settings and their effect:
None: No physical examination of the disk space is performed.
Normal: Data on sectors written is compared to the sectors read. A standard read/write verify of each sector in the partition occurs.
Extreme: Read and written sectors are re-read and compared a second time.

The standard (and recommended) setting is **Normal**. The **None** setting is the quickest but will cause errors if it encounters defective sectors. The e**xtreme** method is recommended only for situations in which multiple defective sectors are anticipated. In this instance, two comparisons will often detect a bad sector where one may not.



On tab 2 (page 2) the localisation table you may choose the time zone and the language font.

Confirm settings before NTFS → FAT/FAT32 conversion
If selected you will be asked to conform the settings of a file conversion from NTFS to FAT.

2.3 The *Partition* pulldown menu

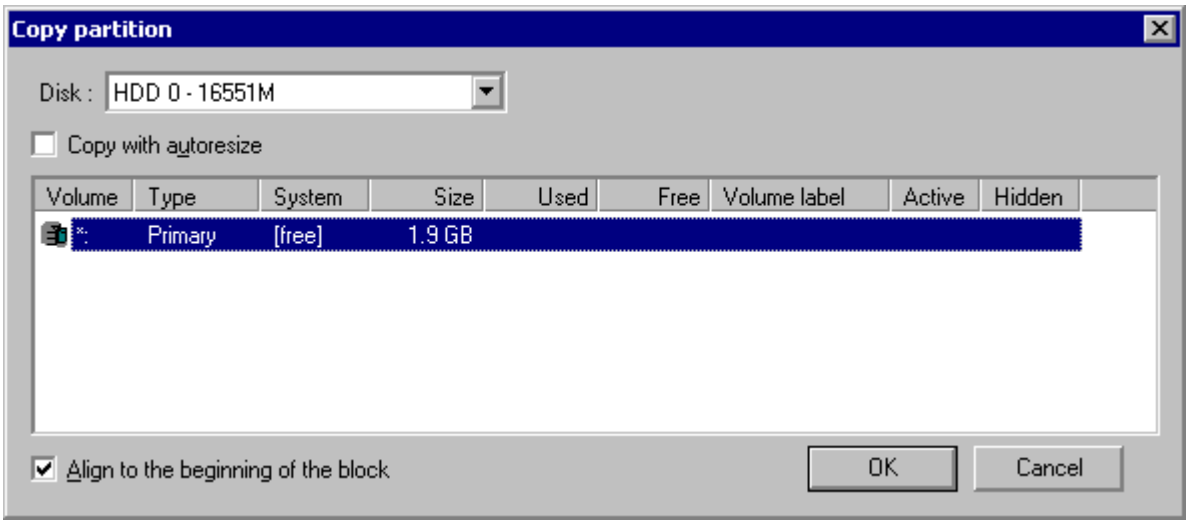
Only options that apply to the currently selected partition will be valid here. If there are no partitions selected, no options will be valid (active).

Note that some of these options/functions are also available as follows:
a) Right-clicking a partition brings up the same options as the pulldown.
b) Some functions are also available on buttons 2 through 7 on the toolbar.

2.3.1 COPY PARTITION (Partition | Copy partition ...)

Shortcut for copying single partitions only: Button number two on the toolbar.
Purpose: Clone a partition to free disk space.

The selected partition is the source(to be copied from) partition. This function is valid only if the source partition is a formatted logical drive (primary, or logical drive within an extended partition). An extended partition can also be selected as source. In this event all logical drives within this partition will copied to the destination as a unit.



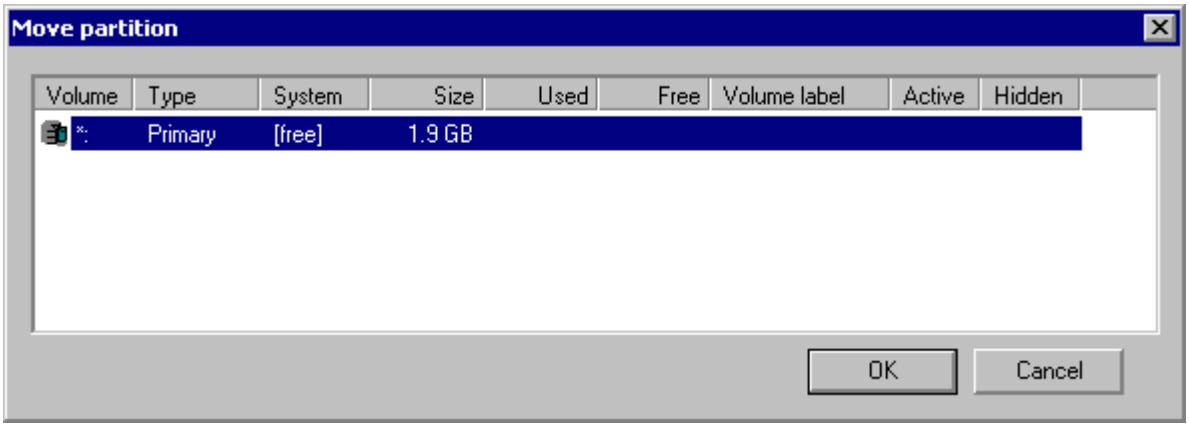
As target, only primary <free> or logical <free> disk space can be used. This empty space must be at least as large as is required to accommodate the data in the source partition. Free disk space as target is selectable from the 1. drive as well as the 2. drive.

Copy can occur as 1:1 without any further specification as well as in the autoresize mode. The Autoresize mode has the ability to adjust the size of the target partition from a minimum (amount of data in the source partition) to a maximum (total size of the target partition) or anywhere in between. This auto-adjusting size (to the target) feature is possible only in partitions using the following file systems: FAT-16, FAT-32, NTFS, HPFS and EXTFS.

The uniqueness of this auto-size adjustment feature lies in the fact that the target partition can be smaller as well as larger than the source partition. The key in the auto-size adjustment feature lies in the size of the source data rather than the size of the source partition. The target partition must of course be at least as large as the data in the source partition, plus a slight security margin.

2.3.2 MOVE PARTITION (Partition | Move partition ...)

Shortcut: Button number three on the toolbar.
Purpose: Move a partition to adjacent free space. The partition to be moved must be selected. This function is only valid if the source partition is a formatted logical drive (primary or logical within an extended partition).



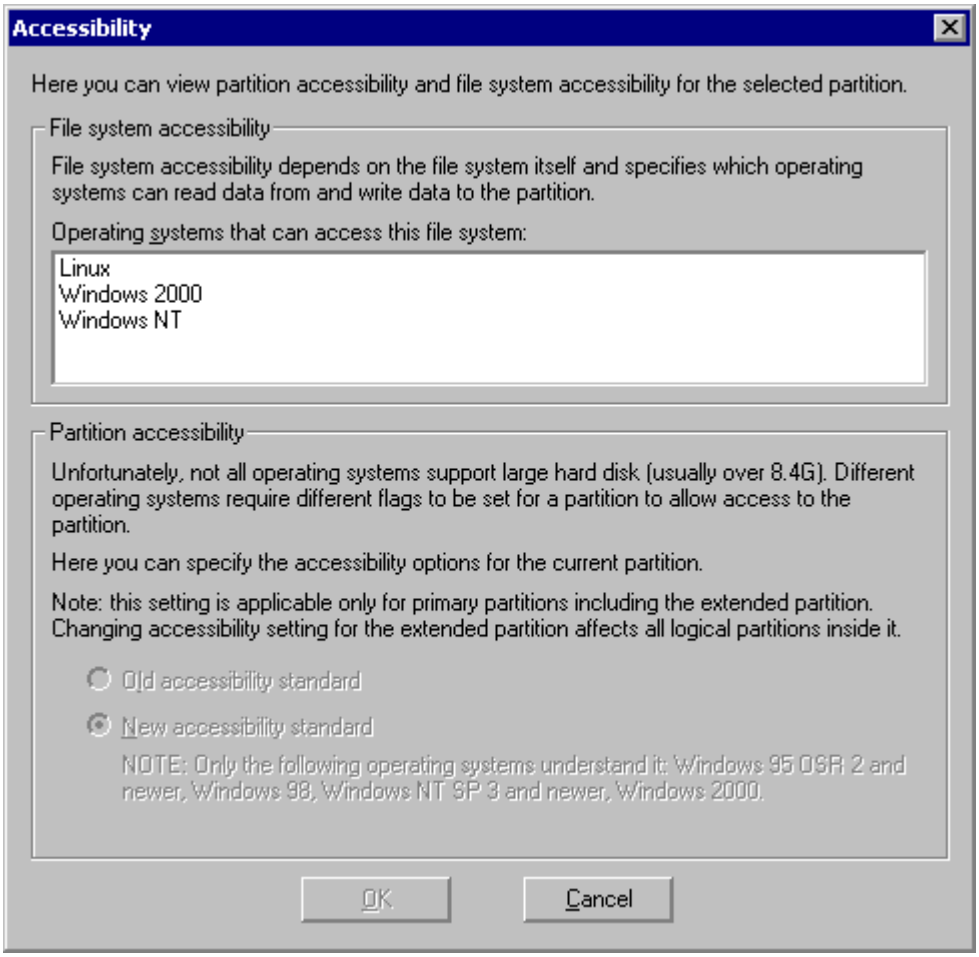
The selected partition can be moved parallel in both directions on the hard drive. This is however only possible if the partition has free space (Primary or Logical <free>), of any size directly before or behind it. If these conditions do not exist, PM will let you know with the message: "*No free entries of requested size on this drive*".

The moved partition will be aligned automaticly to the begin of the free block, if the free block is before the partition or to the end of the free block if the free block is behind the partition. This creates in both cases the biggest possible free block where the moved partition was located.

2.3.3 DISPLAY PARTITION ACCESSIBILITY (Partition | Accessibility ...)

Purpose: Mainly informational only, shows OS-compatibility for partition.

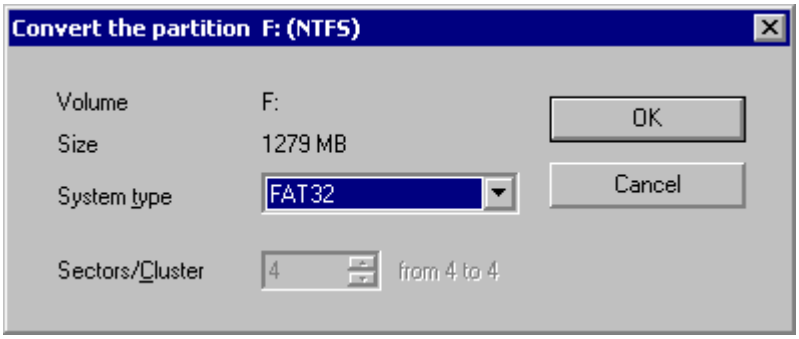
This option is relevant only to formatted partitions. It supplies information regarding which operating systems are able to recognize and use the selected partition. For certain systems, the way the partition ID is recognized can be changed, using the radio buttons **New accessibility standard** and **Old accessibility standard**. The short explanation at the bottom of the dialog box indicates to which operating systems this option is even relevant.



2.3.4 CONVERT FILE SYSTEM (Partition | Convert ...)

Purpose: Convert FAT file systems from one to the other (FAT-16 to FAT-32 and vice versa) and FAT to NTFS and vice versa.

A conversion is relevant only if the installed OS systems supports the new target file system. While all OS understand FAT-16, the FAT-32 systems is supported by Windows 95 (release B), Windows 98 ME, WIN 2000 and Windows XP only. The NTFS file system is supported by WIN NT4, WIN 2000 and Windows XP.



This function is valid only for logical drives with the FAT-16 or FAT-32 or NTFS file system. The conversion offers for a selected FAT-16 drive FAT-32 and NTFS, for a selected FAT-32 drive NTFS and FAT-16(*) and for a selected NTFS drive FAT-32 and FAT 16(*). (*) FAT-16 is offered only if the source drive is below 2 GB.

The following restrictions apply:

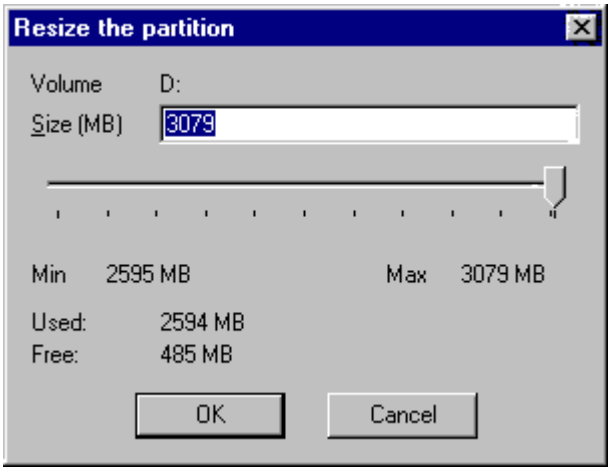
- The minimum size for a FAT-16 partition to be converted to FAT-32 is 300 MB.
- The maximum size from FAT-32 to FAT-16 is 2.0 GB.

2.3.5 RESIZE PARTITION (Partition | Resize ...)

Shortcut: Button number four on the toolbar.
Purpose: Increase or reduce logical drives, dynamically resize entire extended partitions in both directions.

This function is valid for all primary partitions and logical drives within an extended partition, as well as for the extended partition as a whole.

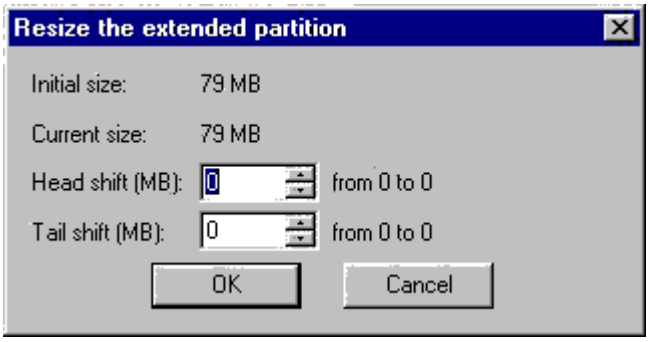
Screen for primary partition or single logical drive:



In the case of a primary partition or a logical drive, only the end of the partition can be moved, either outward (increase) – provided there's enough free space - or inward (decrease) - only as small as the data area. In either case, simply move the pointer to indicate the desired resulting size, within the two limits:
Maximum = size of the partition plus possible free space following it.
Minimum = size of the data within the partition.

The pointer, of course will keep the desired selection within these boundaries.

Resizing an extended partition shows the following dialog:



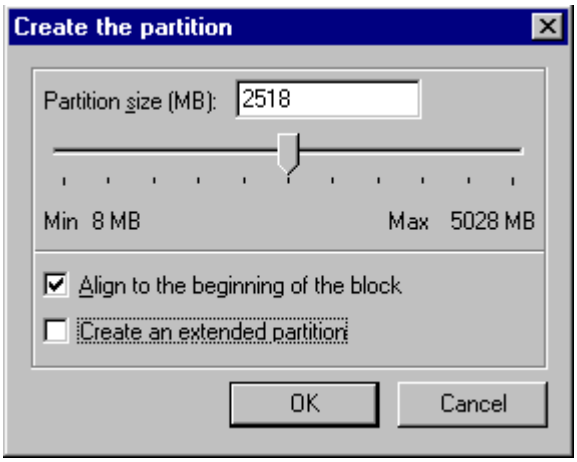
For an extended partition, the size can be dynamically adjusted at either end, in both directions. In order to increase the size in any direction, naturally free space has to exist there. An inward resizing can occur provided there is no data in that region of the partition. PM automatically determines the resizing limits in both directions, for both ends of the partition and these are shown after the **Head shift** and **Tail shift** windows. The **from x to n** values displayed after each of these windows represent the possible change for either the head or the tail of the partition. A negative value represents an inward shift of the tail (size reduction), or a outward shift of the head (size increase in that direction) and a positive number an outward shift of the tail (size increase in that direction) or in inward shift of the head (size reduction).

Once the desired increase or decrease for head and/or tail are determined, you can either explicitly enter values in the two windows, or use the up and downarrow keys to increment or decrement the desired change by one single MB.

2.3.6 CREATE PARTITION (Partition | Create ...)

Shortcut: Button number five on the toolbar.
Purpose: Creates a new partition in free disk space.

This function is valid only if either a primary or logical free space has first been selected with a mouse-click. Together with the copy function, this is the function that allows you to create new partitions (in any case by copying, new partitions are created with the contents of the old partition).



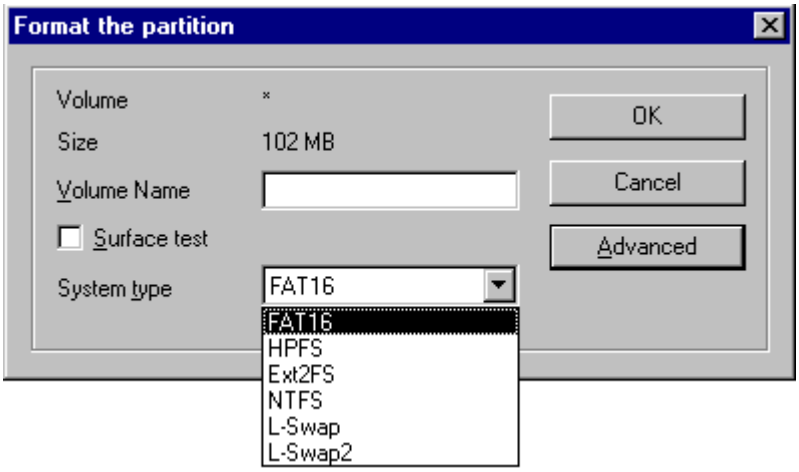
Use this to create primary partitions, extended partitions or logical drives within an extended partition. If you selected primary free disk space, you can create either a primary partition or an extended partition, however an extended partition can only be created if no extended partition already exists. Within an extended partition only logical drives can be created. The desired size of the partition to be created can be easily adjusted with the slider control (see screenshot above).

Align to the beginning of the block option (see screenshot):
When the size of the free space is larger than the desired partition size, checking this box ensures that the partition you create will be at the very beginning of the free space. **If the box is unchecked**, the partition will be placed *at the **end** of the free space.*

2.3.7 FORMAT PARTITION (Partition | Format ...)

Shortcut: Button number six on the toolbar.
Purpose: Create a file system for a partition or logical drive. **Destroys previous contents.**

Critical:
It's possible to format both unformatted partitions, as well as already formatted partitions containing valid data. If a partition having data on it is formatted, the addressability of the data contained therein is **completely destroyed**. ***Before selecting the format option, be absolutely certain you've selected the correct partition to be formatted.***



This function is valid for all primary partitions and for all logical drives within an extended partition.

A file **System type** must be selected for the partition to be formatted. There's a nice table in section 1.3.7 that explains not only which file systems can be used with which operating systems, but also shows the extent of PM's capabilities using each.

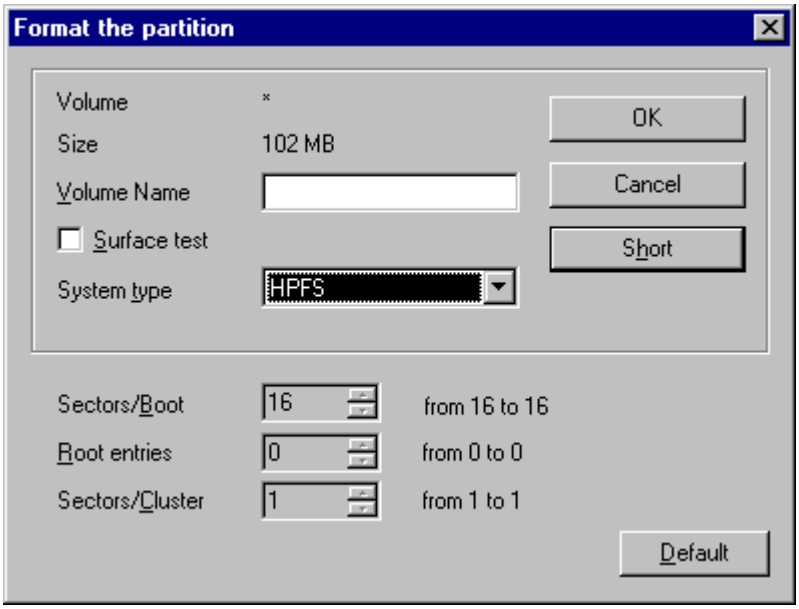
The file system to be used in a given partition is limited by the partition size and the type of partition. (For example, no FAT-16 formatting is possible in a 4 GB partition.)

You can specify a **Volume Name** now or later. If a surface test is desired, check the **Surface test** option box (for detail see section **2.2.3 Hard disk | Settings** … *surface test*). Press OK to initiate the format.

Since formatting destroys the usability of all data in the partition, PM confirms: **"Format Partition, Are you sure ?"**. The format is not initiated until you answer ***Yes*** to this dialog. This confirmation is used to ensure that you:
a) are sure you want to format this partition, and …
b) have selected the correct partition to be formatted.

By pressing the **Advanced** button you can expand the format dialog to include advanced (count) parameters such as **Sectors/Boot**, **Root entries** and **Sectors/Cluster**. These are considered expert options.

Caution: These options should be used only when one knows the exact consequences the selected parameters will have upon the new partition's file system.



Sectors/Boot
This is the number of sectors that are required for the boot area, normally only one single sector. Apart from your own objectives (for your specific system programming), there are certain instances in which more than one boot sector are used. (Some embedded systems have several "reserved" boot sectors.)

Root Entries
This is the number of sectors reserved for the root directory. Apart from your own objectives (for example simply having a large number of free sectors at the beginning of a partition) this is important for FAT-16 partitions if you plan to have either many files or a lot of long file names in the root directory.

Sectors/Cluster
Depending on the file system used, and the selected size of the partition, the cluster size can be manually defined here. For a FAT-16 system, if the partition is small enough, even a very small cluster size of 1:1 is possible (1 cluster = 1 sector = 512 bytes) For more information on the FAT system, please see the section on "Fundamentals".

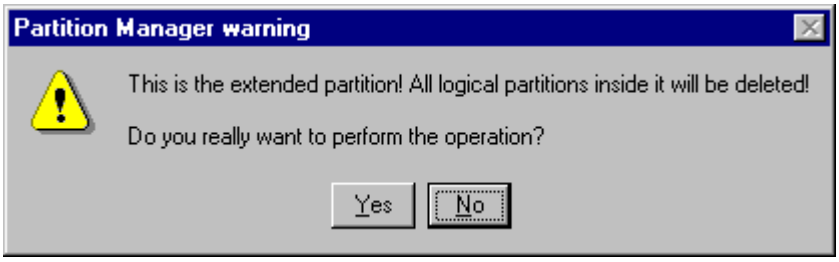
2.3.8 DELETE PARTITION (Partition | Delete)

Shortcut: Button number seven on the toolbar.
Purpose: Deleting partitions and creating new free disk space.
This function is valid for all primary and extended partitions as well as for all logical drives within an extended partition. The deletion of an extended partition also deletes all logical drives within the partition.

Note: Deleted partitions are converted to raw disk space, which are denoted as primary or logical free by Partition Manager. All data within a deleted partition is lost. Partition Manager therefore confirms the partition to be deleted by asking you to supply the volume label of the partition to be deleted. If the volume has not been named, **NO NAME** should be supplied as the volume label.

If you are certain you wish to delete the specified partition, answer yes to whichever is displayed:

OR

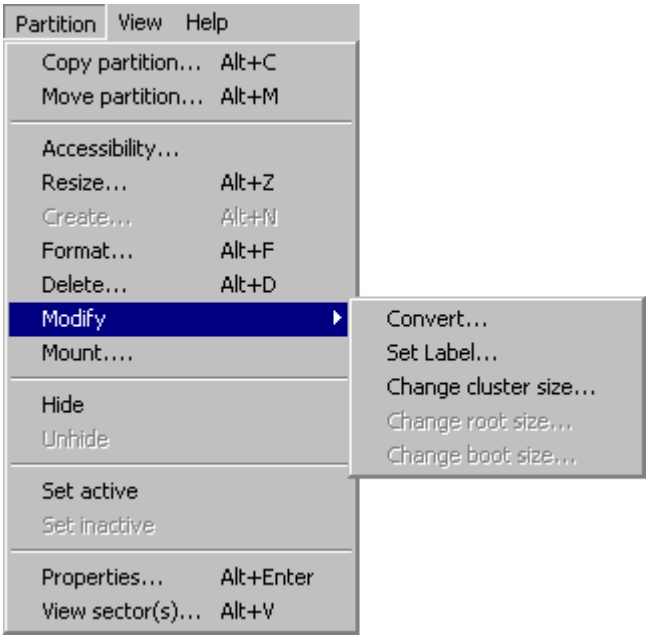


This confirmation ensures that you:
a) Really want to delete the selected partition.
b) Have selected the correct partition to be deleted.

2.3.9 MODIFY PARTITION (Partition | Modify)

Purpose: Change partition parameters

These parameters are reserved for the use of experts, since expert knowledge of the file system is required, in order to accurately predict the effects of manually made modifications and variations available here. In theory everything can be adjusted here, including enabling FAT-16 to FAT-32 conversion (and vice-versa). Some of the options should really only be used when one knows the exact consequences the newly selected parameters will have upon the partition's file system.



Volume
Logical drive Letter

Volume Label
This is the name identifier for the partition in question. **This entry is case sensitive**. This label is used by any and all critical operations that deal with the partition. Whatever you enter here will be used to identify this volume, at the base partition level as well as in your operating system. In Windows 95/98, for example, if you right-click the volume (logical drive) and do **properties** the volume name will be displayed, along with other data, such as the size, whether it's a local hard drive, etc.

System Type
Defines the file-system type of the selected partition. Be extremely careful when using this choice for a formatted partition.

Used (MB)
For informational purposes only. Displays the total amount of data in MB which resides in this partition. This is also the minimum size to which this partition can be resized.

Free (MB)
For informational purposes only. The amount of free (unused) space in this partition

Size (MB)
We recommend to use the Resize Partition option.
This input is changeable only when a formatted partition is selected. To the right of the data, the potential input range for the size is displayed.
Minimum size: This is the size occupied by existing data in the partition.
Maximum size: Either the current size of the partition, or if there's free space directly following the end of the partition, the current size of the partition plus the size of this free space.

Sect./Boot
This is the number of sectors that are required for the boot area, normally only one single sector. Apart from your own objectives (for your specific system programming), there are certain instances in which more than one boot sector are used. (Some embedded systems have several "reserved" boot sectors.)

Root Entries
This is the number of sectors reserved for the root directory. Apart from your own objectives (for example simply having a large number of free sectors at the beginning of a partition) this is important for FAT-16 partitions if you plan to have either many files or a lot of long file names in the root directory.

Sect./Cluster

Depending on the file system used, and the selected size of the partition, the cluster size can be manually defined here. For a FAT-16 system, if the partition is small enough, even a very small cluster size of 1:1 is possible (1 cluster = 1 sector = 512 bytes) For more information on the FAT system, please see the section on "**Fundamentals**".

2.3.10 HIDE PARTITION (Partition | Hide)

Purpose: Make a visible partition invisible to the operating system.

This function is valid for all partition types including unformatted partitions, but doesn't apply to free disk space (primary or logical <free>). It allows you to make logical drives or entire extended partitions invisible, i.e. operating systems will not see them any longer.

This function will be active only for partitions not already hidden.

Note: To hide a partition, PM assigns an invalid partition-ID to it. An invalid partition-ID is an ID that is not recognized by any OS (or the MBR-code), therefore it is not visible.

2.3.11 UNHIDE PARTITION (Partition | Unhide)

Purpose: Make invisible partitions visible again to operating systems.

This function is only active if a hidden logical drive or an extended partition is selected. The function **Unhide** makes hidden logical drives or whole extended partitions visible again to operating systems.

2.3.12 SET PARTITION ACTIVE (Partition | Set active)

Purpose: Make a primary partition bootable.

This function applies only to primary partitions that are not already set active (bootable). A primary partition must therefore be selected prior to attempting to execute this function. Providing an operating is already installed on the partition in question, the partition will after this function be entirely bootable. If not, the OS may be installed after making the partition active. Furthermore, setting a partition active enables the installation program of any OS to see this partition and install its OS into it.

2.3.13 SET PARTITION INACTIVE (Partition | Set inactive)

Purpose: Disable the bootability of a primary partition which is currently bootable.
This function applies only to primary partitions that are currently active (bootable). By setting the partition inactive, this makes the partition not bootable. Furthermore, the MBR, Boot Managers and the installation programs of OSes will no longer identify this partition as a valid boot partition to start or install operating systems.

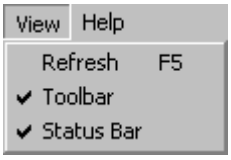
2.3.14 DISPLAY PARTITION PROPERTIES (Partition | Properties)

Shortcut: Button number eight on the toolbar.
Purpose: Displays information about logical drives, such as file system, total space, space used, etc. The space used/free is represented as a pie chart.

This function is only valid when a logical drive is selected (either primary or logical inside an extended partition).

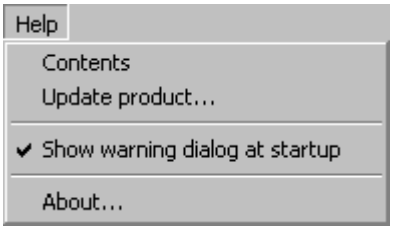
2.4 The view pulldown

Like other Windows applications, the toolbar and status line can be hidden. By clicking either of the options on this pulldown, a checkmark will appear to the left of that option and that option will be made active. If it's currently active (checked), it will be turned off. To turn the toolbar or the status line off or on, simply click its option on this pulldown and it will change to the opposite state (hide/unhide).



2.5 The help pulldown

This combines Windows help, an update program function, copyright and support information, and contact information. Selecting ***contents*** invokes the Windows help function.

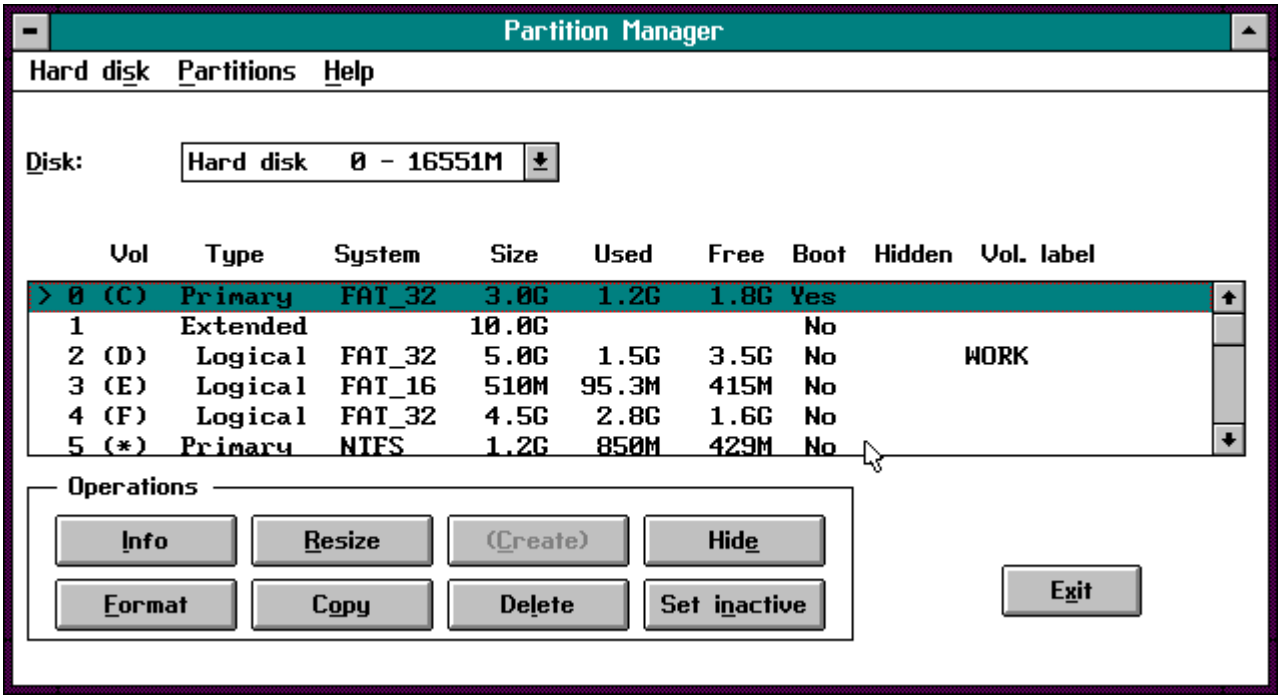


3. The DOS version of Partition Manager

The main display shows a list of the detected hard drives and partitions as well as the partition manager's main functions (buttons).

3.1 Display after successful analysis of the available hard drives

This display appears after each call to Partition Manager, whether running PM from Windows (restart of Windows in single-task DOS mode) or directly from DOS (booted from the hard drive or from floppy).



In this example, the partitions of the first hard drive are displayed. If partitions on another hard drive are to be processed, that drive must first be selected using the ***Drives*** pulldown.

Hard drive selection:
If there is more than one hard drive available, the desired drive should first be selected from the **Drives** pulldown. To do this, first click the *downarrow* on the rightmost side of the **Drives** window. When the choices are displayed, click on the desired drive.

Partition selection:
A partition is selected by moving the highlight-line to the desired partition using the mouse or the up and downarrow (cursor) keys.

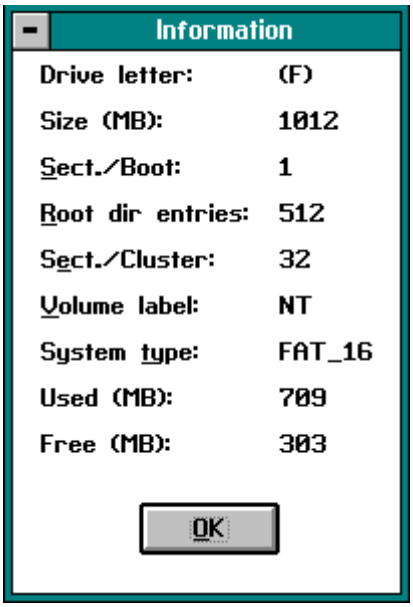
3.2 The function buttons

Note: Throughout this document, the words active and valid may be used interchangeably to denote whether or not a function applies to the current situation (selected partition). In the case of not active, the button, or menu selection will be greyed-out and not be selectable. The term active as used here should not be confused with the term active as it applies to a bootable partition.

The (actively) displayed buttons apply only to the currently highlighted partition (illuminated by the highlight-bar). The exception here is **Exit**, which is always active, and is used to shut down PM DOS and return control to Windows.

3.2.1 Info Button

Purpose: Display statistical information about the selected partition.



This button performs no action on the hard drive(s), only displays information about the selected partition/volume. Example screen:

Info abbreviations defe

Vol	= Volume, the logical drive designator (at the operating system level).
Size (MB)	= total size of the partition in Megabytes.
Sect./Boot	= number of sectors in this partition reserved for system boot records.
Root Entries	= number of sectors reserved for data and directories in the root directory (/).
Sect./Cluster	= number of sectors per cluster.
Volume Label	= name of the partition, if defined. Used by various functions to confirm deletion.
System Type	= file system that the partition was formatted in, if any.
Used (MB)	= number of Megabytes of partition occupied by data.
Free (MB)	= available free space in the partition, in Megabytes.
Access. For OS	= Which OS understand this file system.

3.2.2 Hide/Unhide Button

Purpose: Make partitions invisible or recognizable to the system. A toggle switch.

Switches the partition between the states hidden and not hidden. If the highlight-bar is on a non-hidden partition, this button will be marked **Hide**. This button will be marked **Unhide** if the partition is currently hidden (**<hid>** is displayed in the hidden column, on the detail line, far right). This button is functional with all partition types including unformatted partitions. A hidden partition is not recognizable to the system at all.

This function doesn't apply to free disk space (primary or logical <free>) and is thus inactive on these hard drive segments.

Note: To hide a partition, PM assigns an invalid partition-ID to it, which is one that is not recognized by any OS (or the MBR-code), thus making the partition "invisible".

3.2.3 Set Active/Set Inactive Button

Note: The term **active** in this context does not refer to button functionality, but partition bootability.

Purpose: Tells the partition table this partition is to be recognized as bootable **(active).**
Actually becomes bootable if an operating system is installed on it.

Also a toggle switch. If the highlight-bar is on an active partition (**Yes** in the **Bootable** column), this button will be marked **Set Inactive**. If the partition is inactive, the button will be marked **Set Active**. This function applies only to primary partitions. A primary, active partition is regarded as a bootable partition by the master boot record. Only one active partition per hard drive is ever allowed. Whenever that drive is booted, the system will boot the OS in this partition.

Note: This is a flag stored in the partition table. It tells it whether to consider the partition bootable.

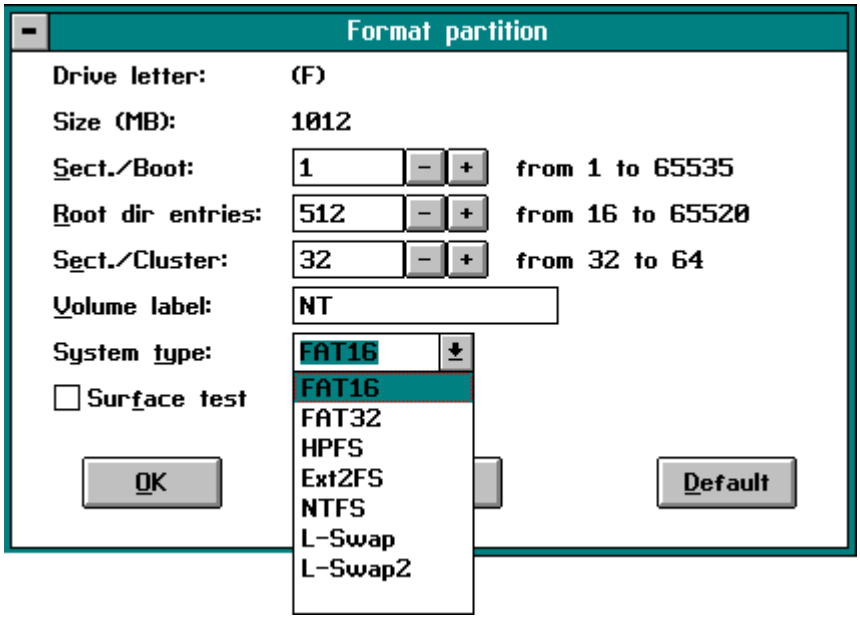
3.2.4 Format Button

Purpose: Prepare a partition with a specific file system. *Destroys the addressability of all previous data.*

Critical: It's possible to format both unformatted partitions, as well as already formatted partitions containing valid data. If a partition having data on it is formatted, the addressability of the data contained therein is completely destroyed. Before selecting the format option, be absolutely certain the highlight-bar is positioned on the right partition to be formatted.

This button is active for all primary partitions and all logical drives within an extended partition, whether they are already formatted or not.

After **Format** is selected, another dialog appears: **"Format Partition"**. The format is not initiated until you answer ***Yes*** to the ***Are you sure*** dialog. This confirmation is used to ensure that you:
a) Are sure you want to format this partition, and …
b) have selected the correct partition to be formatted.



Note: The file system to be used in a given partition is limited by the partition size and the type of partition. *(For example, no FAT-16 formatting is possible in a 4 GB partition.)*

3.2.5 Delete Button

Purpose: Completely removes the partition from the partition table, and frees up the disk space which the partition now occupies. ***Destroys all data that was there before.***

Critical: Pressing this button when a partition with valid data is selected, will cause **permanent loss of all data** in the partition.

This button is active for all primary and extended partitions as well as for all logical drives within an extended partition. **The deletion of an extended partition also deletes all logical drives within the partition.**

Note: Deleted partitions are converted to raw disk space (which are denoted as primary or logical <free> by Partition Manager). All data within a deleted partition is lost. Partition manager therefore confirms the partition to be deleted by asking you to supply the volume label of the partition to be deleted. If the volume has not been named, "NO NAME" should be supplied as the volume label.
This confirmation ensures that you:
a) Really want to delete the selected partition.
b) Have selected the correct partition to be deleted.

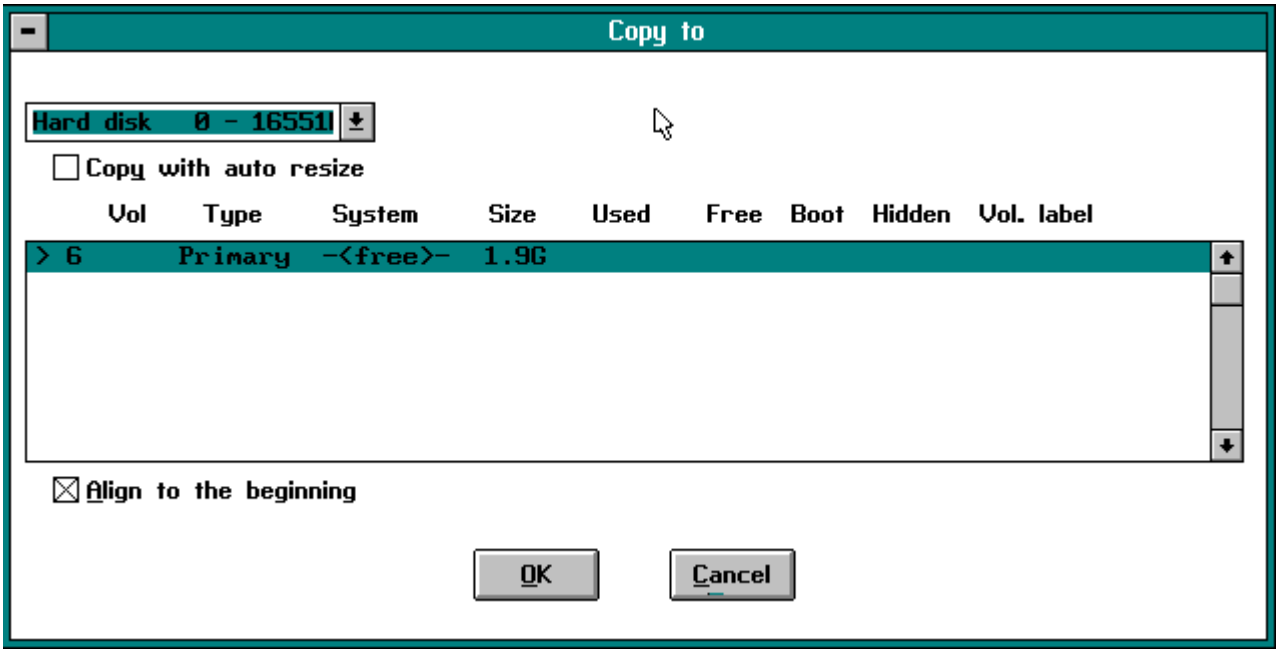
3.2.6 Copy Button

Purpose: Copy (clone) a partition to an another (free) partition.

This function applies to primary and extended partitions and to all logical drives within an extended partition, previously formatted or not.

As target (copy destination), only empty disk space (primary <free> or logical <free>) can be used. Free disk space as target is selectable both on the source drive or on any other available hard drive. The partition to be copied can be aligned to the begin or to the end of the free block.

The Partition can be copied in a 1:1 mode or in Autoresize mode. In the



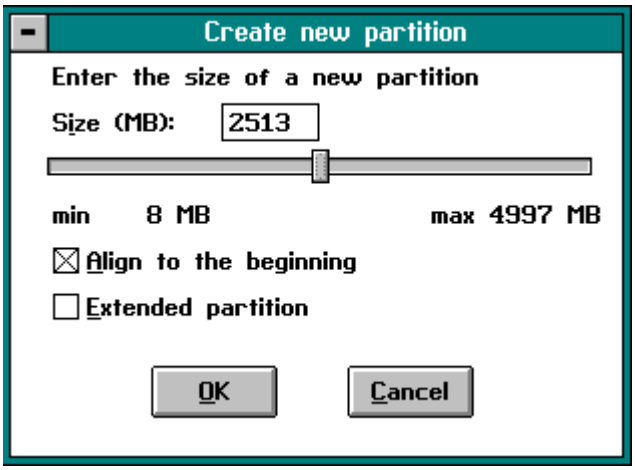
Autoresize mode has the ability to adjust the size of the target partition from a minimum (amount of data in the source partition) to a maximum (total size of the target partition) or anywhere in between. This auto-adjusting size (to the target) feature is possible only in partitions using the following file systems: FAT-16, FAT-32, NTFS, HPFS and EXTFS.

The uniqueness of this auto-size adjustment feature lies in the fact that the target partition can be **smaller** as well as larger than the source partition. The key in the auto-size adjustment feature lies in the size of the **source data** rather than the size of the **source partition**. The target partition must of course be at least as large as the data in the source partition, plus a slight security margin.

3.2.7 Create Button

Purpose: Create a new partition within an unused area of the hard drive.

The object to be acted on here is free space, therefore this button is active only when the highlight-bar is positioned on primary <free> or logical <free> disk space. This is the command that enables the creation of new partitions, just as the copy function creates new partitions with the contents of the old partition.



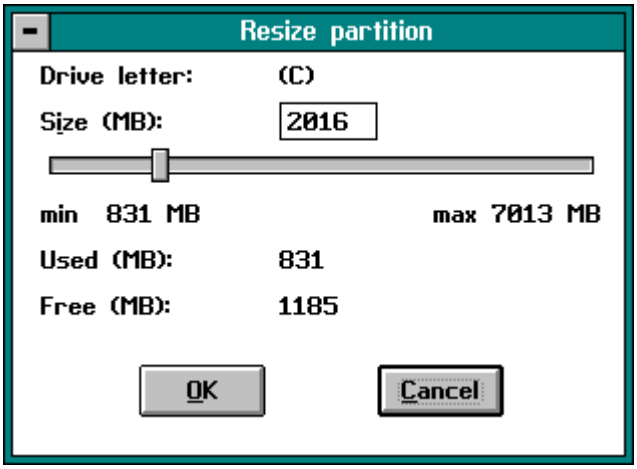
If you do not want to use the full size of the free block, this is how to determine where to create the remaining free disk space: Conventionally, it is created at the end of the new partition. For this, the **Align to beginning** box should be **unchecked**. If instead, you wish to align the partition at the start of the free space, then **check the box.**

3.2.8 Resize Button

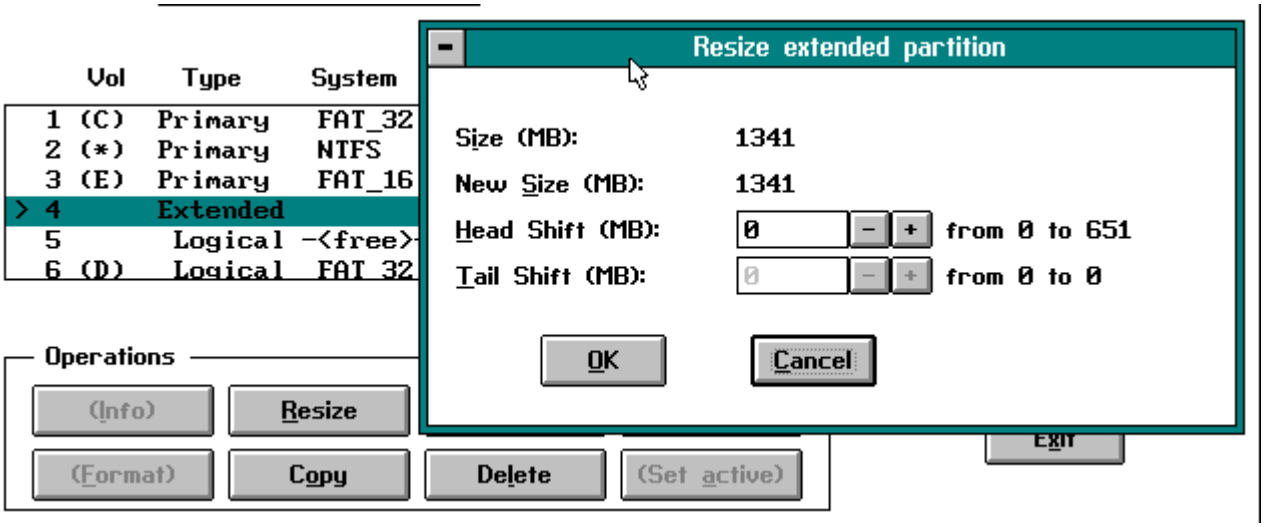
Purpose: Enlarge a partition with free space at either end, or shrink the partition to fit the data in it.

This button is active if the highlight-bar is positioned on an extended partition or one using the following file systems: FAT-16, FAT-32 NTFS EXT2 or HPFS.

In this example a primary 2016 MB partition was selected. Since this partition contains 831 MB data, it is reducable up to 831 MB. The size of this partition can be increased up to 7013 MB because there is a primary free disk space available after it. The resizing takes place after the requested action is confirmed with OK.



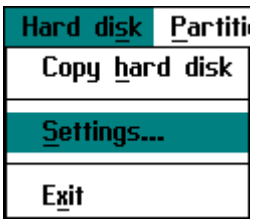
When resizing a *primary* or *logical* partition, we can only move the partition's end (to shrink, inward and to grow, outward). A unique feature of Partition Manager is that it allows you (unlike primary or logical partitions) to redefine an extended partition's beginning **and** end, as illustrated by the screen for resizing an extended partition:



In order to increase the size in any direction, naturally free space has to exist there. An inward resizing can occur provided there is no data in that region of the partition. PM automatically determines the resizing limits in both directions, for both ends of the partition and these are shown after the **Head shift** and **Tail shift** windows. The **from x to n** values displayed after each of these windows represent the possible change for either the head or the tail of the partition. A negative value represents an inward shift of the tail (size reduction), or an outward shift of the head (size increase in that direction) and a positive number an outward shift of the tail (size increase in that direction) or in inward shift of the head (size reduction).

Once the desired increase or decrease for head and/or tail are determined, you can either explicitly enter values in the two windows, or use the up and downarrow keys to increment or decrement the desired change by one single MB.

3.3 The Hard disk Pulldown



3.3.1 Hard disk copy

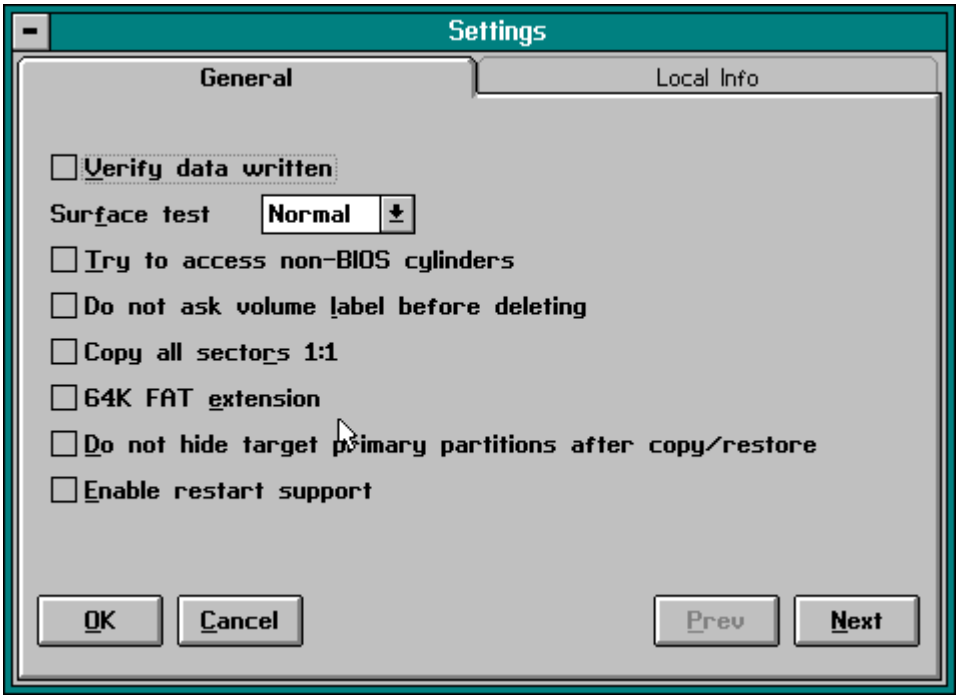
Purpose: Copy one entire hard drive to another, sector-by-sector (which ensures the exact contents will be the same).

This option is active only when at least two hard drives are available in the system. When this is the case, the currently highlighted drive will be used as the source drive and a 1:1 straight copy will be made to the target drive. The source drive is copied 1:1 to the target drive, and the previous contents of the target drive are lost in the process.

3.3.2 Settings

Purpose: Adjust critical system parameters for certain functions.
These options don't apply to any specific hard drive or partition. They are adjustments that regulate the mode of operation of some functions of Partition Manager.

The setting windows consists of 2 tabs (pages)



Verify data written
Regulates the checking (verification) of all writing procedures of Partition Manager. If this parameter is set, a verification (comparison) of data read and data written is done, at the sector level. Although use of this switch slows all operations down considerably, it also ensures error-free data writing operations.

Surface test
By setting of this parameter to one of the various options, the usability (physical state) of the disk space involved in a partition Create, Copy, Move or Format operation will be verified.

Possible settings and their effect:
None: No physical examination of the disk space is performed.
Normal: Data on sectors written is compared to the sectors read. A standard read/write verify of each sector in the partition occurs.
Intensive: Read and written sectors are re-read and compared a second time.

The standard (and recommended) setting is normal. The none setting is the quickest but if defective sectors are encountered, they will not be tagged and can cause data loss later on. The intensive method is recommended only for situations in which multiple defective sectors are anticipated. In this instance, two comparisons will often detect a bad sector where one may not. Needless to say, identifying the bad sectors before any important data is entrusted to them is preferable at this point in time.

Try to access non-BIOS cylinders
This function is relevant only if there are operating systems such as Windows NT, Windows 2000 and some newer Linux versions installed. These OSes bypass the PC BIOS and use their own internal BIOS to access cylinders (invisible to the PC-BIOS). If this parameter is set (switched on), PM will attempt to access these cylinders also.

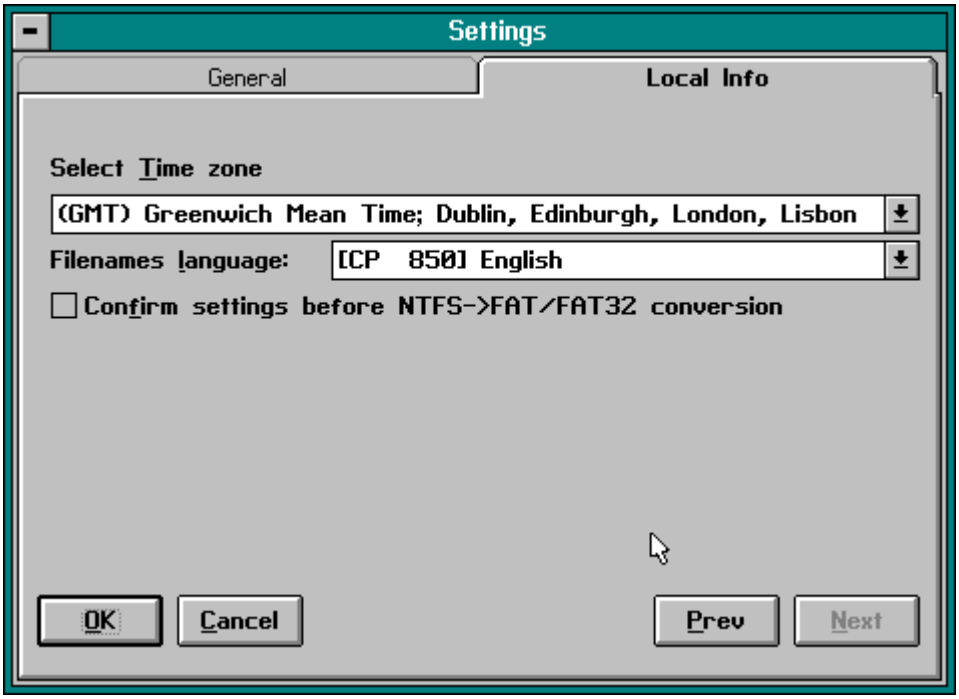
Do not ask volume label before deleting
If selecting a partition for deleting you will be asked for the name of that partition(if the volume has not been named, NO NAME should be supplied as the volume label). If switched on you do not need to enter the name to proceed in deleting a pratution.

Copy all sectors 1:1
If the box is checked, a 1:1 sector copy will be enforced in all copy-operations.

64K FAT extension
This option allows the creation of a 64 KB cluster size within FAT-16 partitions (which enables Windows NT to read this file system, since it can use 64 KB clusters). This also doubles the upper limit from (a standard) 64 sectors (512 bytes) to 128 sectors. Only Windows NT will be able to read and write the resulting clusters. A partition whose file

system has been altered in this way will no longer be recognizable to a standard DOS. Partition Manager detects both cluster sizes and creates (in the standard mode) 32 KB clusters, unless this parameter is set (checked), in which case 64 KB clusters are created.

Do not hide target primary partitions after Copy/Backup/Restore
In the default installation the new copy of a partition (the target partition) is automatically hidden. The reason for this is that the logical drive designator sequences should be kept. When this parameter is enabled (partition visible), the target partition is visible and therefore a change in the drive designator sequence can lead to confusion.



On tab 2 (page 2) the localisation table you may choose the time zone and the language font.

Confirm settings before NTFS → FAT/FAT32 conversion
If selected you will be asked to conform the settings of a file conversion from NTFS to FAT.

3.4 The Partition Pulldown

All functions in this section refer to, and act upon, the currently selected partition (as highlighted by the highlight-bar), or in the case of free space, that free section of the drive.

Some of the functions presented here are identical to their button counterparts. (For example: **Resize** button, **Resize Partiton** function). Other functions have slight variations in the types and options. For example, the **Copy** button, as opposed to **Partition | Copy Partition ...** (autoresize mode). The partition pulldown also has some functions that the buttons don't, such as **Convert.**

3.4.1 Partitions | Copy Partition ...

Purpose: Copy partitions, with automatic adjustment to the target (certain file systems only).

This function is identical to the **Copy** button. **See section 3.2.6. Copy Button.**

3.4.2 Partitions | Move Partition ...

Purpose: Moving a partition to directly adjacent free space before or after the partition (adjacent primary or logical <free>), regardless of how big the adjacent free space is

The partition to be moved must first be selected using the highlight-bar. This option is only selectable if the source partition is a formatted logical drive (either primary or *logical within extended*, partition). An extended partition and all the logical drives it contains can be moved all together as a unit.

The selected partition can be moved (on the hard drive) parallel in both directions. The moving can take place only if there is a primary or logical <free> space either directly before, or directly following the selected source (to be copied) partition.

The moved partition will be aligned automaticly to the begin of the free block, if the free block is before the partition or to the end of the free block if the free block is behind the partition. This creates in both cases the biggest possible free block where the moved partition was located.

3.4.3 Partitions | Accessibility

Purpose: View/change which operating systems have access to the selected partition.
This option regulates the access of Windows NT and OS2 on the one hand and WIN-95/-98 on the other hand, to FAT-16 partitions with differing FAT-IDs in the partition table. In this regard, one must understand that FAT-16 partitions can have different IDs, as well as from which system and in which BIOS addressing mode they were created.

For example, Windows 95 release B and Windows 98 assign a signature (ID) of 0xE to a partition at the end of a hard drive. However, this signature won't be detected as FAT-16 by Windows NT or OS2 and a partition having this ID will therefore be unaddressable by these two OS's.

Only when the partition signature is modified from 0xE to 0x6, are Windows NT and OS2 able to recognize this FAT-16 partition. Only now the partition won't be recognizable by Windows 95 release B or Windows 98.

The reason for this lies in the fact that Windows 95/98 uses the BIOS extension (termed EBIOS) to address such partitions in the normal mode, since such partitions (at the end of the hard drive) frequently cross the 1024 cylinder boundary in the normal mode, and thus blow up the normal BIOS addressing.

The special signature tells Windows 95/98 to use the EBIOS instead of the BIOS. On the other hand, Win NT and OS2 (in order to access these partitions), use no BIOS at all but instead address the partitions directly, using the hard drive controller. Therefore the special FAT-16 signatures are totally foreign to these OS's.

When such partitions are created by Partition Manager they receive the Windows 95/98 addressing signature. If a partition is to be used under Windows NT or OS2, the change must be specified here to make the partition recognizable to these operating systems.

When such a change must be made, the modified partition status can be verified using the **Info** button. The resulting addressability statuses (seen in plain text), can be summarized as:

- All (Partition is recognizable to all known operating systems.)
- Windows NT, OS2
- Windows 95/98

3.4.4 Partitions | Convert

Purpose: Convert FAT file systems from one to the other (FAT-16 to FAT-32 and vice versa) and FAT to NTFS and vice versa.

The partition to be converted must first be selected. This is always the partition illuminated by the highlight-bar. Next, select **Partitions | Convert**. To the right will appear the possible destination format.

A conversion is relevant only if the installed OS systems supports the new target file system. While all OS understand FAT-16, the FAT-32 systems is supported by Windows 95 (release B), Windows 98 ME, WIN 2000 and Windows XP. The NTFS file system is supported by WIN NT4, WIN 2000 and Windows XP only.

This function is valid only for logical drives with the FAT-16 or FAT-32 or NTFS file system. The conversion offers for a selected FAT-16 drive FAT-32 and NTFS, for a selected FAT-32 drive NTFS and FAT-16(*) and for a selected NTFS drive FAT-32 and FAT 16(*). (*) FAT-16 is offered only if the source drive is below 2 GB.

The following restrictions apply:

- The minimum size for a FAT-16 partition to be converted to FAT-32 is 300 MB.
- The maximum size from FAT-32 to FAT-16 is 2.0 GB.

After the new (to be converted to) filesystem is specified, a confirmation dialog appears asking: "Convert partition. Are you sure ?". Only with an affirmative response does the conversion commence.

Critical:
Converting a DOS partition to FAT-32 renders it unusable to DOS thereafter.

3.4.5 Partitions | Resize Partition ...

Purpose: Enlarge a partition with free space at either end, or shrink the partition to fit the data in it.

This function is identical to the **Resize** button. **See section 3.2.8, Resize Button.**

3.4.6 Partitions | Create Partition ...

Purpose: Create a new partition within an unused area of the hard drive.

This function is identical to the Create button. See section 3.2.7, Create Button.

3.4.7 Partitions | Format Partition ...

Purpose: Prepare a partition with a specific file system. ***Destroys all previous data.***

This function is identical to the **Format** button. **See section 3.2.4, Format Button.**

3.4.8 Partitions | Delete

Purpose: Completely removes the partition from the partition table, and frees up the disk space which the partition now occupies. ***Destroys all data that was there before.***

This function is identical to the **Delete** button. **See section 3.2.5, Delete Button.**

3.4.9 Partitions | Hide

Purpose: Make partitions invisible (or recognizable) to an operating system.

This function is identical to the **Hide/Unhide** button. **See section 3.2.2, Hide/Unhide Button.**

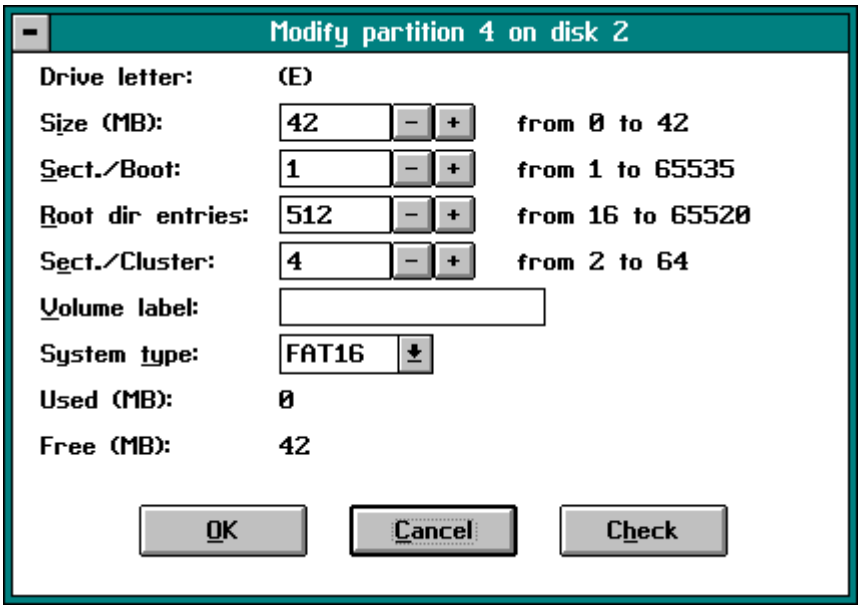
3.4.10 Partitions | Set inactive

Purpose: Tells the partition table this partition is to be recognized as bootable or not bootable.

This function is identical to the **Set Active/Set Inactive** button.
See section 3.2.3, Set Active/Set Inactive Button.

3.4.11 Partitions | Modify entry

These parameters are reserved for the use of experts, since expert knowledge of the file system is required, in order to accurately predict the effects of manually made modifications and variations available here. In theory everything can be adjusted here, including enabling FAT-16 to FAT-32 conversion (and vice-versa).



These options should really only be used when one knows the exact consequences the newly selected parameters will have upon the partition's file system.

This window also appears (in most cases) using the **Format** function. Whether some or all of the parameters will be changeable depends on where this window was called from, and also on the status of the selected partition (free, primary, formatted, etc).

Size (MB)
This input is changeable only when a formatted partition is selected. To the right of the data, the potential input range for the size is displayed.
Minimum size: This is the size occupied by existing data in the partition.
Maximum size: Either the current size of the partition, or if there's free space directly following the end of the partition, the current size of the partition plus the size of this free space.

Sect./Boot
This is the number of sectors that are required for the boot area, normally only one single sector. Apart from your own objectives (for your specific system programming), there are certain instances in which more than one boot sector are used. (Some embedded systems have several "reserved" boot sectors.)

Root Entries
This is the number of sectors reserved for the root directory. Apart from your own objectives (for example simply having a large number of free sectors at the beginning of a partition) this is important for FAT-16 partitions if you plan to have either many files or a lot of long file names in the root directory.

Sect./Cluster
Depending on the file system used, and the selected size of the partition, the cluster size can be manually defined here. For a FAT-16 system, if the partition is small enough, even a very small cluster size of 1:1 is possible (1 cluster = 1 sector = 512 bytes) For more information on the FAT system, please see the section on "**Fundamentals**".

Volume Label
This is the name identifier for the partition in question. **This entry is case sensitive**. This label is used by any and all critical operations that deal with the partition. Whatever you enter here will be used to identify this volume, at the base partition level as well as in your operating system. In Windows 95/98, for example, if you right-click the volume (logical drive) and do **properties** the volume name will be displayed, along with other data, such as the size, whether it's a local hard drive, etc.

This pertains to a confirmation prompt used by partition programs that are trying to confirm critical operations upon a given partition. If, for example, you wish to delete a specific partition, partition managers (in general, and also Paragon Partition Manager), will ask you for the *volume name*, prior to deleting the partition in question. This is a security issue. *NOTE: The partition programs always ask you for the volume name, but If there is no name assigned to a given volume, and an operation such as delete or other potentially destructive function is selected, then you must enter **NO NAME** to proceed with the delete or other operation.*

System Type
Defines the file-system type of the selected partition. Be extremely careful when using this choice for a formatted partition. Instead, it is preferred to use the **Format** button.

For more information see: **Section 3.2.4 Format Button.**

Used (MB)
For informational purposes only. Displays the total amount of data in MB which resides in this partition. This is also the minimum size to which this partition can be resized.

Free (MB)
For informational purposes only. The amount of free (unused) space in this partition

3.4.12 Partitions | Info

This function is identical to the **Info** button. See: **Section 3.2.1, Info Button.**

3.5 Help

This pulldown menu is self-explanatory. It contains the submenus (options list), the (general index) and info. Info includes, among other things, the version information and support addresses.

4. Operations – Step-by-step instructions for Windows and DOS functions

This chapter supplements the manual with step-by-step instructions of how to execute the functions of Partition Manager. It is also meant to serve as a reference and as such refers to the functions and options in Chapter 2 (Windows) and Chapter 3 (DOS), as well as the examples in Chapter 5.

Important Note for all Windows versions:
You can never do a resize, copy, converting or move of the boot partition directly under Windows. On Windows 9x the program will ask for a reboot in DOS, in WIN 2000, NT and XP the task will be done in a boot time service (blue screen mode) and in Windows ME you can do this operations only by booting from the PM Diskette (DOS) version. Please check chapter 1.5.2 Windows – Operation on Boot partition.

4.1 Create new partition (or logical drive)

This refers to the following functions/sections of the manual:
Windows: Section 2.3.6 - CREATE PARTITION (**Partition** pulldown, **Create ...**)
DOS: Section 3.2.7 - ***<Create>*** button.
Section 3.4.6 - **Partitions** pulldown, **Create Partition** (identical to button)

Note:
In order to create a new partition, the current selection must be primary or logical free space. (In the DOS version, the highlight-bar must be on free space, in Windows free space must be clicked-on with the mouse.) If this is not the case, then the function is in both versions disabled (inactive, greyed-out).

4.1.1 Creating primary or extended partitions.

Examples in chapter 5.1: Creating 3 partitions (1 Primary, 2 logical drives)
Procedure:
1. Select primary or logical free space, using the highlight-bar (DOS) or the mouse (Windows).
2. In DOS or Windows: Pull down **Partition**, select **Create** or **Create Partition**. In DOS, you can click on the ***<Create>*** button. In Windows, you can press the **Create** (5th) button on the toolbar. In Windows, right-clicking primary or logical free space produces a pop-up with the create option.
3. The **Create partition** dialog appears.
4. Enter the desired size of the new partition, either explicitly or using the slider bar.
5. Check **Create an extended partition**, if creation of an extended partition is required (Win).
6. If not using the entire free space, set **Align to the beginning of the block** as needed (see **Notes**, below)
7. Click **OK** to create the new partition. No further confirmation is required.

Notes:
The maximum size of the new partition will depend on the size of the selected primary free space. The **Align to the beginning of the block** option is only relevant when it is not desired to use the entire selected free space. If you don't want to use the entire free space to create the new partition, checking this box creates the partition at the very beginning of the free space. **If the box is unchecked**, the partition will be placed *at the **end** of the free space.*

The **Create an extended partition** option appears only if there isn't already a partition of this type, and if a primary free space is currently selected. To create a primary, uncheck the box. If the box is checked, an extended partition will be created.

4.1.2 Creating logical drives

Examples in chapter 5.1: Creating 3 partitions (1 Primary, 2 logical drives)
A logical drive can be created only within an extended partition. The free disk space contained in this extended partition will be identified as *logical free.*

Procedure:
1. Select logical free space in an extended partition, using the highlight-bar (DOS) or the mouse (Windows).
2. In DOS or Windows: pull down **Partition**, select **Create** or **Create Partition**. In DOS, you can click on the ***<Create>*** button. In Windows, you can press the **Create** (5th) button on the toolbar. In Windows, right-clicking a logical free space also allows you to create a logical drive.

3. Enter the desired size of the new logical drive, either explicitly or using the slider bar.
4. If not using the entire free space, set **Align to the beginning of the block** as needed (see **Notes**, below)
5. Click **OK** to create the new logical. The new logical drive is created immediately.

Notes:
The maximum size of the new partition will depend on the size of the selected logical free space. The **Align to the beginning of the block** option is only relevant when it is not desired to use the entire selected free space. If you don't want to use the entire free space to create the new logical drive, checking this box creates the logical drive at the very beginning of the free space. **If the box is unchecked**, the logical drive will be placed *at the **end** of the free space.*

4.2 Changing the size of a partition (resize)

This refers to the following functions/sections of the manual:
Windows: Section 2.3.5 - RESIZE PARTITION (**Partition** pulldown, **Resize ...**)
DOS: Section 3.2.8 - ***<Resize>*** button.
Section 3.4.5 - **Partitions** pulldown, **Resize Partition** (identical to button)

Note:
This function is relevant only for formatted partitions, therefore a formatted partition (either primary or logical) would have to be selected either with a mouse-click (Windows), or with the highlight-bar (DOS). If anything else is selected, these functions become inactive. Further, the selected partition has to have one of the following file systems, in order for these functions to be active: FAT-16, FAT-32, NTFS, EXT2FS or HPFS. Any other partition types or unformatted areas as selections disable the resize function.

Enlarging a partition is possible only when primary free space exists at the end of the partition. The exception here is, in the case of an extended partition, which can be enlarged when the free disk space is at the beginning of the partition (or at the end).

Shrinking a partition is possible only when the entire partition isn't filled with data.

4.2.1 Shrinking and/or enlarging a primary partition or logical drive

Examples in sections 5.4 through 5.5.4 Shrinking and/or enlarging partitions.

When shrinking a primary partition the partition end is shifted inward (offset forward). When enlarging, the partition end is shifted outward.

Procedure:
1. Select a primary formatted partition or a logical formatted drive, using the highlight-bar (DOS) or the mouse (Windows).
2. In DOS or Windows: pull down **Partition**, select **Resize** or **Resize Partition**. In DOS, you can click on the ***<Resize>*** button. In Windows, you can press the **Resize partition** (4th) button on the toolbar. In Windows, right-clicking a valid existing formatted partition, will give a resize option.
3. The **Resize partition** dialog appears.
4. Enter the desired size (within the indicated boundaries).
5. Click **OK** to resize the primary partition or logical drive. The task is immediately executed.

Notes:
The indicated maximum size displayed in the **Resize partition** dialog consists of the current size of the partition, plus any primary free space following it. If none is available, then the maximum size is the current size of the partition, and no further enlargement is possible. The indicated minimum is determined by the dataset contained within the partition. If the partition is full of data, then it cannot be shrunken and the minimum size is identical with the current size.

4.2.2 Changing the size of an extended partition.

This refers to the following functions/sections of the manual:
Examples in:

Section 5.5.2 - **Enlarging a primary partition at the expense of a logical drive.**
Section 5.5.4 - **Enlarging a logical drive at the expense of a primary partition.**
When changing the size of extended partitions, (unlike primary partitions and logical drives), both the beginning (head) and ending (tail) of the partition can be offset toward the front or rear (in or out).

Procedure:

1. Select an extended partition, using the highlight-bar (DOS) or the mouse (Windows).
2. In DOS or Windows: pull down **Partition**, select **Resize** or **Resize Partition**. In DOS, you can click on the ***<Resize>*** button. In Windows, you can press the **Resize partition** (4th) button on the toolbar. In Windows, right-clicking a valid existing formatted extended partition, will allow you to resize it as well.
3. The **Resize extended partition** dialog appears.
4. Enter the desired offset/shift of either head, tail or both (within the indicated boundaries), either explicitly or using the slider bars. PM automatically keeps your input within the required boundaries. See explanation below for more detail.
5. Click **OK** to proceed to resize the extended partition.
6. Click **Yes** at the **Are you sure ?** confirmation dialog (Windows).
7. The progress report dialog lets you know how far along the resize is (Windows).

For an extended partition, the size can be dynamically adjusted at either end, in both directions. In order to increase the size in any direction, naturally free space has to exist there. An inward resizing can occur provided there is no logical drive, but a logical free in that region of the partition. PM automatically determines the resizing limits in both directions, for both ends of the partition and these are shown after the **Head shift** and **Tail shift** windows. The **from x to n** values displayed after each of these windows represent the possible change for either the head or the tail of the partition. A negative value represents an inward shift of the tail (size reduction), or an outward shift of the head (size increase in that direction) and a positive number an outward shift of the tail (size increase in that direction) or an inward shift of the head (size reduction).

Once the desired increase or decrease for head and/or tail are determined, you can either explicitly enter values in the two windows, or use the up and downarrow keys to increment or decrement the desired change by one single MB.

4.2.3 Enlarging a partition with no free space at either end.

This refers to the following functions/sections of the manual:
Examples in:
Section 5.5 - **Enlarging partitions at the expense of other partitions.**
Section 5.5.1 - **Enlarging a primary partition at the expense of another primary partition.**
Section 5.5.2 - **Enlarging a primary partition at the expense of a logical drive.**
Section 5.5.3 - **Enlarging a logical drive at the expense of another logical drive and**
Section 5.5.4 - **Enlarging a logical drive at the expense of a primary partition.**

Regardless of what type of partition we're dealing with (primary, extended or logical drive), partitions can be enlarged only when there's free disk space outside of its borders (for primary and extended partitions, primary free space; for logical drives logical free space).

In the case of primary partitions and logical drives, there is the further limitation that the free disk space must be at the end of the partition or logical drive. In the case of extended partitions, free space at the beginning and end of the partition can be utilized to enlarge the partition.

If no such free space is available, then the first step is to create such a free space at the expence of an another existing partition or logical drive; Examples in sections 5.5.1 through 5.5.4

4.2.4 Changing the size of a logical drive

Examples in:
Section 5.5.3 - **Enlarging a logical drive at the expense of another logical drive.**
Section 5.5.4 - **Enlarging a logical drive at the expense of a primary partition.**

Changing the size of a logical drive is similar to changing the size of a primary partition. In fact, the prerequisite conditions are pretty much identical. For enlargement, there has to be ***logical free*** space after the end of the logical

drive (yet within the extended partition containing the logical drive), and for shrinking there has to be less data than the current size (of the logical drive).

4.3 Formatting partitions

This refers to the following functions/sections of the manual:
Windows: Section 2.3.7 - FORMAT PARTITION (**Partition** pulldown, **Format ...**)
DOS: Section 3.2.4 - ***<Format>*** button.
Section 3.4.7 - **Partitions** pulldown, **Format Partition** (identical to button)

Critical:
It's possible to format both unformatted partitions, as well as already formatted partitions containing valid data. If a partition having data on it is formatted, the addressability of the data contained therein is **completely destroyed**. ***Before selecting the format option, be absolutely certain you've selected the correct partition to be formatted.***

Only genuine partitions (format-ready or unformatted, yet created) are formattable. That is, only when an existing primary or logical partition (not free space) is selected with either the highlight-bar (DOS) or the mouse (Windows), is the format function active. In any other instance this function will not be available.

Procedure:
1. Select a valid existing partition or logical drive (not free space), using the highlight-bar (DOS) or the mouse (Windows).
2. In DOS or Windows: pull down **Partition**, select **Format** or **Format Partition**. In DOS, you can click on the ***<Format>*** button. In Windows, you can press the **Format** (6th) button on the toolbar. In Windows, right-clicking any valid existing partition also gives a **Format** option.
3. The **Format partition** dialog appears.
4. Select the desired file **System type** (See the table in section 1.3.7 for a summary).
5. Optional: Specify a **Volume Name**. This can be done at a later time as well.
6. Optional: Check the **Surface test** box to check the hard drive surface (section 2.2.2 for details).
7. Click the **Advanced** button for expert options, if needed (section 2.3.7 for details).
8. Click **OK** when finished entering parameters. ***All data will be lost.***

Notes:
If a **Surface test** is desired, the intensity can be set under **Hard disk, Settings**.

Depending on the partition size and other parameters, not all file systems are available for all partitions. In order to better understand what other parameters may need to be modified depending on the situation, see Section **2.3.3, Partition Accessibility** and other related sections. Of particular interest are the extended parameters in the **Format** section (2.3.7) as well as (2.2.2) the **Settings** pulldown in the **Hard Disk** menu (Windows).

4.4 Converting partitions

This refers to the following functions/sections of the manual:
Windows: Section 2.3.4 - CONVERT FILE SYSTEM (**Partition** pulldown, **Convert ...**)
DOS: Section 3.4.4 - **Partitions** pulldown, **Convert**.

This function applies only to partitions with the FAT-16, FAT-32 and NTFS file systems.

Procedure:
1. Select a partition or logical drive with the FAT-16 or FAT-32 or NTFS file system, using the highlight-bar (DOS) or the mouse (Windows).
2. In DOS or Windows: pull down **Partition**, select **Convert**.
3. The possible target formats are displayed in the **System type** window.
4. Select a target file system
5. Click **OK** to convert the partition.
6. Click **Yes** at the **Are you sure ?** warning dialog.
7. PM converts and re-checks the partition integrity and shows the progress (Windows).

4.5 Copying partitions

This refers to the following functions/sections of the manual:
Windows: Section 2.3.1 - COPY PARTITION (**Partition** pulldown, **Copy partition ...**)
DOS: Section 3.2.6 - ***<Copy>*** button.
Section 3.4.1 - **Partitions** pulldown, **Copy partition**.

It's only possible to copy actual existing primary or logical partitions, to either primary free or logical free space. If it's desired to copy the partition into another actual existing partition (the "target" partition), then that target partition must first be deleted.

Exercise 1: Make a copy of a primary partition, to one of equal size. Please refer to the following section 4.5.1, Creating a 1:1 copy of a partition.

Exercise 2: Make a backup copy of the same partition (as in Exercise 1), in which both the type of partition and the size of the partition may change. All the data in this partition is actually to be transferred to the new one. For this, please refer to the following section, 4.5.2 Copying with adjustment to the target partition.

4.5.1 Creating a 1:1 copy of a partition

This refers to the following functions/sections of the manual:
Windows: Section 2.3.1 - COPY PARTITION (**Partition** pulldown, **Copy partition ...**)
DOS: Section 3.2.6 - ***<Copy>*** button
Section 3.4.1 - **Partitions** pulldown, **Copy partition**

Example in section 5.2.1 - **Copying in the 1:1 mode (sector-by-sector copy).**

All real partitions are copyable, including extended partitions. What isn't copyable is free disk space (either type primary or logical free). Furthermore, this mode of copying is usable on all types of partitions, including unknown file systems (A 1:1 copy is a sector-by-sector image copy; the contents of the sectors is ignored).

Further conditions: A partition of type primary or logical free of equal size or larger as the source partition (or logical drive) must be available as target.

Procedure:
1. Select a valid existing partition or logical drive (not free space), using the highlight-bar (DOS) or the mouse (Windows).
2. In DOS or Windows: pull down **Partition**, select **Copy Partition**. In DOS, you can click on the ***<Copy>*** button (step 1 being done) in Windows, you can press the **Copy Partition** (2nd) button on the toolbar. In Windows, right-clicking any valid partition will also offer a copy option, provided there is a target of sufficient size available.
3. The **Copy partition** dialog appears.
4. Select the desired hard drive using the pulldown, then click (or highlight) the target partition.
5. Turn the **Copy with autoresize** option off (no check).
6. If not using the entire free space, set **Align to the beginning of the block** as needed (see **Notes**, below)
7. Click **OK** when all data has been entered into the **Copy partition** dialog.
8. PM may tell you that the new partition will be hidden. Click **Yes** in any case to proceed.

Notes:
The **Align to the beginning of the block** option is only relevant when it is not desired to use the entire selected free space. If you don't want to use the entire free space to create the new partition, checking this box creates the partition at the very beginning of the free space. **If the box is unchecked**, the partition will be placed *at the **end** of the free space.*

If Partition Manager finds no target partitions of type free of the required size or the required type, then no partitions (steps 4,5) are offered and no 1:1 copy is possible. In Windows, the message **No free entries of requested size on this drive** may appear.

If for some reason the copy function isn't active/available, then the currently selected (source/to be copied) block is most likely free space (of type primary or logical free), rather than an existing partition.

4.5.2 Copying with adjustment to the target partition (autoresize)

This refers to the following functions/sections of the manual:

Windows: Section 2.3.1 - COPY PARTITION (**Partition** pulldown, **Copy partition ...**)
DOS: Section 3.2.6 - ***<Copy>*** button
Section 3.4.1 - **Partitions** pulldown, **Copy partition**

Note: To use the autoresize option, in the DOS mode, the copy function must be invoked from the **Partitions** pulldown, the ***<Copy>*** button works only in 1:1 mode.

Example in section 5.2.2 **Copying in autoresize mode**

This copy option (also known as copying in the autoresize mode), offers you much more flexibility in that you can both copy partitions into a smaller free space as well as also being able to copy primary partitions into a free space of type logical free within an extended partition. On the other hand, this option is not available for all types of file systems. For example, NTFS partitions cannot be copied using this copy option.

Further conditions for this copy option: The target partition has to be at least as big as the data contained in the source partition.

Procedure:

1. Select a valid existing partition or logical drive (not free space), using the highlight-bar (DOS) or the mouse (Windows).
2. In DOS or Windows: pull down **Partition**, select **Copy Partition**. In DOS, you can click on the ***<Copy>*** button (step 1 being done), in Windows, you can press the **Copy Partition** button (2nd button) on the toolbar. In Windows, right-clicking any valid partition will also offer a copy option in a pop-up window, if a target of sufficient size exists.
3. The **Copy partition** dialog appears showing the available target partitions **on the selected drive**.
4. Select the target drive using the pulldown, then the desired target partition.
5. The **Copy with autoresize** option must be turned on (checked) for the autoresize. *PM may at this point give you more target choices, depending on the amount of **data** to be copied in the originally selected source partition.*
6. If not using the entire free space, set **Align to the beginning of the block** as needed (see **Notes**, below)
7. Click **OK** when all data has been entered into the **Copy partition** dialog.
8. PM will now put up a dialog asking you for the desired size of the target partition. Enter the size or use the slider bar. Click **OK**.
9. Click **Yes** at the next dialog to copy the partition using autoresize. PM may warn you that the new partition will be hidden.

Notes:

The **Align to the beginning of the block** option is only relevant when it is not desired to use the entire selected free space. If you don't want to use the entire free space to create the new partition, checking this box creates the partition at the very beginning of the free space. **If the box is unchecked**, the partition will be placed *at the **end** of the free space.*

If the source is an extended partition, and the target is of a different size, all logical drives within the extended partition will be proportionately resized to fit the target.

If Partition Manager finds no target partitions of type free of the required size or the required type, then no partitions (steps 3,4) are offered and no autoresize copy is possible ***on that drive***. In Windows, the message **No free entries of requested size on this drive** may appear. You may have to select a different drive (step 4).

If for some reason the copy menu option isn't active, then the currently selected (source/to be copied) partition is most likely an un-copyable partition (in the autoresize mode), such as an NTFS partition. In this case, only the 1:1 copy option can be executed with this source partition.

4.6 Moving partitions

This refers to the following functions/sections of the manual:

Windows: Section 2.3.2 - MOVE PARTITION (**Partition** pulldown, **Move partition ...**)

DOS: Section 3.4.2 - (**Partitions** pulldown, **Move partition ...**)

Example in section **5.3 Moving a partition**

Moving a partition is appropriate whenever there is a genuine partition with data to be moved parallel into free disk space either before or after it, in order to make more space at the opposite end of the partition. In doing so, the contents of the partition are moved the total of the adjacent free space, which subsequently may completely disappear.

Further conditions necessary for such a move: There must be some directly adjacent free space (either primary or logical free space), either directly before or directly after the selected source partition. Providing such free space is available, this free space is then offered as a target for the move. Moving is possible both forward and backward within adjacent free space.

Procedure:

1. Select a valid existing partition or logical drive (not free space), using the highlight-bar (DOS) or the mouse (Windows).
2. In DOS or Windows: pull down **Partition**, select **Move Partition**. In Windows, you can press the **Move Partition** button (3rd button) on the toolbar. In Windows, a right-click on any valid partition with adjacent free space will also allow a move.
3. The **Move partition** dialog will appear. The dialog box shows the possible target areas.
4. Select the desired target area (free space) from the list offered.
5. Click **OK** when all data is entered at the **Move partition** dialog.
6. The source is immediately moved to the target as shown by the progress indicator.

Notes:
If Partition Manager cannot find any directly adjacent free space, there will be no target areas offered and no move therefore possible. In Windows, the message **No free entries of requested size on this drive** may appear.

If the the move functions are not active, then the currently selected partition is not movable.

4.7 Additional functions and settings

4.7.1 Deleting partitions

This refers to the following functions/sections of the manual:

Windows: Section 2.3.8 - DELETE PARTITION (**Partition** pulldown, **Delete**)
DOS: Section 3.2.5 **- *<Delete>*** button
Section 3.4.8 - (**Partitions** pulldown, **Delete**)

Critical: Using one of these functions when a partition with valid data is selected, will cause **permanent loss of all data** in the selected partition. Further, if it's an extended partition, ***all logical drives it contains will also be permanently lost.***

These functions are active for all primary and extended partitions as well as for all logical drives within an extended partition. The deletion of an extended partition also deletes all logical drives within that extended partition. Deleted partitions are set to free space (primary or logical free). If more than one adjacent partition is deleted, the entire contiguous area becomes one primary or logical free space. Only actual existing partitions can be deleted. Delete does not apply to areas that are already free space.

Procedure:

1. Select a valid existing partition or logical drive (not free space), using the highlight-bar (DOS) or the mouse (Windows).
2. In DOS or Windows: pull down **Partition**, select **Delete**. In DOS, the ***<Delete>*** button can alternately be used. In Windows, you can press the **Delete Partition** (7th) button on the toolbar. In Windows, a right-click on any valid existing partition will also allow a Delete.
3. The **Delete partition** dialog will appear, asking for confirmation of the volume name to be deleted.
4. Enter the exact name of the volume to be deleted as indicated by the **Delete partition** dialog. *This entry is case-sensitive and so must be typed exactly as shown.* Click **OK**.

5. Click **Yes** at the **Are you sure ?** confirmation dialog to delete the partition to free space. ***WARNING: All data will be lost. If extended partition, all logical drives in it, too.***
6. The deleted area will be integrated into any adjacent free space and become a single free block.

Notes:
Deleted partitions are converted to raw disk space (which are denoted as primary or logical free by Partition Manager).

Since all data within a deleted partition is lost, Partition Manager confirms the partition to be deleted by asking you to supply the volume label of the desired partition to be deleted. The volume name is case-sensitive and the exact name of the volume must be supplied in order for the partition to be deleted. If the volume has not been named, the entry **NO NAME** should be supplied (as indicated by PM) as the volume label.

If deletion of an extended partition is attempted, PM advises you that the extended partition may contain other logical drives. If you still choose to proceed with deletion of the extended partition, ***the extended partition will be deleted along with any and all logical drives it may contain****, without any further confirmation.*

If one of the delete functions is not active, then the currently selected partition is not deletable.

4.7.2 Set partition active or inactive

Windows:	Section 2.3.11 - SET PARTITION ACTIVE (**Partition** pulldown, **Set active**)
	Section 2.3.12 - SET PARTITION INACTIVE (**Partition** pulldown, **Set inactive**)
DOS:	Section 3.2.3 **- *<Set Active>/<Set Inactive>*** button (offers opposite of current state)
	Section 3.4.10 - **Partitions** pulldown, **Set inactive** (toggles)

Note: The term active as used here should not be confused with active as it refers to whether a function is available.

Relevant to primary partitions only. This is a toggle switch that allows you to change the current status of the selected partition. If the partition is currently active it can be set inactive and vice-versa. The active status is a flag in the partition table that tells the MBR code this primary partition is to be considered bootable. Installation programs for an OS automatically recognize such a partition on a given drive as the one to install their OS on.

Conditions: Only primary partitions can be set active or inactive.

Procedure:
1. Select a valid *primary* partition, using the highlight-bar (DOS) or the mouse (Windows).
2. In DOS or Windows: pull down **Partition**, select **Set active** or **Set inactive** (In DOS **Set inactive** does both by toggling). In DOS, the ***<Set Active/Set Inactive>*** button can alternately be used. This button will display one or the other, depending on the current state of the partition. In Windows, a right-click on any valid primary partition will also allow the "active" status to be changed using **Set active** or **Set inactive**.
3. The active status of the primary partition is immediately changed to the opposite state.
4. The updated status of the primary partition will be shown as **YES** or **NO** in the **Active** column (Windows) or the **Bootable** column (DOS) of the detail line for that partition.

Notes: If the set active/set inactive functions are disabled, the selected partition is not a valid primary partition. Return to step one.

4.7.3 Hiding/Unhiding partitions

Windows:	Section 2.3.9 -	HIDE PARTITION (**Partition** pulldown, **Hide**)
	Section 2.3.10 -	UNHIDE PARTITION (**Partition** pulldown, **Unhide**)
DOS:	Section 3.2.2 **-**	***<Hide>/<Unhide>*** button.
	Section 3.4.9 -	**Partitions** pulldown, **Hide**

Hidden partitions are not visible to booted operating systems and are thus not assigned a logical drive assignment. Therefore, by hiding partitions, you can not only make certain partitions invisible, but also force a change in the entire drive designator sequence.

Conditions: If an extended partition is selected then all logical drives within the extended partition will be set to hidden.

Procedure:

1. Select a valid partition (not free space), using the highlight-bar (DOS) or the mouse (Windows).
2. In DOS or Windows: pull down **Partition**, select **Hide** or **Unhide** (In DOS **Hide** does both by toggling). In DOS, the ***<Hide/Unhide>*** button can alternately be used. This button will display one or the other, depending on the current state of the partition. In Windows, a right-click on any valid partition will also allow the "hidden" status to be changed using **Hide** or **Unhide**.
3. If the partition is an active primary, PM warns you that the drive will no longer be bootable. If you still wish to continue, click **Yes**.
4. The hidden status of the partition is immediately changed to the opposite state.
5. The updated status of the primary partition will be shown as **YES** or **NO** in the **Hidden** column (Windows) of the detail line for that partition. In the DOS version this shows as ***<hid>*** in the **Hidden** column (or nothing if not hidden) of the detail line for that partition.

Notes:

In the event the hide/unhide functions are disabled (inactive), then the selected area is not a valid partition, but rather free primary or free logical disk space.

5. Examples of working with Partition Manager (scenarios)

Here we present some basic exercises (Examples 5.1 thru 5.3), as well as some examples that are somewhat more difficult (Examples 5.5.1 thru 5.5.4) The steps necessary to execute each exercise are presented one by one.

5.1 Creating three partitions (One primary partition, two logical drives)

Configuration requirements:
Empty hard drive or in any case enough free unoccupied disk space available and a minimum of 2 free entries in the partition table. Furthermore, no extended partitions should exist.

Summary: This section describes how to create each possible kind of partition. We will create a primary (bootable) partition, an extended partition, and two logical drives within that extended partition.

Step one: Create a primary partition

Notes:
In order to create a primary partition, the **Extended Partition** box should be unchecked. The example we are discussing here shows how to create a **primary** partition, which is the main, bootable partition on that drive. Extended partitions are for additional logical drives only.

Also, for the installation of Windows 95 or 98, a minimum of 150 MB disk space has to be available in the partition in which either of these operating systems is desired to be created.

1.1 Select an unused block of disk space, type ***primary free*** (in Windows, right click the free block in any of the windows. In DOS, select the block using the highlight bar). In DOS, press the ***<Create>*** button or pull down the **Partition** menu, then select **Create**. In Windows, the latter option also applies, but you can also just press the **Create** button on the toolbar as a shortcut.

1.2 The **Create new partition** dialog will be displayed. Enter the desired size of the new partition and click **OK**. As explained in previous sections, the **Align to beginning of block** setting controls whether the partition will be created at the beginning or end of the free space.

1.3 The new partition will be created but since not formatted, therefore also not addressable by any operating system. No logical drive designator will be assigned until a format is done. You have the choice of what sort of file system/OS to place into the partition. Please see the format option for more information.

Step two: Creating an extended partition.

Note: Only one extended partition per physical hard drive is possible or allowable.

2.1 As above, once again select free space using the instructions given in the previous example. In Windows, you click on the free space. In DOS you highlight the free space using the highlight bar. In both DOS and Windows you can then select the **Create** option using the **Partition** pulldown, or with the DOS version, just click the **<Create>** button. In Windows you can also use the Create button on the toolbar.

2.2 The **Create new partition** dialog will be displayed. Indicate the desired new partition size, check the **Extended Partition** box, then click OK. Partition Manager will automatically give you the option to create an extended partition in this free space if it is appropriate.

2.3 A new extended partition is created, containing one single large ***logical free*** *partition.* The creation of this extended partition does not in fact create any addressable logical drives, only tells the partition table to now recognize this area as one in which logical drives can be created. This area is now in fact designated as one in which you can at some point create any number of logical drives at some later point in time, however is now officially a real partition, which takes up space on the hard drive for the purpose of creating new logical drives.

Step three: Create a logical drive in the extended partition

3.1 Select the newly created ***logical free*** block of disk space (in Windows, right click the block in any of the windows. In DOS, select the block using the highlight bar). In DOS, press the ***<Create>*** button or pulldown the

Partition menu, then select **Create**. In Windows, the latter option also applies, but you can also just press the **Create** button on the toolbar as a shortcut.

3.2 The **Create new partition** dialog will be displayed. Enter the desired size of the new logical drive, then click **OK**. Do not use the entire ***logical free*** space, because we want to create a second logical drive within the extended partition.

3.3 A new partition (logical drive) will be created, designated as ***logical, unformatted***. The remaining free space within the extended partition, will still be designated by Partition Manager as ***logical free***.

Notes: Logical drives can be created only in extended partitions. If a second logical drive is desired to be created within the same extended partition after creating the first one, some free space must be left for the second logical drive.

Step four: Create a second logical drive within the extended partition

4.1 Click on (or select, using the highlight bar) the remaining ***logical free*** space in the extended partition (that space not taken by the previously created logical drive). In Windows, right click the partition in any of the windows. In DOS, select the partition using the highlight bar. In DOS, press the ***<Create>*** button or pulldown the **Partition** menu, then select **Create**. In Windows, the latter option also applies, but you can also just press the **Create** button on the toolbar as a shortcut.

4.2 The **Create new partition** dialog will be displayed. The remaining free space will be shown as default. Select all of it, then click **OK**.

4.3 The second partition (logical drive) will be created, designated as ***logical, unformatted***. Notice that there will be no further free space available within that extended partition, because we already used it all up with the last two logical drives we created.

What we have now on this hard drive, is one primary (bootable) partition, and one extended partition containing two logical drives. All these partitions are at this point unformatted and therefore not addressable by any operating system. None of these partitions will be addressable until they are formatted. At that point, the partition table will recognize these partitions and assign them logical drive letters. Making a working partition involves two steps. Number one, you have to create the partition, then two, format it using the file system/OS of your choice.

5.2 Copying a partition

5.2.1 Copying in the 1:1 mode

Configuration requirements for this exercise:

Disk size:	6.4 Gigabytes.
Partition one:	2 Gigabytes
Primary free:	700 Megabytes
Partition three:	1.7 Gigabytes
Primary free:	2 Gigabytes

Summary:
Primary partition one is to be copied into partition two primary free space (size: 2 Gigabytes). This involves one step only.

1. Position the highlight bar on the first partition, and click the ***<Copy>*** button in DOS or in DOS and Windows, click on the partition and select **Copy** from the **Partition** pulldown. In Windows, you can also either use the partition copy button on the toolbar or right click the partition and then select copy.

2. The copy dialog will appear and the 2 GB sized block (designated as ***primary free***) will be offered as target. Be sure to check **Copy with autoresize** off. After you click **OK**, you'll now find a new copy of the number one partition in the target area.

There should now be two identical partitions and the target partition will be either hidden or not depending on the **Do not hide target (entries)** setting (in the **Hard disk | settings** dialog).

Note:
To complete this operation successfully, the **Copy with autoresize** option box must be set to **off** (unchecked).

5.2.2 Copying in the autoresize mode

Configuration requirements for this operation:

Disk size:	6.4 Gigabytes.
Partition one:	2 Gigabytes, with 500 MB data in it.
Primary free:	700 Megabytes
Partition three:	1.7 Gigabytes
Primary free:	2 Gigabytes

Summary: Partition one is to be copied into the first primary free (the 700 MB partition). This exercise consists of a single step.

1. In both DOS and Windows, click on the first primary partition and select **Copy** from the **Partition** pulldown. In DOS you can also use the Copy button and in Windows, you may either use the partition copy button on the toolbar or right click the partition and then select copy.

2. The copy dialog will appear. Both ***primary free*** partitions are offered as targets. Select the first (700 MB) partition as the target. Be sure to check (activate) the **Copy with autoresize** option before you click **OK.**

3. You should now find a data copy of the 2 GB source partition in this 700 MB target partition. What we have now is two of the same (data-wise) identical partitions (which are nevertheless of differing sizes). There now exist two identical partitions and the target partition will be either hidden or not depending on the **Do not hide target entries** setting (**Hard disk | Settings** dialog).

Note:
To complete this operation successfully, the **Copy with autoresize** option box must be set to **on** (checked).

5.3 Moving a partition

Configuration requirements for this operation are the same as those for the exercise in 5.2.1

Disk size:	6.4 Gigabytes.
Partition one:	2 Gigabytes, with 500 MB data in it.
Primary free:	700 Megabytes
Partition three:	1.7 Gigabytes
Primary free:	2 Gigabytes

Here, however, the third partition (with 1.7 GB) is to be moved directly after the first partition, so that the 700 MB free space between these two partitions will completely disappear.

1. Position the highlight bar on the third partition, and click in DOS and Windows on the partition and select **Move** from the **Partition** pulldown. In Windows, you can also either use the move partition button on the toolbar or right click the partition and then select move.

2. The move partition dialog window will appear and both ***primary free*** partitions are offered as targets, namely the 700 MB partition preceding the selected (to be moved) partition, and the subsequent one with 2 GB. Select the preceding 700 MB partition as target for the move and click **OK**.

What you should have now are two adjacent (sequential) partitions with a 2.7 GB free disk space following them.

5.4 Enlarging or shrinking partitions (resize function)

The shrinking of a partition is always possible, specifically to the size of the available data contained within that partition. This shrinking to the data dependent minimum size is selfexplaining and therefor needs no further

explanation. Only a few simple steps are necessary, selecting the partition, push the resize button and enter the required new smaller size.

Enlarging a partition, on the other hand (just as in the case of creating a new partition), is executable in one step only in the case of there being primary or logical free disk space directly following the partition to be enlarged. In all other cases, the required free space at the end of the partition must be made available first.

5.5 Enlarging a partition at the expense of (shrinking of) another partition

If no free disk space (either primary or logical free) exists either at the beginning nor the end of the partition desired to be enlarged, then the partition in question can be enlarged only at the expense (shrinking) of the partition either preceding or following the partition to be enlarged.

In other words, either the directly preceding partition, or the directly following partition must first be shrunken, in order that the in-between partition can be enlarged. If the preceding partition is shrunken, the (to be enlarged) partition can then be moved into the free space made available by the shrinking of the preceding partition.

For example, in the case of an extended partition, it is sufficient to move the head of the partition forward. Another possibility exists in that one of the two adjacent partitions can first be moved or copied to make the needed free space.

5.5.1 Enlarging a primary partition at the expense of another primary partition.

Configuration requirements:

Disk size: 2 Gigabytes.
Partition one: 1 Gigabyte, with 700 MB data in it. Primary, bootable.
Partition two: 1 Gigabyte, with 500 MB data in it. Primary, but no SBR.

Summary:

The first partition is to be enlarged to a size 200 MB larger than its current size, at the expense of the second partition, which of course means that the second partition must be shrunken by 200 MB to accommodate the first partition.

This exercise involves three steps. First the second partition is reduced in size by 200 MB, which creates the free space after it. It must then be moved into the end of this block to create the free space between the two partitions (after the first partition). Then the first partition can be expanded (enlarged), using the primary free space after it.

Step one: Decrease the size of partition two by 200 MB

1.1 In DOS, position the highlight bar on the second partition, and click the ***<Resize>*** button. In Windows, click on the second partition and select **Resize** from the **Partition** pulldown. In Windows, you can also either use the Resize Partition button on the toolbar or right click the second partition and then select Resize from the popup.

1.2 The **Resize Partition** dialog window will be displayed and you can then resize the partition anywhere in the range of 500 MB (minimum, the data size) to 1000 MB (maximum, the size of the partition, since there is no primary free space after it). Click the size field with your mouse and delete the default entry. Enter the desired target size of 800 MB (shrinking 200 from the original 1 GB). The resizing takes place after you click **OK**.

Step two: Move partition two into the free space (to the end of the hard drive)

2.1 After the resize, you should have a smaller partition of 800 MB and a free space of 200 MB, which should be designated as ***primary free***. Stay on the 800 MB partition and select the move (or copy) function as follows: In DOS and Windows, select **Move** from the **Partition** pulldown. In Windows, you can also either use the move partition button on the toolbar or right click the partition and then select move.

2.2 The move dialog window will be displayed, and the 200 MB block designated as ***primary free*** will be displayed. After clicking **OK**, you should find that the parallel move of the 800 MB partition resulted in its being placed at the end of the hard drive, and the free disk space (the 200 MB of type ***primary free***), winds up in the middle, between the two true primary partitions.

Step three: Enlarge the first partition 200 MB

3.1 Select the first partition with the highlight bar (DOS) or by clicking on it (Windows), and selecting the **Resize** function, as in step one, above.

3.2 The **Resize Partition** dialog window will be displayed and you can then resize the partition anywhere in the range of 700 MB, up to 1200 MB. (Partition Manager automatically recognizes the following primary free and adds it to the maximum allowable size). Click the size field with your mouse and delete the default entry. Enter the desired target (total) size of 1200 MB. The resizing takes place after you click **OK**.

The task is now complete and you'll have now two primary partitions resized to the desired sizes.

5.5.2 Enlarging a primary partition at the expense of a logical drive

Configuration requirements:

Disk size:	2 Gigabytes.
Partition one:	1 Gigabyte (primary and bootable) partition, containing 700 MB of data.
Partition two:	1 Gigabyte extended partition (the space for logical drives), containing …
Partition three:	1 Gigabyte logical drive, containing 500 MB of data.

Summary:
Partition one is to be enlarged by 200 MB, at the expense of the logical drive within the extended partition, which means that not only the logical drive, but also the extended partition must both be reduced in size by this same 200 MB.

This exercise involves four steps. The logical drive must first be reduced 200 MB, then moved to the end of the extended partition (two steps). Then the extended partition must be reduced in size by 200 MB by shifting its head inward, to create extra primary free space between the partitions. Finally the primary partition can be enlarged, using the primary free space after it.

Step one: Reduce the logical drive 200 MB in size

1. Select the third partition (the logical drive) and then the **Resize** function, as described in steps one and three of the previous exercise. The **Resize Partition** dialog window will be displayed and you can then resize the logical drive anywhere in the range of 500 MB, up to 1000 MB. Click the size field with your mouse and delete the default entry. Enter the desired target size of 800 MB. The resizing takes place after you click **OK**.

Step two: Move the logical drive to the end of the extended partition

2.1 After the resize, you should have a smaller logical drive of 800 MB and a free space of 200 MB, which should be designated as ***logical free***. Stay on the 800 MB logical drive and select the move (or copy) function as follows: In DOS and Windows use the **Partition | Move** pulldown, in Windows you may also click the move partition button on the toolbar.

2.2 The move dialog window will be displayed, and the 200 MB block designated as ***logical free*** will be displayed. After clicking **OK**, you should find that a parallel move of the 800 MB logical drive resulted in its being placed at the end of the extended partition, and the free disk space (the 200 MB of type logical free), winds up at the beginning of the extended partition.

Step three: Reduce the size of the extended partition

3.1 After the move, you should now have the smaller logical drive of 800 MB at the end of the extended partition, and the free space of 200 MB (which should be designated as ***logical free***) at the beginning of the extended partition. Now select the extended partition (partition two), and select the resize function as follows: In DOS, click on the ***<Resize>*** button. In Windows, click the Resize Partition button on the toolbar, or right click the extended partition and click on resize.

3.2 The **Resize extended partition** dialog window will be displayed. It will at this point be possible to shift the head only backward 200 MB. A forward shift of the head, as well as shifting of the end of the partition is not possible and is therefore not available as an option (showing 0, 0). Enter 200 in the head shift field, and press **OK**.

Step four: Enlarging partition one to a size of 1200 MB

4.1 After resizing the extended partition, the logical free space that was there disappears. In its place, free space (of type primary free) of the same size now appears, in between the primary partition (one), and the extended partition (two).

4.2 Now select partition one and then the **Resize** function, once again.

4.3 The **Resize Partition** dialog window will be displayed and you can then resize the partition anywhere in the range of 700 MB, up to 1200 MB. Click the size field with your mouse and delete the default entry. Enter the desired target size of 1200 MB. The resizing takes place after you click **OK.**

The task is now complete and you should have as a result, a primary partition of 1200 MB and an 800 MB logical drive within an extended partition of the same size.

5.5.3 Enlarging a logical drive at the expense of another logical drive

Configuration requirements:

Disk size:	2 Gigabytes.
Partition one:	1 Gigabyte, primary and bootable, containing 700 MB of data.
Partition two:	1 Gigabyte extended partition, containing the two following logical drives:
Partition three:	500 MB, logical drive one, containing 100 MB of data.
Partition four:	500 MB, logical drive two, containing 450 MB of data.

Summary:
Partition four (the second logical drive) is to be increased 250 MB in size, at the expense of the first logical drive. All significant events will occur within the boundaries of the extended partition containing the two logical drives, thus partition one (the primary) is not relevant nor affected by this operation.

This consists of three steps. First we reduce the first logical drive by 250 MB, then move the second logical drive into the free space, adjacent to the first again. Finally we expand the second logical drive to fill the logical free space (following it), created with the prior operation.

Step one: Reduce the size of the first logical drive by 250 MB

1. Select the first logical drive (partition three) and then invoke the **Resize** function. The **Resize Partition** dialog window will be displayed and you can then resize the partition anywhere in the range of 100 MB, up to 500 MB. Click the size field with your mouse and delete the default entry. Enter the desired reduced target size of 250 MB (500 MB – 250 MB). The resizing takes place after you click **OK.**

Step two: Move the second logical drive into the free space

2.1 After completion of the prior operation, you should now have a logical drive reduced by 250 MB, and between the two logical drives, a 250 MB free disk space designated as ***logical free***. Now select the second logical drive and then the move function (In DOS and Windows use the **Partition | Move** pulldown, in Windows you may also click the move partition button on the toolbar).

2.2 The move dialog window is displayed and the newly created 250 MB logical free space is offered as target for the move. After clicking **OK**, a parallel move of the second logical drive occurs into the beginning of the logical free space, and creating 250 MB of logical free space after the second logical drive (at the end of the extended partition).

Step three: Increase the size of the second logical drive

3.1 After the move, the two logical drives will once again be together and the 250 MB logical free space winds up at the end of the extended partition. Once again select the second logical drive (if not already selected), and then invoke the **Resize** function.

3.2 The **Resize Partition** dialog window will be displayed and you can then resize the second logical drive anywhere in the range of 450 MB, up to 700 MB. Click the size field with your mouse and delete the default entry. Enter the desired target size of 700 MB. The resizing takes place after you click **OK.**

You should now have two logical drives with different sizes. The first logical drive is 250 MB smaller and the second that much larger.

5.5.4 Enlarging a logical drive at the expense of a primary partition

Configuration requirements:

Disk size:	2 Gigabytes.
Partition one:	1 Gigabyte, primary and bootable, containing 300 MB of data.
Partition two:	1 Gigabyte extended partition, containing the following logical drive:
Partition three:	1 Gigabyte logical drive containing 800 MB of data.

Summary:
Partition three (and thus also the extended partition two containing it), is to be increased in size to 1.5 GB at the expense of (the primary) partition one.

This will consist of four steps. First we will cut the size of the primary partition in half, to make 500 MB free space after it (step 1). The second (extended partition) can then be expanded backward (step 2). *(If this were a primary, it would not be possible, using this method).* Now the logical drive within the extended partition can be moved to the beginning of the extended partition (step 3), and finally the logical drive can be enlarged to fill the entire extended partition (step 4).

Step one: Reduce primary partition one 500 MB in size

1. Select partition one and then the **Resize** function. The **Resize Partition** dialog window will be displayed and you can then resize the second logical drive anywhere in the range of 300 MB, up to 1000 MB. Click the size field with your mouse and delete the default entry. Enter the desired target size of 500 MB. This still leaves partition one 200 MB to expand. The resizing takes place after you click **OK.**

Step two: Enlarge the extended partition (partition two)

2.1 After completion of the previous step, you now have a primary partition reduced to 500 MB, followed by primary free space of 500 MB, and then directly following that, the extended partition of 1 GB containing the logical drive of the same size. Now select the extended partition and once again invoke the resize function.

2.2 The **Resize extended partition** dialog window will be displayed. It will at this point be possible to shift the head only forward 500 MB. A backward shift of the head, as well as shifting of the end of the partition is not possible and is therefore not available as an option (showing 0, 0). Enter -500 in the head shift field, and press **OK**. Note this type of operation is available for extended partitions only.

Step three: Move the logical drive to the beginning of the extended partition

3.1 The (primary) free disk space of 500 MB, that was previously between the primary partition (one) and the extended partition two, disappears and in its place appears a logical free space of the same size at the beginning of the extended partition, followed by the 1 GB logical drive.

3.2 Select the logical drive and then the move function. (In DOS and Windows use the **Partition | Move** pulldown, in Windows you may also click the move partition button on the toolbar).

3.3 The move dialog window is displayed and the newly created 500 MB free space (designated as ***logical free***) is offered as target for the move. After clicking **OK**, a parallel move of the second logical drive occurs into the beginning of the logical free space, and creating 500 MB of ***logical free*** space after the second logical drive (at the end of the extended partition).

Step four: Increasing the size of the logical drive

4.1 The 500 MB free disk space of type logical free is now at the end of the extended partition. Position the highlight bar (or click, in Windows) on the 1 GB logical drive and select the **Resize** function.

4.2 The resize partition dialog window will be displayed and you can then resize the second logical drive anywhere in the range of 800 MB, up to 1500 MB. Click the size field with your mouse and delete the default entry. Enter the desired target size of 1500 MB. The resizing takes place after you click **OK.**

This concludes the exercise and now you should have a primary partition of 500 MB, and an extended partition of 1.5 GB containing a logical drive of the same size.

5.6 Preparing for subsequent installation of an additional operating system

Configuration requirements:

Disk size:	4 Gigabytes.
Partition one:	2 Gigabytes, with 300 MB of data in it. Primary and bootable with Windows 95.
Partition two:	2 Gigabytes, extended partition, containing the following logical drive:
Partition three:	2 Gigabytes, logical drive containing 800 MB of data.

Summary:
You want to add Windows '98 to your system, but yet wish to retain the native Windows '95. In any case, an additional primary partition must be created. The following solution is only one of many possibilities.

Reduce the first partition from 2 GB to 1 GB. Create a new primary partition in the resulting free space. Format this new partition, then hide the original primary and set it inactive. Set the new partition active, and install the Win 98. You can then boot from the Win 98 partition or install one of the boot managers to select which partition/OS to boot from.

Step one: Create a new primary partition

1.1 Reduce the primary partition one to 1 GB
1.2 Create a second primary partition within the resulting ***primary free*** space.
1.3 Format this new primary partition using FAT-32.

These steps will not be discussed in detail here, as the previous sections cover these types of operations in some detail.

Step two: Preparation for the installation of Windows '98

2.1 Set the partition with Windows '95 inactive and then hide it. In DOS, position the highlight bar on the primary partition one, then click the **Set Inactive** button, then the **Hide** button. In Windows, click (or right click) the primary partition one in either the right or left windows, then (if you right clicked), you can select the appropriate subfunction from the popup, or if you clicked on it, use the **Partition | Set Inactive** and **Partition | Hide** pulldown functions.

This partition should then be designated as hidden and inactive in the appropriate columns of the detail line for that partition. In DOS, this is shown as ***<hid>*** (or nothing), in the **Hidden** column. DOS will show either Yes or No in the **Bootable** column. Windows shows either Yes or No in both the **Active** and **Hidden** columns.

2.2 Set active the new primary partition created in 1.2 above (in which Windows '98 is to be installed), using the technique in the previous step. In DOS the button will be labeled ***<Set Active>***. Also make sure the partition is not hidden. If it is, unhide it, using the opposite function, as described in the previous step.

Now you can install Windows '98 into the second primary partition. The Windows '98 installation program will recognize the second primary partition as the only possible target for the installation (by virtue of its being the only active primary partition), and will install the system boot record as well as all required Windows program and data files.

After the installation and the self-configuration of Win 98 (after some restarts), you can install BootManager Easy or BootManager 99, to enable you to select whether you want to boot from the Windows '95 partition or the Windows '98 partition, every time you boot the system.

6. Reference Section

6.1 64 KB FAT-cluster size for Windows NT

This option allows the creation of a 64 KB cluster size for FAT 16 partitions. When this option is set on (box checked), then the formatting of partitions larger than 2 GB will result in the enabling of FAT-16 partitions, otherwise these are unavailable and the only FAT system permitted will be FAT-32. The standard installation defaults to *no FAT-64 clusters allowed.* For further information, see Sections 2.2.2 and 3.3.2

This achieves the doubling of the upper sector limit from 64 sectors (512 bytes) to 128 sectors. Partitions converted/copied in such a way will subsequently be accessible only by Windows NT, however these will then no longer be recognizable by DOS. Partition manager detects both cluster sizes and creates (in the standard mode) a 32 KB cluster size. Only if this parameter is set, is a 64 KB cluster size created.

6.2 The extended OS/2 attribute and FAT

In the event that OS/2 is to be installed in a FAT rather than an HPFS partition, the extended data attribute (that is not available in a FAT) will be stored in a data called EDATA. This data contains references to the data that the FAT system had stored. If the partition is modified by Partition Manager then PM will also change these references in the data EDATA so that OS/2 can operate error-free after the partition modification.

6.3 The automatic resuming of program execution after a power shortage

The following refers to a critical operation such as a copy being interrupted by a power shortage or other unanticipated event which caused the operation to be interrupted prematurely.

This occurs when, at the beginning of said operation, the program writes a code into the MBR, which immediately after a reboot, is again accessed by the program. If the program is terminated normally, the code of the MBR will still be removed, so that the normal boot process will be resumed.

Any time an operation is interrupted in this way, once the PC is rebooted, the following message will appear:

The last operation on drive C was not completed successfully.
Resume operation (Yes/No) ?

By clicking on **Yes,** the operation will be resumed. If **No** is clicked, the operation will be cancelled and the data on the partition in process can be lost.

6.4 Enabling/disabling the automatic detection of bad sectors on the hard drive

Bad sectors on the hard drive can be automatically detected in all PM functions this applies to, by enabling the surface test option in the **Hard Disk | Settings** dialog. This setting applies to any Partition Manager function which writes to the disk.

Depending on the setting of the surface test option (see 2.2.2 **Surface Test** and the following section 6.5), either a **Normal** or **Intensive** verification of the hard drive surface will be done. An entry of **None** results in no surface check being performed.

6.5 The handling of defective sectors during a simple copy operation

Summary:
This option is controlled by the **Surface test** setting, in the **Hard disk | Settings** dialog window.

By setting of this parameter to one of the three options, the usability (physical state) of the disk space involved in a partition **create, copy, move** or **format** operation will be verified. The surface test, if enabled, (as described above), will be done prior to the selected operation (ie copy, etc.)

Surface test settings and their effect:

None: No physical examination of the disk space is performed.

Normal: Data on sectors written is compared to the sectors read. A standard read/write verify of each sector in the partition occurs.

Intensive: Read and written sectors are re-read and compared a second time.

The standard (and recommended) setting is **Normal**. The **None** setting is the quickest but will cause errors if it encounters defective sectors. The **Intensive** method is recommended only for situations in which multiple defective sectors are anticipated. In this instance, two comparisons will often detect a bad sector where one may not.

With the **Normal** and **Intensive** settings, a test of the target blocks is first performed and then the copy, etc. Using the **None** setting, no test is performed, and upon encountering a defective sector (upon which no data can be written), the surface test is aborted with a "*critical error*". It is therefore highly recommended to set this option at least to **Normal**, particularly when first creating and formatting a partition, to ensure that the partition is free of defective sectors. Once a defective sector is identified, the OS will from that point on, avoid that sector.

If Partition Manager detects bad sectors on the target drive (using the **Normal** or **Intensive** settings) when copying partitions without automatic size adjustment, then the size adjustment is automatically enabled, so the bad sectors will subsequently be avoided. The copy routine uses the "automatic size adjustment" in the following way: The size of the target partition and the source partition will coincide), but the data-cluster will be newly regrouped, to avoid the bad sectors. At the same time, the defective sectors will be identified as "bad sector", so they will not be used in the future.

This mode of handling defective sectors works only when Partition Manager recognizes the file system of the partition that is being checked.

Enabling this option is the only real insurance against data loss (with the **Surface test** option set to either **Normal** or **Intensive**).

6.6 Running PCs with EZ-Drive installed

Programs like EZ-Drive read the BIOS and partition parameters upon booting and make these values memory-resident, to make them (translated) available for the use of whatever operating system is to be booted.

In any case, since Partition Manager contains its own read and write processes using the booted DOS (either the internal Windows-DOS or an external DOS, like the provided basic DOS), EZ-Drive must always be loaded before Partition Manager. If you boot from a drive with EZ-Drive installed, then EZ-Drive will be active.

In case you want to boot Partition Manager from a floppy diskette, you have to interrupt the normal EZ-Drive boot sequence with either the ***<Space>*** or ***<Ctrl>*** keys, and then change the EZ-Drive option *"boot from disk"*, to turn it on.

We advise with great emphasis to be extremely careful when using EZ-Drive in combination with other hard drive utilities like boot programs and partitioning utilities. Better yet, avoid these types of combinations altogether.

6.7 Using Partition Manager and Boot Manager in combination

Partition Manager can change the allocation of a hard drive altogether, as in creating new partitions, and moving existing partitions to an entirely different location on the hard drive. If a boot manager is installed when these operations are to be performed, that boot manager must be reinstalled to recalculate the new locations of the partitions involved, with respect to the new partition table, so it will be able to subsequently find the partitions for booting, etc.

You also have re-evaluate partitions that were hidden from Boot Manager by Partition Manager or other partition tools, by either re-analysis or explicit setting of the hidden attribute. Contrary to this, inactive primary partitions are always visible to the Boot Manager.

6.8 Error messages

«**Incorrect file system! Use CHKDSK**»- file system mismatch. Partition Manager can not work with such a partition. You need to run special utility programs (for example CHKDSK or ScanDisk) to fix the file system mismatch, then run the program again.

«**Incorrect directory structure! Use CHKDSK**» -
file system mismatch. Partition Manager can not work with such a partition. You need to run special utility programs (for example CHKDSK or ScanDisk) to fix the file system mismatch, then run the program again.

«**File I/O error!**» - this error occurs during a read or write to a file. If the file is on a removable disk check the presence of that disk. If the file is on a hard disk there might exist a hardware malfunction.

«**Disk I/O error!**» – No access to the hard disk; the hard disk may have any kind of hardware problem, like beeing shut down (remote hard disk) or an energy save program.

«**Bad BOOT-area – disk is unusable!**» - The boot record of the OS in this partition is unusable (corrupted). It is therefor not possible to boot the OS in this partition.

«**Incorrect data in partition!**» The information found in the partition differs from the data entry in the partition table. Generally, such a partition can not be used.

«**Partition parameters overflows specified limitations!**» – The sum of all partition sizes specified in the partition table is greater than the size of the whole hard disk. This problem usualy occures when the hard disk has been partitioned in a different adres mode as it is used now.
To solve this problem try to find the right adres mode (in the BIOS).

«**Crossed partitions!**» - The data in the partition table is invalid as far as it concerns the end of one and the begin of an another partition. The partitions are overlapping, which means the end of one partition is inside the next partition. This problem usualy occures when the hard disk has been partitioned in a different adres mode as it is used now. You may try to find the original adres mode (there are 3 modes LBA, Large, Normal), otherwise you have to repartition the hard disk.

«**FAT16-disk creation is impossible!**
This message will be displayed during resizing or copying in autoresize mode
if the new partition size is greater than 2 GB. This size is too big for a FAT16 file system. Nevertheless, the program can automaticaly convert the existing FAT-16 file system to FAT-32. You should be aware that FAT-32 is not supported by any DOS and also not by Windows 95 release 1.

«**BootManager or EZ-Drive system already present. Can not initialize BootManager Easy**» – the programm found that an another bootmanager is already installed and will therefore not install BootManager Easy. To install BootManager Easy the installed bootmanager has to be deinstalled.

«**Not enough memory!**» - there is not enough free memory (RAM) to execute the current operation. For example, the resizing of large partitions is very memory demanding and may go above the available resources. In windows you may try to increase the size of virtual memory but in DOS you need to add more physical RAM.

«**Incorrect restart parameters!**» - The data necessary to restart from an interrupted operation is not sufficent or corrupted. There is no restart possible.