

```
md. finance
```

If the new directory will be a subdirectory of the current directory, you need only type the part of the path that is not current.

Whenever you have a group of files that belong together (such as specialized files from the same program, files that belong to a certain person, files that are from the same project) consider storing them in their own directory. To create the directory, you use the **md** (make directory) command. For example suppose that the current directory is the root (\). To create a subdirectory to the root called **WORD**, you enter the following command:

```
md word
```

MS-DOS makes a new subdirectory of the root directory called **\WORD**. Instead of **md** you can use **mkdir** and get the same result. For example, this command does exactly the same thing as the preceding example:

```
mkdir word
```

The **md** command makes a subdirectory in the current directory unless you specify otherwise. For example, if the current directory is **\WORD**, to make a directory called **\WORD\HOME**, you need only type the last subdirectory in the path:

```
md home
```

MS-DOS makes a subdirectory in the **\WORD** directory called **HOME**. If you are not making a subdirectory of the current directory, you have to type the entire path of the new directory. For example, if the current directory is **\WIN**, to make **\WORD\HOME**, you would enter the following command:

```
md \word\home
```

If you include a drive letter in the **md** command, you can make a directory on a disk that is not the current disk. For example, suppose you wanted to copy the files in the current directory, **C:\WORD\HOME** to the disk in drive **A**. To keep the files separate, you might want to create a subdirectory on **A** to hold them. To create a directory on the disk in drive **A** without having to leave the current directory, you enter this command:

```
md a:\home
```

Shell ► **To create a directory as a subdirectory of the currently selected directory:**

1. Display the File System screen.
2. Select the directory for which you want to create a new subdirectory.
3. Choose Create Directory from the File menu.

MS-DOS displays the Create Directory dialog box. The currently selected directory is shown as the parent directory.

4. Type the name of the new directory in the xx box.
A name can be up to eight characters long, and you may add a three-character extension.
5. Choose OK or press ENTER.
MS-DOS adds the new directory to the directory tree, as a subdirectory of the currently selected directory.

You can add directories at any level in the tree, but the directory you create will always be a subdirectory of the currently selected directory.

Moving Between Directories

The disks in each drive of your system have a current directory. If you have two floppy disks and one hard disk, there are three current directories.

If a disk has no subdirectories, the root directory is always the current directory for that disk. If a disk has subdirectories, you can use the `cd` (change directory) command to move from the one directory to another.

Finding Out Which Directory Is Current

In Brief

To display the current directory on the current drive, enter a `cd` command with no parameters as in the following command:

```
cd
```

If the command prompt is set up to show you the path of the current directory on the current drive, you need only look at the prompt to see which directory is current. If the prompt does not show the current directory or if you want to see which directory is current on another drive, you can use the `cd` command. For example, the following command displays the path of the current directory on the current drive:

```
cd
```

The following command displays the current directory on drive B:

```
cd b:
```

It does not make drive B current.

Changing Directories

In Brief

To move to a different directory on the current disk, use the `cd` command (also called `chdir`). For example, the following command changes the current directory to `\WIN\EXCEL\FINANCE`:

```
cd \win\excel\finance
```

Use the path `..` to change to the parent directory current as in the following command:

```
cd ..
```

Suppose that the current directory is the root (`\`). You can move to the `\WORD` directory and make it current by entering the following command:

```
cd word
```

Instead of `cd` you can use `chdir` and get the same result. For example, this command is the same as the previous example:

```
chdir word
```

As in the `md` command, unless you specify otherwise, MS-DOS assumes you want to change to a subdirectory of the current directory. For example, if the current directory is `\WORD`, to change to `\WORD\HOME`, you need only type the last subdirectory in the path:

```
cd home
```

If the current directory is `\WIN`, to change to `\WORD\HOME`, you have to enter the entire path:

```
cd \word\home
```

To change to the parent of the current directory (the directory one level closer to the root) you can use the `..` combination. For example, if the current directory is `\WORD\HOME`, you can change to `\WORD` with this command:

```
cd ..
```

No matter which directory is current, you can change to the root directory of the current drive by entering this command:

```
cd \
```

You can't use the `cd` command to change the current drive, but you can use it to change the current directory on a drive that is not current. For example, suppose

Deleting Directories

the current drive is A. To change the current directory on drive C to \WORD *without making drive C current*, use the following command:

```
cd c:\word
```

Now, when you type C: in a command without a directory, MS-DOS assumes you want to use the \WORD directory. For example, if A is the current drive and \WORD is the current directory on drive C, the following command copies all the files from the current directory on drive A to C:\WORD:

```
copy *.* c:
```

The current directory on a drive is the root unless you change it.

Shell

In the Shell, the files in the currently selected directory are displayed on the right side of the File System screen. The name of the currently selected directory appears in the directory indicator just above the drive indicator in the upper-left corner of the screen. In the directory tree, the currently selected directory is highlighted.

For information about selecting directories, see "XX".

Deleting Directories

In Brief

To delete a directory, use the rd command (also called rmdir), as in the following example:

```
rd \win\excel\finance
```

MS-DOS removes the FINANCE subdirectory from the \WINEXCEL\ directory. The directory you remove must contain no files or subdirectories.

If you no longer need a directory, you can remove it with the rd command. Before you remove a directory, it must be empty and it must not be the current directory. It cannot contain any files or subdirectories. If you display the contents of an empty directory with a dir command, it should have only two items listed: the period (.), which stands for the directory itself, and the double period (..), which represents the parent directory.

If a directory is empty, you can enter a rd command to delete it. For example, suppose you have a directory called \WORD. Before you can remove the directory, you must delete its contents with the following command:

```
del \word\*.*
```

You can then enter the following command to remove the directory:

```
rd word
```

MS-DOS removes \WORD from the current drive. If \WORD has subdirectories, you must remove them before you can remove \WORD.

Instead of `rd` you can use `rmdir` and get the same result. For example, this command is the same as the preceding example:

```
rmdir word
```

As with the `md` and `cd` commands, MS-DOS assumes you want to remove a subdirectory of the current directory. For example, if the current directory is \WORD, to remove \WORD\HOME you need only type the last subdirectory in the path:

```
rd home
```

If the current directory is \WIN, to remove \WORD\HOME, you must enter the entire path:

```
rd \word\home
```

If you include a drive letter in the `rd` command, you can remove a directory from a disk that is not the current disk. For example, to remove the A:\HOME directory while the current drive is C, use the following command:

```
rd a:\home
```

Shell ► **To delete a directory or subdirectory:**

1. Make sure the directory does not contain any files or subdirectories.
2. Select the directory.
3. Choose Delete from the File menu.
MS-DOS displays the Delete Directory dialog box.
4. Choose option 2 to delete the directory. Choose option 1, or Cancel, if you don't want to delete the directory.
5. Choose OK or press ENTER.

If you try to delete a directory that contains files or subdirectories, the shell displays the Delete Directory dialog box with an "Access denied" message. If this message appears, delete the files and subdirectories in the directory and try again.

Copying Directories

To copy directories and their subdirectories, you use the **xcopy** command. The **xcopy** command is similar to the **copy** command. Both commands copy files from one directory or disk to another. However, while the **copy** command is designed to work with single files or groups of files, the **xcopy** command is designed to work with single directories or groups of directories. Both commands create new files in the "destination" directory, but only the **xcopy** command creates new subdirectories as well.

Copying All Files in a Directory

In Brief

To copy single directories (without subdirectories), use the **xcopy** command with no switches. For example, the following command copies all files in the **C:\WINEXCEL\FINANCE** directory to the **\FINANCE** directory on drive A:

```
xcopy c:\win\excel\finance a:\finance
```

If you want to copy all the files in a directory to another existing directory, the **copy** command is equivalent to the **xcopy** command. Unlike the following **copy** command which uses wildcards to copy all of the files from drive A to drive B:

```
copy a:*. * b:
```

Because the **xcopy** command assumes you want to copy whole directories, you don't need to use wildcards. For example, the following **xcopy** command performs exactly the same function as the preceding command:

```
xcopy a: b:
```

Before MS-DOS copies the files, it displays a "Reading source file(s)..." message while it prepares to copy the files. As in the **copy** command, MS-DOS displays the names of the files it copies and tells you how many files were copied when the operation is complete.

Shell ► To copy all of the files in a directory to another existing directory:

1. Select the disk drive or directory containing the files you want to copy.
2. Choose **Select All** from the **File** menu to select all the files in the directory.

If you don't want to copy all of the files, you can cancel the selection of individual files by holding down the **CTRL** key and clicking them, or by pressing **SHIFT+F8**, moving the selection cursor to the files you want to deselect, and

pressing CTRL+SPACEBAR. You can also use the Deselect All command on the File menu to deselect all the files and start again.

3. Choose Copy from the File menu to copy the files.
MS-DOS displays the Copy dialog box.
4. Type the path of the directory you want to copy to in the To box. If the directory is on a different drive, include the drive name.
5. Choose OK or press ENTER.

For more information about paths and drive names, see "XX".

Creating Directories as You Copy

In Brief

If the destination path in an `xcopy` command does not exist, MS-DOS creates it. For example, the following command copies all files from the root directory of drive A to the C:\ATMP directory:

```
xcopy a: c:\atmp
```

If the directory does not exist, MS-DOS asks you if you want to create it (to stop MS-DOS from prompting you by adding a backslash (\) to the end of the directory name).

You can use the `xcopy` command to create directories as you copy files. For example, suppose you want to copy all the files on the disk in drive A to drive C. Using the following `xcopy` command, you can tell MS-DOS to put the files in a directory called \NEWFILES:

```
xcopy a: c:\newfile
```

If the \NEWFILE directory does not already exist on drive C, MS-DOS creates it as a subdirectory of the root directory. Then, MS-DOS copies the files from drive A to the \NEWFILE directory. In this example, only the files in the root directory of drive A are copied. If there are subdirectories on drive A, MS-DOS does not copy them.

If you do not type a path, MS-DOS copies the files to the current directory.

Shell

To duplicate a directory that has several subdirectories, first use the Create Directory command on the File menu to create the same directory structure in the new location. Then copy the files from each of the source directories to each of the destination directories.

Copying Subdirectories

In Brief

To reproduce a directory structure completely in another directory or on another disk, use the /s and the /e switches. For example, the following command recreates the directory structure and files within C:\WIN on drive B:

```
xcopy c:\win b:\ /s /e
```

To tell MS-DOS to copy the files in a directory along with any subdirectories that have files, you add the /s switch to the xcopy command. For example, suppose the disk in drive A has the following directories on it: \SCHOOL, \WORK, and \HOME. The following command copies the files in the root directory of A, the three subdirectories, and all their files to the directory on drive C called \MEMOS:

```
xcopy a:\ c:\memos /s
```

The backslash (\) after the A: tells MS-DOS to start at the root directory. The /s tells MS-DOS to copy every file in every subdirectory that has files. MS-DOS copies files from the root directory of drive A to C:\MEMOS, from A:\SCHOOL to C:\MEMOS\SCHOOL, from A:\WORK to C:\MEMOS\WORK, and from A:\HOME to C:\MEMOS\HOME. If any of the directories do not exist on drive C, MS-DOS creates them. In this example, empty subdirectories on drive A are not copied.

To copy empty directories add a /e switch in addition to the /s switch. For example, if the disk in drive A has an empty subdirectory called \MISC, in addition to the three subdirectories mentioned above, MS-DOS would not have copied it. To include the empty directory, enter this command:

```
xcopy a:\ c:\memos /s /e
```

MS-DOS copies the same files as before, but now also creates an empty directory called C:\MEMOS\MISC. You can use the /s switch with or without the /e switch. However, if you do use the /e switch you must also include the /s switch.

Renaming Directories

In Brief

To rename a directory, use the xcopy, del, and rd commands. For example, the following sequence of commands rename the \OPS\STATS directory to \OPS\FIGURES:

```
xcopy \ops\stats \ops\figures
del \ops\stats\*.*
```

```
rd \ops\stats
```

If the original directory has subdirectories, add the /s and /e switch to the xcopy command, and delete each subdirectory separately.

► To rename a directory:

1. Copy the contents of the directory to a directory with the new name.
2. Delete the contents of the original directory
3. Delete the original directory.

For example, suppose you want to rename the C:\TEMP directory to C:\LETTERS. The first step is to copy the contents of the directory to a directory with the new name. You can accomplish this most easily with this xcopy command:

```
xcopy c:\temp c:\letters
```

MS-DOS creates a subdirectory of the root directory of drive C called LETTERS, and copies all the files from C:\TEMP into it.

The second step is to delete the files in C:\TEMP. To do this, use the following del command:

```
del c:\temp\*.*
```

MS-DOS asks you if you really want to delete all of the files in this directory. If you are sure the files were successfully copied to C:\LETTERS, enter y. If you have any doubt, enter n and use the dir command to double-check.

Once the C:\TEMP directory is empty, the final step is to delete the directory itself by entering the following rd command:

```
rd c:\temp
```

MS-DOS removes the old directory. As a result of this three-step process, the C:\TEMP directory has been renamed to C:\LETTERS.

Shell ► To change the name of a directory or subdirectory:

1. Choose Deselect All from the File menu if any files are selected.
2. Select the directory you want to rename.
3. Choose Rename from the File menu.
A dialog box appears.

4. Type the new name for the directory in the box.
5. Choose OK or press ENTER.
The directory name changes.

Updating Directories

Sometimes you will have two directories that should contain the same files. For example, when you are making backups, you will have a primary directory that contains the files you are working on and a secondary directory or disk that contains the backup or most recent version of the files. To keep the secondary directory current, use the replace command.

Replacing Outdated Files

In Brief

To replace the files in a destination directory that are older than the corresponding files in a source directory use the replace command with a /u switch, as in the following command:

```
replace c:\win\*. * a: /u
```

MS-DOS replaces the files in A:\ that have more recent versions in C:\WIN.

Suppose you have a directory called C:\OPS\STATS containing statistics files you periodically update. To keep a backup of these reports you can copy them to a floppy disk and periodically update them with this command:

```
replace c:\ops\stats\*. * a: /u
```

MS-DOS compares the files in C:\OPS\STATS with the files on the disk in drive A. If a file on drive A has a more recent version on drive C, MS-DOS replaces the older version. This form of the command does not add any new files to the backup disk; it only updates those that are already there.

Adding New Files

In Brief

To add files to the destination directory that are currently only in the source directory, use the replace command with an /a switch. For example, the following command compares the files in C:\WIN with those in A:\ and copies any that exist in C:\WIN, but not in the A:\ directory:

```
replace c:\win\*. * a: /a
```

In the example in the previous section, you used the **replace** command with the **/u** switch to update the files on the backup disk in drive A. To add files to the backup disk, you use the **/a** switch rather than the **/u** switch. For example, the following command compares the files in C:\OPS\STATS with the files on the disk in drive A:

```
replace c:\ops\stats\*. * a: /a
```

If there are any files in the C:\OPS\STATS directory that aren't on the disk in drive A, MS-DOS copies them over.

Telling MS-DOS Where to Look for Files

When you want to run a program, you enter its filename. In order to run the program, MS-DOS must first locate it. Likewise, when you type a filename as part of a command, or when you try to get to a file from inside a program, MS-DOS must locate the file. Unless you tell it otherwise, MS-DOS only looks for files in the current directory.

There are two commands you can use to tell MS-DOS which other directories to search for files:

- The **path** command tells MS-DOS which other directories to search for program files and commands.
- The **append** command tells MS-DOS to search additional directories for all other kinds of files.

Using the Path Command

In Brief

To have MS-DOS search specific directories for program or batch files, use the **path** command. For example, the following command tells MS-DOS to look in the three directories listed in addition to the current directory when it searches for program and batch files:

```
path \;c:\bin;c:\utilities
```

Each directory in the **path** command is separated by a semicolon (;). The first backslash (\) tells MS-DOS to search the root directory of the current drive.

One way to run a program stored in a directory that is not current is to enter its path and filename. Another way is to first make its directory current with the `cd` command. To avoid the extra work involved with either of these procedures, you can type all the paths you commonly use once in a `path` command. Then, if MS-DOS cannot find the file you type in the current directory, it searches the other directories you specified. The `path` command you enter remains active until you restart or reset your system.

For example, suppose you commonly run programs stored in the `C:\PBRUSH` directory, the `C:\WORD` directory, and the `C:\EXCEL` directory. You can save having to type these paths each time you want to use them by entering them once in the following `path` command:

```
path c:\word;c:\excel;c:\pbrush
```

Each directory is separated from the others by a semicolon (;). MS-DOS searches the directories in the order you type them (in this case, MS-DOS would search `\WORD` before `\EXCEL`). To have MS-DOS look first in the root directory of the current drive, you add it to the beginning of the list:

```
path \;c:\word;c:\excel;c:\pbrush
```

You can tell MS-DOS to search as many directories as you care to list in the `path` command as long as the command doesn't exceed 127 characters (not including the word *path*).

To set the search directories automatically each time you reset or restart the system, you can include a `path` command in your `AUTOEXEC.BAT` file. If you find that the search directories have already been set, it means that there is already a `path` command in your `AUTOEXEC.BAT` file. For more information, about `AUTOEXEC.BAT`, see *xx*.

Using the Append Command

In Brief

To have MS-DOS search additional directories for text or data files, use the `append` command. For example, the following command tells MS-DOS to look in the `C:\WORD\notes` directory in addition to the current directory for files that are included in commands or are requested by programs:

```
append c:\word\notes
```

The `append` command is similar to the `path` command except that it applies to non-program files. The `append` command is useful when you have directories that contain files you often use in commands.

For example, suppose you commonly use the files stored in the C:\PBRUSH\PICS directory, the C:\WORD\DOCS directory, and the C:\EXCEL\SHEETS directory. You can save having to type these paths each time you want to use them by entering them once in the following **append** command:

```
append c:\word\docs;c:\excel\sheets;c:\pbrush\pics
```

As in the **path** command, each directory is separated from the others by a semicolon (;). MS-DOS searches the directories in the order you type them (in this case, \WORD\DOCS would be searched before \EXCEL\SHEETS).

Be careful that you don't end up using the wrong file. For example, suppose there is a file called LIST.TXT in both the C:\WORD\DOCS directory and the C:\EXCEL\SHEETS directory. After entering the **append** command in the preceding example, you might be tempted to delete the C:\EXCEL\SHEETS\LIST.TXT file by simply entering:

```
del list.txt
```

If you did, MS-DOS would delete C:\WORD\DOCS\LIST.TXT, not C:\EXCEL\SHEETS\LIST.TXT as you had intended.

To set the **append** path automatically each time you start the system, you can include an **append** command in your AUTOEXEC.BAT file. If you find that an **append** path has already been set, it means that there is already an **append** command in your AUTOEXEC.BAT file. For more information, see xx.

Telling MS-DOS Where to Look for Files

Chapter 7

Managing Disks

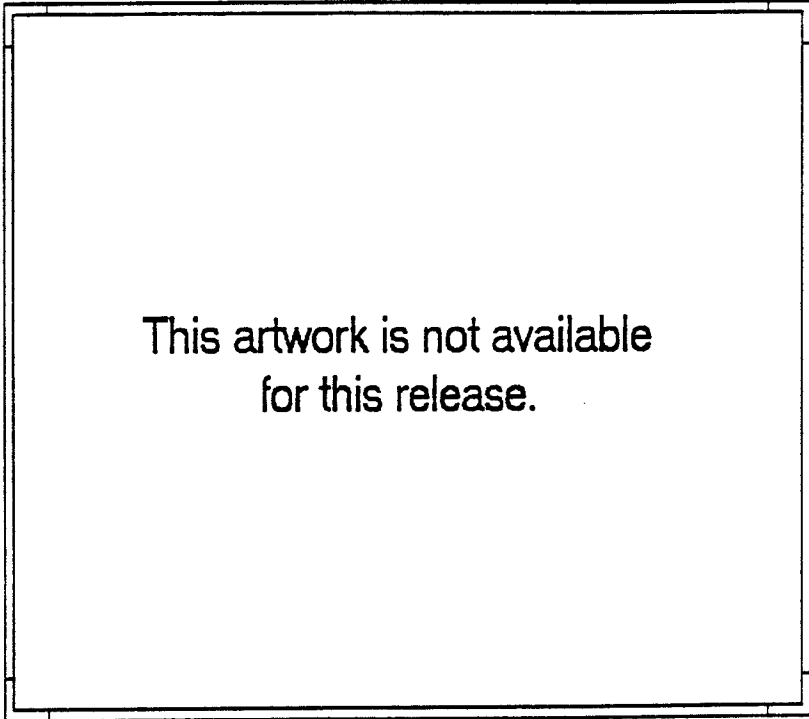
7

Disks store information permanently. In contrast to your system's random access memory (RAM), which is lost when you turn the power off, the information on a disk remains intact until you erase it. In addition, floppy disks provide a convenient method for transferring information. Almost all the programs you run come to you on floppy disks, and when you want to transfer files to other systems, you are likely to use floppy disks.

Types of Disks

The disks your system uses store information on magnetic surfaces. Floppy and hard disks get their names from the kind of magnetic surface they have. In floppy disks the magnetic surface is a thin flexible disk inside a protective plastic cover. Hard disks have two or more rigid disks stacked on top of each other in a sealed case. Hard disks are also called fixed disks because they remain in your system.

The information on all disks is divided into tracks somewhat like the grooves on a record. Each track is a concentric circle that can hold a certain amount of information. The more tracks a disk has, the more information it can hold. Floppy disks with the fewest tracks per side are called single sided or low density. Floppy disks with more tracks are called double sided, double density, quad density, or high density, depending on the number of tracks they have.



This artwork is not available
for this release.

dsk_1

Both hard disks and floppy disks have tracks. Hard disks hold more information than floppies because they have more sides and more tracks per side.

Floppy disks vary in physical size and the amount of information they can hold. The following table lists the major kinds of floppy disks that MS-DOS can work with and the amount of information they can hold:

- 5.25-inch, single sided/double density, 160K
- 5.25-inch, single sided/double density, 180K
- 5.25-inch, double sided/double density, 320K
- 5.25-inch, double sided/double density, 360K
- 5.25-inch, double sided/quad density, 1200K (1.2 MB)

- 3.5-inch, double sided/double density, 720K
- 3.5-inch, double sided/quad density, 1440K (1.44 MB)

Most floppy disks have labels that tell you which kind they are. However, sometimes the only way to tell one type of disk from another is to use a `dir` or `chkdsk` command to display information about the storage capacity of a formatted disk.

Bytes, Kilobytes, and Megabytes

Your files are measured in *bytes*. One byte is the amount of space it takes to store a single letter. A kilobyte is 1024 bytes. In this guide, the word *kilobyte* is abbreviated K.

A megabyte is 1024K (about a million bytes). In this guide, the word *megabyte* is abbreviated MB. For example, if a disk can store about 1.2 million bytes of information, it is a 1.2 MB disk. The following terms are equivalent:

$$1.2 \text{ MB} = 1228.8\text{K} = 1,258,291 \text{ bytes}$$

Kinds of Disk Drives

Hard disks are indistinguishable from hard disk drives because the two are a single unit. Once your hard disk is installed, you should never have to remove it. MS-DOS adjusts its operations to suit the hard drive you use.

On the other hand, for every type of floppy disk there is a floppy disk drive that is designed specifically to work with it. Not all types of floppy disks are compatible with all types of floppy disk drives. In general, the capacity of the disk must be less than or equal to the capacity of the drive in order for them to be compatible. For example, if you have a quad density, 5.25-inch disk drive that is designed to work with 1.2 MB floppy disks, you can use 360K double-density floppy disks with it. However, if you have a 360K drive, you cannot normally use 1.2 MB disks with it. If you are unsure whether a disk will work with a drive, try it. MS-DOS will display a "General failure error" message if the drive and disk are incompatible.

MS-DOS automatically adjusts its operations to suit the disk drive you are using. In some commands, you must type an additional switch if your disk drive and disk don't have the same capacity.

Formatting Disks

MS-DOS must format disks before using them. To format a floppy disk, MS-DOS divides the disk into smaller segments called *sectors* and sets up a file tracking system. Hard disks are already divided into sectors by the time you are ready to format them, so MS-DOS ensures that the disk is usable and set up the file tracking table. If you are using a new hard disk, however, you need to partition it before you can format it. You can use the MS-DOS installation program to partition and format the hard disk automatically when you install MS-DOS. For more information about installing MS-DOS on a hard disk, see "XX". You can also partition a new hard disk by using the `fdisk` command. For more information on partitioning a hard disk with `fdisk`, see "XX".

A sector is the basic unit of storage on a disk. Each sector on a hard disk or floppy holds one half of a kilobyte of information. When MS-DOS formats a disk it checks each sector for defects. MS-DOS marks "bad" sectors so that it will not store information there. When MS-DOS stores a file on the disk it uses groups of sectors called *allocation units*. The number of sectors depends on the size of the disk.

When you format a floppy or hard disk, MS-DOS reserves a small part of the disk for its tracking system. The allocation system consists of two parts: a file allocation table (which tracks the location of each file on the disk) and the root directory (which stores the name, size, creation date and time, and file attributes for the files on the disk).

One of the reasons you can't get information back from a newly formatted disk is because MS-DOS destroys any existing file table to create a new one. With floppy disks there is another reason. As MS-DOS divides a floppy disk into sectors, it destroys the contents of any sectors that were already there. When it formats a hard disk, the old file table is destroyed, but any existing sectors remain intact. The old sectors are filled as you copy new information to the disk.

During the formatting process, you can copy the three main MS-DOS system files to the disk. This makes the disk a system disk that you can use to start your system. For more information about system disks, see "xx" later in this chapter.

CAUTION The `format` command destroys all information on a floppy or hard disk. You should develop the habit of using the `dir` command before formatting your floppy disks or you might accidentally destroy important files. The `format` command is used to prepare your hard disk, so it is possible to format a hard disk drive if you accidentally enter the wrong drive letter. As a safety measure, MS-DOS displays a warning message if you attempt to format your hard disk. If you accidentally format your hard disk you may be able to use the

`unformat` command to recover its contents. For more information, see "xx" later in this chapter.

Formatting a Single Disk

In Brief

To format a floppy or hard disk for use with MS-DOS, use the `format` command. For example, the following command formats a floppy disk in drive A:

```
format a:
```

You must specify the drive that contains the disk you want to format.

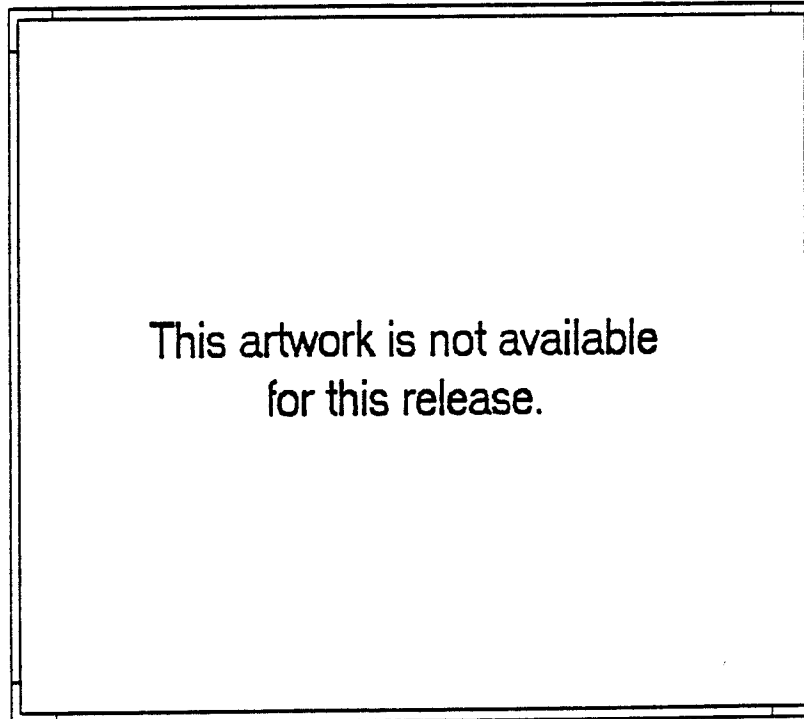
You format disks using the `format` command. For example, the following command formats the floppy disk in drive B:

```
format b:
```

MS-DOS requires you to specify the drive letter when you use the `format` command.

As it formats the disk, MS-DOS displays a message that tells you what percentage of the disk has been formatted. When MS-DOS is finished, it asks you if you want to give the disk a *volume label*. You can enter a label to describe the disk or press ENTER if you do not want a label. If you give the disk a volume label, be sure you also write it on the outside cover of the disk to help identify it.

After MS-DOS asks you for a label, it displays the following information that describes how the disk was formatted:



dsk_2

The "bytes total disk space" line tells you how much storage the disk contains. The number should be as much as or more than the capacity you expect for the disk. In the preceding example the total disk space is xx, a little more than the 1.2 MB the disk is designed for.

The "bytes used by system" line appears if you have transferred the MS-DOS system files to the disk. This line shows how much disk space is used by the three system files.

The "bytes in bad sectors" line tells you how much of the disk is unusable because of bad sectors. If there are no bad sectors, this line is omitted. The bad sectors that MS-DOS finds during the format will not be a problem because they have been marked. However, if a floppy disk has any bad sectors, it may be indicative of a growing problem with the disk. You should consider not using a floppy disk with bad sectors to store important files or backups. Most hard disks have a small num-

ber of bad sectors. In general, the portion of a hard disk taken up by bad sectors should be a very small fraction of the total space available on the hard disk. The exact permissible amount depends on the manufacturer of your hard disk.

The "bytes available on disk" line is the total disk space minus the space taken up by the system and any bad sectors. If the disk does not contain the system files and there are no bad sectors, this number should match the total disk space number.

The "bytes in each allocation unit" line and the "allocation units available on disk" line show you the way MS-DOS has divided the available disk for file storage. If you multiply the two numbers on these lines the result will equal the number on the "bytes available on disk" line.

The "Volume serial number is" line shows you the serial number that MS-DOS has assigned to the disk. This number will not change unless the disk is formatted again.

On the last line of the display, MS-DOS asks you if you want to format another disk. Enter *y* to format another disk in the same drive with the same options. Enter *n* to return to the command prompt.

Formatting Different Disk Types

In Brief

To format a floppy disk that has a lower storage capacity than your disk drive, use the */f:* switch. For example, if drive A is a high density disk drive, the following command allows it to format a 360K disk:

```
format /f:360
```

There are a number of different kinds of floppy disks, each of which MS-DOS must format differently. Unless you state otherwise, MS-DOS assumes that the disk you want to format has the maximum capacity for the drive. For example, if you have a high-density (1.2 MB) 5.25-inch disk drive, MS-DOS assumes that the disks you format in the drive are also high density. To tell MS-DOS that you want to format a disk with a lower capacity, you use the */f:* switch.

For example, if drive A is a 1.2 MB, 5.25-inch drive and you want to format a 360K disk in it, you use the following command:

```
format a: /f:360
```

If drive B is a 1.44 MB, 3.5-inch drive and you want to format a 720K disk in it, you use this command:

```
format b: /f:720
```

NOTE Because there are differences in hardware between disk drives, some 360K drives cannot reliably read disks formatted on a 1.2 MB drive with the /f:360 switch.

Creating System Disks

In Brief

To create a system disk that can start your system, use the `format` command or the `sys` command.

To create the system disk during the format operation, add the /s switch to the `format` command. For example, the following command copies the system files to the disk in drive A after it is formatted:

```
format a: /s
```

To create a system disk from a disk that is already formatted, use the `sys` command. For example, the following command makes the disk in drive B a system disk:

```
sys b:
```

System disks contain the MS-DOS program. They are the kind of disks you can use to start your system.

System disks are just like other disks except that they contain the three main MS-DOS system files: `IO.SYS`, `MSDOS.SYS`, and `COMMAND.COM`. When you switch on your system, these three files are moved from the system disk to your system's random access memory (RAM). The `IO.SYS` and `MSDOS.SYS` files are hidden files; you do not see them in directories. The `COMMAND.COM` file is usually in the root directory of every system disk. The other command and system files that MS-DOS uses need not be present on a system disk. If you have a hard disk, it is usually set up to be your main system disk. However, when you switch on the system, it always checks drive A first to see if you want to start the system from a system disk there. That is why you see an error message if there is a non-system disk in drive A when you switch on your system.

You can copy the three system files to a disk as part of the `format` command or with the `sys` command.

To make a disk a system disk when you format it, you include the /s switch in the `format` command. After the disk has been formatted, MS-DOS copies the three

files to the disk. For example, the following command formats the disk in drive B and makes it a system disk:

```
format b: /s
```

To make a disk a system disk after it has been formatted, you use the `sys` command. For example, the following command makes the disk in drive A a system disk:

```
sys a:
```

After you run this command, the disk has two of the three files you need. Before you start MS-DOS from the disk, you must copy the `COMMAND.COM` file to the disk.

When a floppy disk has the three system files on it, you can use it to start your system by putting it in drive A and switching on the system. When your hard disk has the system files on it, you can use it to start the system by making sure no other disk is latched into drive A when you switch on your system.

Unformatting a Hard Disk

In Brief

To recover as much information as possible from a hard disk that has been reformatted, use the `unformat` command as in the following example:

```
unformat c:
```

The `unformat` command will not restore floppy disks.

After you first set up your hard disk, it should rarely, if ever, need to be formatted. If you accidentally format your hard disk, you may be able to recover all of the information. If you run the `unformat` command before you store any files on the disk, MS-DOS will restore the disk to its condition before the format. If you have stored any files on the disk since you formatted it, MS-DOS will not be able to restore the entire disk. The amount of the disk MS-DOS can restore depends on where the new file is stored on the disk.

Unless you tell it otherwise, MS-DOS performs a "safe" format on your hard disk. The safe format allows you to use the `unformat` command. To disable safe formatting you include the `/u` switch with your `format` command.

To unformat the hard disk in drive C, use the following command:

```
unformat c:
```

If MS-DOS cannot restore all the information on the disk, it displays a message asking you if you want to continue.

Using the Shell to Format Floppy Disks

When you format a floppy disk from the shell you are running the MS-DOS **format** command from the Program Manager screen. Therefore you use the same switches and some of the same procedures to format a disk from the shell as you would from the command line.

Switches allow you to format a floppy disk with a smaller storage capacity than your floppy disk drive and transfer the MS-DOS operating system to a disk. For more information about the switches you can use with the **format** command see the previous xx sections.

Shell ► To format a floppy disk:

1. Choose Format from the DOS Utilities group on the Program Manager screen. MS-DOS displays the Format Utility dialog box.
2. To format drive A and use no switches, Choose OK. To format a different drive or change the way the command functions, type a new drive letter and switches in the xx box and then choose OK.

From this point on, MS-DOS displays the same messages and prompts in the Shell as it does at the command prompt. For more information about messages MS-DOS displays, see xx.

3. After the formatting is complete, to give the disk a volume label, type a name at the volume label prompt. If you don't want to give the disk a label, press ENTER at the prompt.
4. At the next prompt, enter y if you want to format another disk or enter n if you want to return to the MS-DOS Shell.

MS-DOS returns you to the Program Manager screen when you press any key.

Labeling Disks

Each disk you use, including a hard disk, can have a name and a number. The name is called the volume label. You can use it to identify the disk electronically. The number is called the volume serial number. MS-DOS uses the volume serial number to keep track of which disk is in a drive. MS-DOS assigns a serial number to the disk when you format it. The serial number will not change unless the disk

is formatted again. Only disks formatted by MS-DOS versions later than 4.0 have a serial number.

MS-DOS displays the disk's volume label and serial number at the top of every directory.

You can change a disk's volume label with the `label` command. The volume labels you choose must be less than 12 characters long and cannot include the following characters: asterisk (*), question mark (?), slash (/), backslash (\), pipe (|), period (.), comma (,), colon (:), semicolon (;), plus sign (+), equal sign (=), double quotation mark ("), square brackets ([]) ampersand (&), and any control characters. Volume labels can include spaces but not tabs.

Assigning and Deleting Labels

In Brief

To assign a volume label to a disk, use the `label` command. For example, the following command gives the disk in drive A the label "backup disk 1":

```
label a:disk 1
```

The label can be no longer than 12 characters including spaces. There should be no spaces between the drive name and the label. If you enter the `label` command without parameters, MS-DOS prompts you for the drive and label you want.

If you work with a large number of disks, you may find it convenient to create a label for the disk that you can see when you use the `dir` or `vol` commands. This electronic label is called a volume label.

To assign a volume label, you use the `label` command. For example, the following command assigns the label "BERNIE'S 2" to the disk in drive A:

```
label a:bernie's 2
```

There must be no spaces between the colon of the drive letter and the label. If you type a drive letter, but no label, MS-DOS prompts you for a label. For example, to label the disk in drive B, you would enter the following command:

```
label b:
```

MS-DOS displays the current label and serial number for the disk in drive B and then prompts you to enter a new volume label.

To delete a volume label, enter the `label` command without a label name. Then, when MS-DOS prompts you to enter a new volume label, press ENTER. If you are giving a new volume label to a hard disk, MS-DOS asks you to confirm that you

want to delete the volume label. Enter **y** to delete the label or **n** to exit from the command without changing the current label.

Viewing Labels

In Brief

To display the volume label and serial number of a disk, use the **vol** command. The following command displays this information for the disk in the current drive:

```
vol
```

To view a disk's volume label and serial number, you use the **dir** command or the **vol** command. These commands can help you identify the disk you are using. With the **dir** command, the volume label and serial number for the disk that you specify are displayed above the list of files.

The **vol** command displays the volume label and serial number of the disk in the drive you specify. For example, the following command displays the volume label and serial number of the disk in drive A:

```
vol a:
```

Making Backup Disks

Because both floppy and hard disks can fail occasionally, and just in case you need a file you deleted, it is wise to back up your files on a regular basis. MS-DOS gives you many ways to make backup copies of files. If you want to make a backup copy of a few files, the simplest way is to use the **copy** or **xcopy** command to save the files on other disks. If you have a lot of files to back up, or if you want to automate the process of backing up files, you can use the **backup** command to:

- Back up single directories.
- Back up directories and their subdirectories.
- Back up selected files.
- Add files to a backup disk that you previously created.

Because the **backup** command uses whole disks, be sure you have enough disks to hold all the files you want to back up. MS-DOS deletes the existing files on the disks you use, so choose a disk for backups that you don't need for any other purpose.

You cannot directly access the files that MS-DOS creates on backup disks. If you need to retrieve the files you backed up, you must use the **restore** command. The **restore** command reads the backup disk and puts the files you specify back where they came from. The **restore** command is described in "xx".

Backing Up Single Directories

In Brief

To create and maintain backup disks, use the **backup** command. For example, the following command makes a backup of the files in the C:\WIN directory on drive A:

```
backup c:\win a:
```

The simplest form of the **backup** command backs up single directories. For example, the following command backs up the files in C:\WORD\HOME to the disk in drive B:

```
backup c:\word\home b:
```

MS-DOS displays the following prompt to tell you to insert your backup disk and to remind you that it erases the existing files before creating the backup files.

```
Insert backup disk 01 in drive B:  
WARNING! Files in the target drive  
B:\ root directory will be erased  
Press any key to continue . . .
```

When you press a key, MS-DOS begins copying files from C:\WORD\HOME. Files in subdirectories of this directory are not copied. To cancel the command without erasing any files from the target disk, press CTRL+C.

MS-DOS creates two files on drive B: BACKUP.001 and CONTROL.001. It combines all the files in C:\WORD\HOME and stores them in BACKUP.001. It stores the pathnames of the files in CONTROL.001. MS-DOS also changes the volume label of the disk to BACKUP 001.

If MS-DOS needs more than one disk to back up your files, it prompts you to insert another disk in drive B. The second disk has the files BACKUP.002 and CONTROL.002 (if there is a third disk the files are BACKUP.003 and CONTROL.003, and so on). Usually, if you don't specify a drive, MS-DOS assumes you want to use the current drive. However, with the **backup** command, you must always type at least a drive letter for both the source and destination directories. You do not have to type the pathname of the current directory. For example, if C:\WORD\HOME is the current directory, the following command is equivalent to the preceding example:

backup c: b:

Backing Up a Directory and its Subdirectories

In Brief

To back up a directory and its subdirectories, include the /s switch with the backup command. For example, the following command backs up every file in every directory on drive C:

```
backup c:\ a: /s
```

You can save time by backing up a directory and all its subdirectories with one command. To include subdirectories in your backup, use the /s switch. For example, to back up C:\WORD\HOME and all of its subdirectories, enter the following command:

```
backup c:\word\home b: /s
```

MS-DOS copies the files from C:\WORD\HOME and all of its subdirectories into the BACKUP.001 file on drive B. The directory structure of the files is preserved in the CONTROL.001 file.

To back up all the files on drive C enter this command:

```
backup c:\ b: /s
```

MS-DOS starts at the root directory and backs up all the files on drive C. MS-DOS prompts you to insert new disks as it needs them.

Backing Up Selected Files

In Brief

To back up selected files from a directory, use the MS-DOS wildcards as in the following command:

```
backup c:\win\*.xls a:
```

It is not always necessary to back up all the files in a directory. Sometimes you need to back up only files of a certain type, or files that have changed since the last backup.

To back up a single file, specify its filename after its drive letter and path. For example, the following command backs up the OUTGO.XLS file in the C:\WORD\HOME\directory:

```
backup c:\word\home\outgo.xls b:
```

To back up files of a certain type, you can use the MS-DOS wildcards. For example, the following command backs up only the files in C:\WORD\HOME that have a .DOC extension:

```
backup c:\word\home\*.doc b:
```

Adding Files to a Backup Disk

In Brief

To copy files to a backup disk without first deleting the files that are already there, use the /a switch with the backup command as in the following example:

```
backup c:\win a: /a
```

To add only files that have later versions than currently backed up file, or that don't already exist on the backup disk, include the /a and /m switches as in this command:

```
backup c:\win a: /a /m
```

In the preceding section, the backup command erased all existing files on the backup disk. To copy files to a backup disk without erasing the ones that are already there, use the /a switch. For example, the following command adds the files from C:\WORD\SCHOOL to those already on the disk in drive A:

```
backup c:\word\school a: /a
```

If some of the files from the WORD\SCHOOL directory are already on the backup disk, MS-DOS overwrites them with the files from drive C. When the operation is complete, the backup disk has the files it originally had plus the most current version of the files in C:\WORD\SCHOOL.

If you want to back up only files that have been added or changed since the last time you backed up a directory, use the /m switch with the /a switch.

For example, suppose that after you backed up the C:\WORD\SCHOOL directory, you added three new files and changed two that were backed up. To back up the files that are new or have changed, you would put the original backup disk in drive A and enter the following command:

```
backup c:\word\school a: /a /m
```

MS-DOS adds the new files to the backup disk and replaces the ones that have changed with the newer version.

NOTE If you use the /m switch without the /a switch, MS-DOS erases the existing files on the backup disk and copies only those that have changed since the last backup.

Using the Shell to Make Backup Disks

When you back up files from the Shell you are running the MS-DOS backup command from the Program Manager screen. Therefore, you use the same switches and wild cards and some of the same procedures to back up files from the Shell as you would from the command line.

Switches and wildcards allow you to back up subdirectories and selected files, and to add files to an existing backup disk. For more information about the switches and wildcards you can use with the backup command see the previous xx sections.

You can use the Backup Fixed Disk and Restore Fixed Disk commands only if you have the Shell installed on a hard disk.

Shell ► To backup files:

1. Choose Backup from the DOS Utilities group on the Program Manager screen. MS-DOS displays the Backup Utility dialog box.
2. To back up your entire hard disk, choose OK. To back up a subdirectory or selected files, or to add files to an existing disk, type the appropriate switches and wildcards in the Parameters box and then choose OK.

From this point on, MS-DOS displays the same messages and prompts in the Shell as it does at the command prompt. For more information about messages MS-DOS displays, see xx.

3. When the backup is finished, press any key to return to Program Manager.
To restore the files you backed up, you can use the Restore Fixed Disk command described in the following section.

Restoring Directories and Files

If you lose files you have backed up with the backup command, you can retrieve the copies stored on the backup disk with the restore command. If you use the backup command to make backup disks for your files, you must use the restore command to retrieve them. For information about the backup command, see the previous section.

Once you have created a backup disk, you can use the restore command to:

- Restore all the files on the disk to a specific directory or to a directory and its subdirectories.

- Restore selected files.

CAUTION Unless you use the /p switch, the restore command will overwrite files that changed since they were backed up. For information on the /p switch, see "xx" later in this section.

Restoring Files to a Directory

In Brief

To restore files backed up with the backup command, use the restore command. For example, the following command restores the files on the backup disk in drive B to their original locations in the root directory of drive C:

```
restore b: c:\*.*
```

The restore command requires a source directory, but it does not require a destination directory. The first parameter tells MS-DOS where to get the files from, and the second parameter tells MS-DOS which files to restore. For example, to restore all the files from the backup disk in drive B to the C:\WORD\HOME directory you would enter the following command:

```
restore b: c:\word\home\*.*
```

B: is the backup disk and C:\WORD\HOME*.* specifies the files you want to restore—every file on the backup disk that came from the C:\WORD\HOME directory.

When MS-DOS restores the files, it puts them in the directory they came from. If the directory they came from no longer exists, MS-DOS creates it.

When you enter the preceding restore command, MS-DOS prompts you to insert the disk with the backup files in drive B. When you press a key, MS-DOS displays the date of the backup and starts copying from the BACKUP.001 file to the C:\WORD\HOME directory. As files are restored, MS-DOS lists them on your screen.

If the files you want to recover were backed up on more than one disk, MS-DOS prompts you to insert the other disks. If the files you want to restore are not on the disk you specified, MS-DOS displays a "File not found" message.

To tell MS-DOS to prompt you before it replaces a file that is read-only or has been changed since the last backup, use the /p switch. For example, if you enter the following command, MS-DOS asks you to confirm your choice if any of the

Restoring Directories and Files

files that are backed up on drive B are read-only or have been changed since they were backed up:

```
restore b: c:\word\home\*. * /p
```

NOTE The /p switch depends upon directory time stamps to determine which file is the most recent. Because time stamps are set according to your system clock, if you depend on the /p switch, make sure your clock is always accurate.

To restore files to a directory and its subdirectories, use the /s switch. For example, the following command restores files from the backup disk in drive B to the C:\WORD\HOME directory and all of its subdirectories:

```
restore b: c:\word\home\*. * /s
```

MS-DOS reads the CONTROL.001 file to find out which directory a file came from and then stores it there. Any directories that don't exist are created.

To make sure that every file on a backup disk is restored, use the /s switch and start restoring at the root directory. For example, the following command restores every file on the backup disk in drive A to its original location on drive C:

```
restore a: c:\*. * /s
```

Restoring Selected Files

In Brief

To restore selected files from a directory, use the MS-DOS wildcards as in the following command:

```
backup a: c:\win\*.xls
```

To see a prompt before MS-DOS replaces files that have changed since the last backup, use the /p switch as in the following command:

```
restore b: c:\*. * /p
```

You can restore a subset of the files that were backed up by typing a single filename or by using the MS-DOS wildcards. For example, to restore only the C:\WORD\HOME\OUTGO.XLS file from the backup disk in drive B, you could enter this command:

```
backup b: c:\word\home\outgo.xls
```

To restore only the files with a .DOC extension, you would enter this command:

```
restore b: c:\word\home\*.doc
```

Using the Shell to Restore Files

When you restore files from the Shell you are running the MS-DOS restore command from Program Manager. Therefore, you use the same switches and wildcards and some of the same procedures to restore files from the Shell as you would from the command line.

Switches and wildcards allow you to restore subdirectories or selected files. For more information about the switches and wildcards you can use with the restore command, see the previous xx sections.

You can use the Backup Fixed Disk and Restore Fixed Disk commands only if you have MS-DOS Shell installed on a hard disk.

Shell ► To restore files:

1. Choose Restore from the DOS Utilities group on the Program Manager screen. MS-DOS displays the Restore Utility dialog box.
2. To restore all of the files on the backup disk in drive A, choose OK. To restore selected files or change the drive that the backup disk is in, type the appropriate drive letter, switches and wildcards in the Parameters box and then choose OK.

From this point on, MS-DOS displays the same messages and prompts in the shell as it does at the command prompt. For more information about the information MS-DOS displays, see xx.

3. When the operation is finished, press any key to return the Program Manager screen.

Verifying All Save Operations

In Brief

To verify any data that MS-DOS writes to disk, use the following verify command:

```
verify on
```

To stop verifying the data use this command:

```
verify off
```

In general, MS-DOS reads and stores information on your disks with no problems. Occasionally, due to problems with the magnetic media of the disk, information is

Recovering Files from Bad Disks

not stored properly on your disk. To tell MS-DOS to check that the information is stored accurately on your disks, you can use the following `verify` command:

```
verify on
```

While `verify` is on, MS-DOS makes sure that none of the sectors it tries to save information in are bad. To turn `verify` off, use this command:

```
verify off
```

To see whether the command is currently on or off, enter this command:

```
verify
```

When `verify` is on, data is written to disks more slowly than when it is off.

Recovering Files from Bad Disks

If you find that MS-DOS or another program can no longer read a file or group of files, there may be one or more bad sectors on the disk where the files or their directories are stored. To recover the parts of the file or files that are not damaged, you can use the `recover` command.

Recovering Files

In Brief

To recover as much information as possible from a file or files that have bad sectors, use the `recover` command. For example, the following command attempts to recover the `COMB.TXT` file from drive A:

```
recover a:comb.txt
```

You cannot get back the part of a file that is stored in a bad sector, but you can recover the rest of the file with the `recover` command. For example, if you have found that part of the `GRAY.HIC` file on drive A is no longer readable by the program that created it, you could use the following command to try to recover some of the information in the file:

```
recover a:gray.hic
```

MS-DOS reads the file one sector at a time. If any of the sectors are damaged, MS-DOS removes them from the file. MS-DOS marks the bad sectors so that it knows not to store any more information there.

When the operation is complete, MS-DOS stores the recovered file in the root directory of the disk it came from. MS-DOS names the files it recovers sequen-

tially, beginning with FILE0001.REC (FILE0001, FILE0002, FILE0003, and so on).

NOTE Even if a file is successfully recovered, it might be unusable if the lost information is necessary to the proper use of the file.

Recovering Disks and Directories

In Brief

If the directory of a disk is unusable, you may be able to recover some of the information on the disk with the **recover** command. For example, the following command recovers files from drive A:

```
recover a:
```

If you find that a disk is unusable because of problems in its directory, you can use the **recover** command to recover as much of the information on the disk as possible. For example, to recover files from drive A, you would enter this command:

```
recover a:
```

All the files that MS-DOS recovers are stored in the root directory of the disk they came from.

CAUTION The root directory can hold only a limited number of entries. If you try to recover more files than the root directory can hold, some files will be lost. In general, you should only use the **recover** command when it is absolutely necessary.

Giving a Drive Letter to a Directory

In Brief

To give a drive letter to a directory, use the **subst** command. For example, the following command assigns the drive letter A to the C:\BAR directory:

```
subst a: c:\bar
```

To remove the letter, use the **/d** switch. For example, the following command removes the association between A and C:\BAR:

```
subst a: /d
```

Sometimes, the program you are using will not accept any drive letters other than A and B. In these situations, you can use the `subst` command to fool the program into thinking that it is getting files from drive A or B when in reality the files are coming from a directory on your hard disk.

For example, suppose you are using a communications program that only accepts files from drive A or drive B. To use the files in `C:\COMM` with this program, you could enter the following command before you start the program:

```
subst a: c:\comm
```

Then, when the program requests files from drive A, MS-DOS will look in `C:\COMM` instead.

The drive letter you specify in the `subst` command must not be higher than the letter specified in the `lastdrive` command in the `CONFIG.SYS` file. For more information about the `lastdrive` command, see the *MS-DOS 5.0 User's Reference*.

When you are finished using the program, remove the association between the drive and the directory with the `/d` switch:

```
subst a: /d
```

The following commands ignore any associations you make with the `subst` command: `backup`, `format`, `chkdsk`, `diskcomp`, `diskcopy`, `fdisk`, `label`, `recover`, `restore`, `sys`.

Partitioning Your Hard Disk

Each operating system has its own conventions for storing files on a hard disk. If you use only MS-DOS, then it is no problem if your entire hard disk is set up using the MS-DOS conventions. However, if you want to use your hard disk with another operating system in addition to MS-DOS, you have to *partition* your hard disk into MS-DOS sections and non-DOS sections.

If you use only MS-DOS, you can create a single MS-DOS partition that occupies your entire disk. This gives MS-DOS access to all of the disk's storage capacity, and in some cases enables MS-DOS to gain access to the disk more quickly. If you use only MS-DOS but you would like to separate the files you store on your hard disk, you can create a second MS-DOS partition. In this case, MS-DOS still has access to the entire hard disk. However, the files in the second partition appear to be on a different drive.

If you want to use your hard disk with another operating system (for example XENIX or OS/2) you tell MS-DOS to use only part of the disk for its files. Then,

you can use the other operating system to set up the remaining disk space for its files. You switch to another operating system by making its partition *active*.

Partitioning your disk is different from formatting it. When you partition a disk, you tell MS-DOS which sections of the disk it can use. When you format a disk, MS-DOS prepares an existing partition to receive files. After you partition your disk, you must still format each partition before it can be used.

You create one or more MS-DOS partitions on a hard disk by using the Fdisk program included with MS-DOS.

Using Fdisk from the MS-DOS Command Prompt

You can partition your hard disk by entering **fdisk** at the command prompt.

The Fdisk program displays a series of menus that allow you to:

- Display information about your current partitions.
- Create, change, or delete MS-DOS partitions.
- Make a partition active.

CAUTION If you use Fdisk to change the existing partitions on a hard disk, you lose the information contained in those partitions. Be sure you have copies of all the data in a partition before you use Fdisk to change the partition.

Using Fdisk from the Install Program

You normally start Fdisk from the MS-DOS command prompt. If you want to partition your hard disk when you install MS-DOS, you can start Fdisk from the installation program. During the installation program, MS-DOS checks to see if your disk is partitioned. If it is not partitioned, MS-DOS asks you how you want it partitioned.

If you choose to create a single MS-DOS partition, MS-DOS creates the partition and continues with the installation. If you choose to create more than one partition, MS-DOS starts Fdisk. You can then set up partitions, using the information in this section. When you are finished creating partitions with Fdisk, MS-DOS continues the installation process. For more information about installing MS-DOS, see "XX".

Understanding Hard Disk Partitions

You can create two MS-DOS partitions on a hard disk:

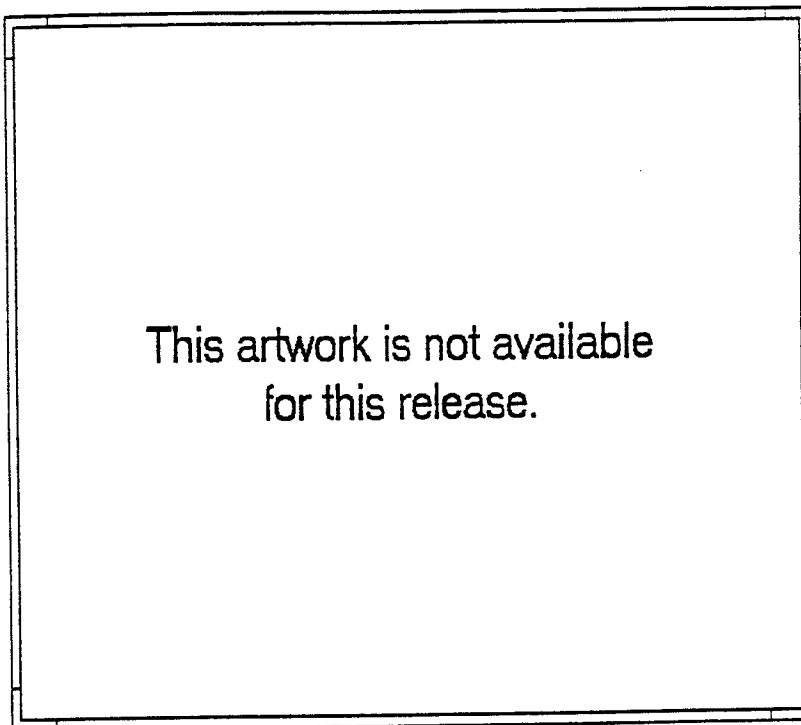
- The *primary MS-DOS partition* is the main storage area for MS-DOS. If you want to start MS-DOS using a hard disk, that disk must have a primary MS-DOS partition.
- An *extended MS-DOS partition* provides a way to further divide the files on a disk. You do not have to create an extended partition.

Whatever disk space you don't allocate to a primary and extended MS-DOS partition can be made into one or more non-DOS partitions by another operating system. The total number of partitions on a hard disk cannot exceed four. Thus, if you create only a primary MS-DOS partition and do not give it all the space on the disk, you can create three non-DOS partitions from another operating system. If you create a primary and extended MS-DOS partition, and you don't use all the disk's storage capacity, you can create two non-DOS partitions from another operating system.

The Primary MS-DOS Partition

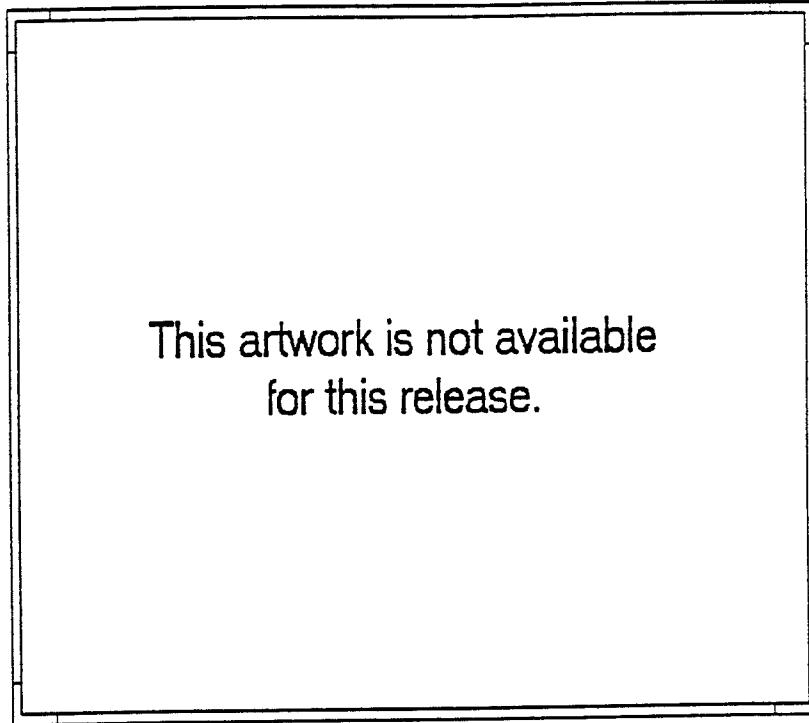
If you want to start MS-DOS using a hard disk, it must have a primary MS-DOS partition. If you want to start MS-DOS from your hard disk, the primary MS-DOS partition must contain the three MS-DOS system files. In general, the primary partition contains all the files that are on drive C. Most users require only this one partition. Your hard disk must have a primary MS-DOS partition to start MS-DOS.

You use Fdisk to set up a primary MS-DOS partition that takes up all the room on your hard disk, as shown in the following illustration:



dsk_3

If you reserve half of the disk's space for the primary MS-DOS partition, the other half will be available for other partitions, as shown in this illustration:



dsk_4

The Extended MS-DOS Partition

You can create a second MS-DOS partition on a hard disk, called an extended MS-DOS partition. An extended partition lets you treat a single hard drive as if it were two or more disk drives. When you create an extended partition, you divide it into one or more *logical drives*. A logical drive is a section of a hard disk that behaves like a separate disk drive. In other words, you can treat each logical drive in the extended MS-DOS partition as though it were its own disk drive.

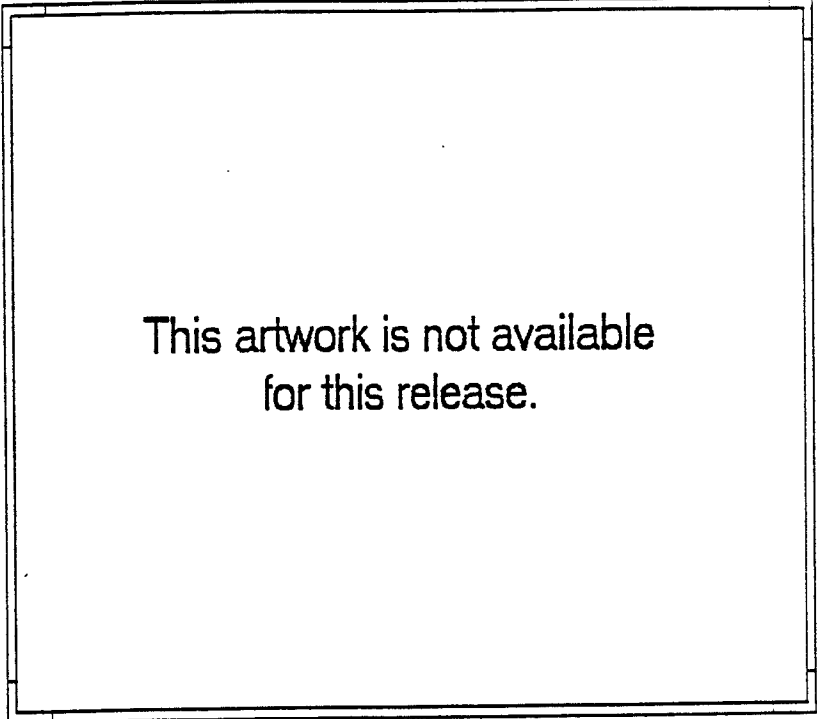
For example, suppose you create an extended MS-DOS partition that takes up 10 MB of hard disk space. If you create one logical drive in the extended MS-DOS partition, that part of the hard disk is named D. Your system would have a C drive that contains the files in the primary partition and a D drive that contains the files in the extended partition. You can change the current drive to D and work with

files and directories stored in those 10 MB of the hard disk as though they were on a separate disk drive.

If you divide the extended partition into two logical drives, your system would have a C drive with the files in the primary partition, a D drive with some of the disk space in the extended partition, and an E drive with the remaining space in the extended partition.

There are 26 letters available for drives (A through Z). A and B are reserved for floppy disk drives. C is usually reserved for the primary partition (unless you have additional floppy drives). That leaves a maximum of 23 logical drives that you can create within an extended partition.

Suppose you create a primary MS-DOS partition and an extended MS-DOS partition containing two logical drives. A diagram of your hard disk might look like this:



This artwork is not available
for this release.

dsk_5

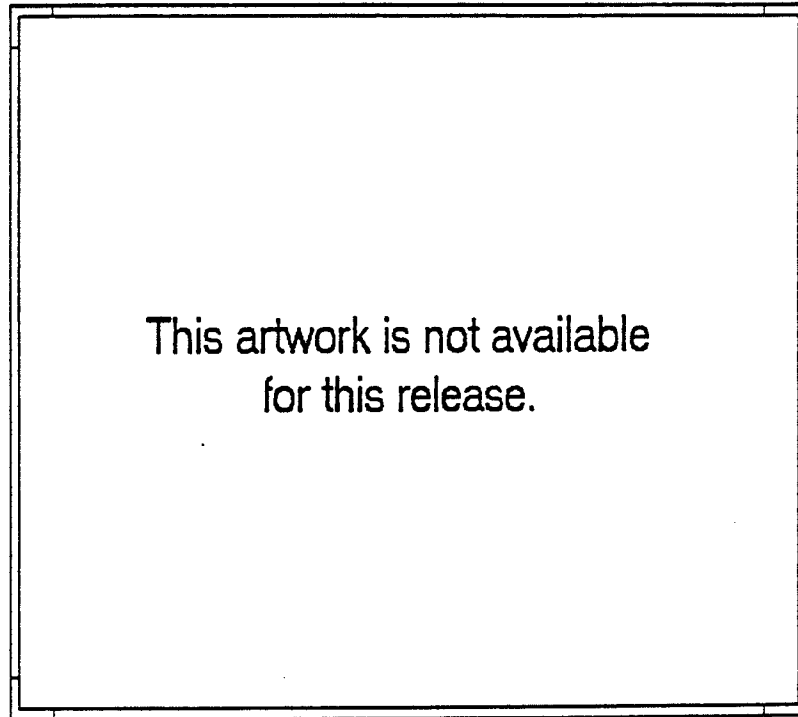
Logical drives do not give you more total disk space, but they do give you more flexibility in storing files.

Non-DOS Partitions

Non-DOS partitions are partitions for other operating systems (such as XENIX). You cannot use Fdisk to create a non-DOS partition. For information about creating non-DOS partitions, see the documentation for the system you want to use.

If you want to use non-DOS partitions, be sure you leave room on your hard disk. For example, if you want to use 40 percent of a disk for a non-DOS partition, you should create MS-DOS partitions that take up a total of 60 percent of the disk space. You can create two or three non-DOS partitions on a disk, depending on the number of MS-DOS partitions you have.

If you use half your disk for MS-DOS and the other half for another operating system, a diagram of your disk looks like this:



dsk_6

The Active Partition

To tell your computer to use the operating system stored in a partition, you make it active. To use MS-DOS, you make your primary MS-DOS partition active (because an extended MS-DOS partition does not contain the MS-DOS operating system it cannot be active). A hard disk can have only one active partition at a time.

If you have only a primary MS-DOS partition that occupies the entire disk, it is automatically the active partition. For more information about setting the active partition, see "XX" later in this chapter.

Using Fdisk

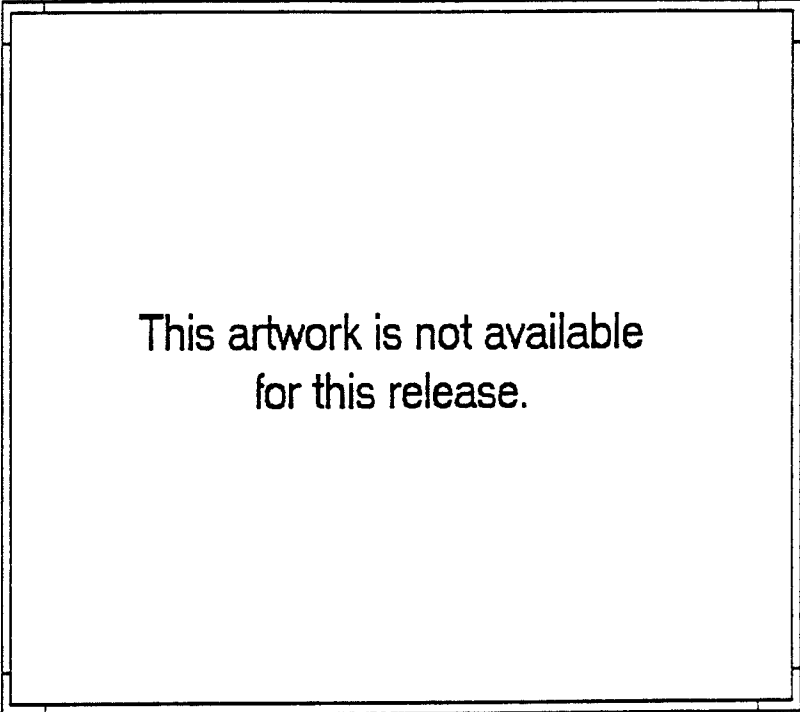
You use Fdisk to display partition information, create partitions and logical drives, set the active partition, and delete partitions and logical drives. This section introduces the basic information you need in order to use Fdisk.

CAUTION Fdisk destroys all existing files in any partitions you modify. If you are using Fdisk to change the partitions on a disk with files on it, be sure you have backup copies of all files you want to keep in the partitions to be modified before you begin. If you want to create smaller partitions on a hard disk that only has a large MS-DOS partition, you need to back up all files on the disk that you want to keep.

To start using Fdisk, enter the following command at the command prompt:

```
fdisk
```

The `fdisk` command has no parameters or switches. When Fdisk starts, it displays the main Fdisk menu:



This artwork is not available
for this release.

dsk_7

Fdisk uses a series of menus to display options, help messages, and warnings. To choose a menu option, type the number next to the option you want and press ENTER. To return to a previous menu, press ESC. To exit Fdisk, return to the main Fdisk menu and press ESC.

Each Fdisk menu displays a "Current fixed disk drive" message, followed by a number. The term *fixed disk* means *hard disk*. If you only have one hard disk drive, this number will always be 1. If you have more than one hard disk drive, the number will show which disk Fdisk is currently working with. The first hard disk drive in your system is 1, the second disk drive is 2, and so on. Changing the current drive in Fdisk does not change the current drive when you return to the system prompt. The Fdisk current drive refers only to physical disk drives, not logical drives.

Displaying Partition Data

You can display information about the status, type, and size of the partitions on a hard disk by choosing Display Partition Information (option 4) from the Fdisk main menu. The Display Partition Information screen looks like this :

```
Display Partition Information

Current hard disk drive: 1

Partition Status Type Size in Mbytes Percentage of Disk Used
C: 1   A  PRI DOS   21   50
% 2   EXT DOS   21   50

%Total disk space is 42 Mbytes (1 Mbyte = 1048576 bytes).

The Extended DOS partition contains
logical DOS drives. Do you want to
display logical drive information? [Y]

Press ESC to continue
```

The information varies, depending on the number, size, and type of partitions on your hard disk. The "Partition" column shows the drive letter of the disk Fdisk is currently working with, and the number of each partition. The "Status" column displays the letter A next to the active partition. The "Type" column shows whether a partition is a primary MS-DOS partition (PRI DOS), an extended MS-DOS partition (EXT DOS), or a non-DOS partition (non-DOS). The "Size in Megabytes" column shows the size of each partition, and the "Percentage of Disk Used" column shows the percentage of the current disk that each partition occupies.

If there is an extended MS-DOS partition that contains logical drives, Fdisk asks if you want to see information about that partition's logical drives. Enter y to display this information.

The screen looks like this :

```
Display Logical DOS Drive Information

Drv Volume Label Mbytes System Usage
D: BACKUPA    18  DOS   90
%E: BACKUPB    2  DOS   10

%Total Extended DOS Partition size is 20 Mbytes (1 Mbyte = 1048576
bytes)

Press ESC to continue
```

The information on the screen varies depending on the number and size of the logical drives. The "Drv" column displays the drive letter of each logical drive. The

"Volume Label" column shows the label assigned to each drive. "Mbytes" is the size of each logical drive and "Usage" shows the percentage of the available space in the extended MS-DOS partition that each logical drive takes up.

Creating a Primary MS-DOS Partition

The first hard disk you use with MS-DOS must have a primary MS-DOS partition. You can create a primary MS-DOS partition that reserves all of a hard disk's storage space or only a part of it. If you want to create an extended MS-DOS partition with logical disk drives, or if you want to leave room for a non-DOS partition, you need to create a primary MS-DOS partition that is smaller than the total size of your disk.

To change the size of an existing primary MS-DOS partition, delete it and recreate a new one that is the size you want. You will lose any information stored in the existing primary MS-DOS partition, so be sure you have backup copies of any files you want to save. See "XX" later in this chapter for information about deleting a partition.

If your hard disk already has a partition on it, you must delete it before you can create a primary MS-DOS partition that occupies the entire disk. If your hard disk does not already have a partition you can use the following procedure to create a primary MS-DOS partition that occupies the entire disk.

► To create a primary MS-DOS partition that occupies the entire the hard disk:

1. Start Fdisk and then press ENTER to choose Create MS-DOS Partition or Logical Drive (option 1) from the Fdisk main menu.
The Create DOS Partition menu appears.
2. Press ENTER to choose Create Primary DOS Partition (option 1).
Another menu appears, with this message:
"Do you wish to use the maximum size for a Primary DOS Partition
and make the partition active (Y/N).....? [Y]"
3. Enter y. If you enter n, Fdisk prompts you to create a smaller primary partition. See the following procedure for more information.
Fdisk creates a primary partition that occupies the entire hard disk. If you have only one hard disk MS-DOS displays the following message:
System will now restart

Insert DOS Install disk into drive A:
Press any key when ready

4. Insert an MS-DOS system disk and press any key. You still need to format your hard disk before you can use it. For more information about formatting, see "XX".

► **To create a primary MS-DOS partition that occupies part of the hard disk:**

1. Press ENTER to choose Create DOS Partition or Logical DOS Drive (option 1) from the Fdisk main menu.
The Create DOS Partition menu appears.
2. Press ENTER to choose Create Primary DOS Partition (option 1).
Another menu appears, with this message:
*Do you wish to use the maximum available size for a Primary DOS Partition
and make the partition active (Y/N).....? [Y]
3. Enter n.
A second Create Primary DOS Partition menu appears.
4. Press ENTER if you want the default size (100%). Otherwise, enter the number of megabytes or percentage of the disk to use. If you enter a percentage, follow the number with a % sign.
The following message appears:
Primary DOS Partition created, drive letters changed or added.
5. Press ESC to return to the Fdisk main menu.
When you leave Fdisk you will need to format the new partitions on your hard disk. For more information see "XX", later in this chapter.

Any part of the disk that you do not use for the primary MS-DOS partition can be used for an extended MS-DOS partition, or for non-DOS partitions.

NOTE When you create a primary MS-DOS partition that does not take up all the room on a hard disk, you must make the primary MS-DOS partition active before you can use the hard disk with MS-DOS. For information about making a MS-DOS partition active, see "XX" later in this chapter.

Creating an Extended MS-DOS Partition

To divide your hard disk into two MS-DOS storage areas, you create an extended MS-DOS partition. Within the extended MS-DOS partition you can create one or more logical drives. Logical drives are parts of your hard disk that MS-DOS treats

as separate disk drives. You can create only one extended MS-DOS partition on a hard disk, but you can create up to 23 logical drives in extended MS-DOS partitions on your hard disks.

If you have one hard disk, there must be a primary MS-DOS partition on the disk before you can create an extended MS-DOS partition. If you have more than one hard disk, the first disk in your system must have a primary MS-DOS partition, but you can create extended MS-DOS partitions on your additional disks without creating a primary partition on them.

► **To create an extended MS-DOS partition:**

1. Choose Create DOS Partition or Logical DOS drive (option 1) from the main Fdisk menu.
2. Choose Create Extended DOS Partition (option 2) from the Create DOS Partition menu.

Fdisk displays a menu that shows the total number of megabytes available for an extended partition. The default for the partition size is the maximum available space on the hard disk drive minus the size of the primary partition. If there is no space available, you must delete and recreate the primary MS-DOS partition so it is smaller, or reduce the size of any non-DOS partitions that exist.

3. Press ENTER to choose the default size. Otherwise, type the number of megabytes or the percentage of the disk to be used for the extended MS-DOS partition. If you enter a percentage, follow the number with a % sign.

The Create Logical Drive(s) in the Extended DOS Partition menu appears.

When you create the extended MS-DOS partition, you can set up one or more logical drives. See the following section for more information.

NOTE If Fdisk finds any defective tracks at the start of an extended MS-DOS partition, it adjusts the partition boundaries to avoid bad tracks.

Creating Logical Drives in the Extended MS-DOS Partition

To store information on the portion of the hard disk assigned to the extended MS-DOS partition, you need to create one or more logical drives. Logical drives are sections of the extended MS-DOS partition that MS-DOS treats as disk drives. You can use logical drives to organize the information on your hard disk. When you create logical drives in an extended MS-DOS partition, they are assigned drive letters. For example, if you create one logical drive on a hard disk with the drive

letter C, the logical drive is given the drive letter D. For more information about how drive letters are assigned to logical drives, see the next section, "XX".

You can store and retrieve information on your logical drives as though they were physical disk drives. For example, you can use logical drive D to store files for a particular application, and work with them by specifying drive D rather than a directory location.

► To create or modify logical drives:

1. Create an extended MS-DOS partition. See the preceding section for more information.
2. On the Create Logical Drive(s) menu, enter the number of megabytes or the percentage of the partition space for the first logical drive you want to create. If you enter a percentage, follow the number with a % sign. If you want one logical drive to take up the whole extended MS-DOS partition, press ENTER to accept the default (the entire disk).
3. Continue entering the sizes of partitions until you have used up the entire partition or until you have created all the logical drives you want.
If the entire partition is assigned to logical drives, Fdisk exits the menu automatically. To exit from the menu before all the space has been allocated, press ESC.

After you create logical drives, you must format them before MS-DOS can use them. For more information about formatting logical drives, see "XX", later in this chapter.

How Drive Letters Are Assigned

The primary partition of the first hard disk in your system is normally drive C. The drive letters of additional hard disks and logical drives depend on the number of disks and how they are partitioned.

NOTE If you have three or four floppy disk drives in your system, the primary partition of the first hard disk may be D or E. In the examples in this section, it is assumed that the primary partition of the first hard disk is C.

If you have only one hard disk, any logical drives you create in the extended MS-DOS partition are given letters beginning with D. For example, if you create five logical drives in the extended partition, they are named D, E, F, G, and H.