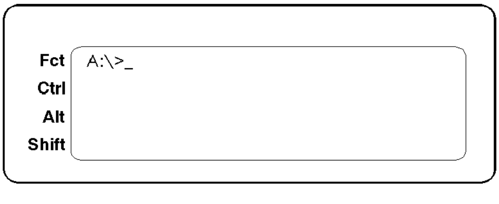
ROM-DOS™ 6.22 User Manual

```
if "%1"=="A" goto sample2
rem*******************************
rem If the user doesn't type A, run sample1
rem
cd \sample
sample1
rem Skip over sample2 by jumping to
rem  the line labeled :end.
goto end
rem
rem*******************************
rem If DOS jumps to this label, the user
rem  wants sample1
rem
:sample2
cd \sample
sample2
rem The following line marks the end of the
rem batch program.
:end
```



```
A:\>_
```

MONARCH® DOS-based PATHFINDER® Ultra® System

Monarch®

PAXAR

TC6035RD Rev. A 7/98

Each product and program carries a respective written warranty, the only warranty on which the customer can rely. Monarch reserves the right to make changes in the product, the programs, and their availability at any time and without notice. Although Monarch has made every effort to provide complete and accurate information in this manual, Monarch shall not be liable for any omissions or inaccuracies. Any update will be incorporated in a later edition of this manual.

WARNING

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

CANADIAN D.O.C. WARNING

This digital apparatus does not exceed the Class A limits for radio noise emissions from digital apparatus set out in the Radio Interference Regulations of the Canadian Department of Communications.

Le présent appareil numérique n’émet pas de bruits radioélectriques dépassant les limites applicables aux appareils numériques de la classe A prescrites dans le Réglement sur le brouillage radioélectrique édicte par le ministère des Communications du Canada.

Trademarks
Monarch, Pathfinder, and Ultra are registered trademarks of Monarch Marking Systems, Inc.
6035 is a trademark of Monarch Marking Systems, Inc.
Microsoft and MS-DOS are registered trademarks of Microsoft Corporation.
Datalight is a registered trademark Datalight, Inc.
ROM-DOS is a trademark of Datalight, Inc.

Monarch Marking Systems
P.O. Box 608
Dayton, Ohio 45401

For supplies, service, or assistance call toll free:

1-800-543-6650 (In the U.S.A.)
1-800-263-4650 (In Canada)

www.monarch-marking.com

ROM-DOS 6.22
User Manual

TABLE OF CONTENTS

AN INTRODUCTION TO ROM-DOS 6.22

1

Organization of This Manual

Following “Introduction To ROM-DOS™ 6.22” are the two primary sections of this manual: “Using ROM-DOS 6.22” and the “ROM-DOS 6.22 Command Descriptions.”

“Using ROM-DOS 6.22” provides a general introduction to basic DOS concepts including subjects such as file names and tree-structured subdirectories. This section will be particularly helpful for first-time DOS users.

“ROM-DOS 6.22 Command Descriptions” provides a complete stand-alone entry for each command available in ROM-DOS.

Conventions Used in This Manual

Character Notation

When necessary, individual characters or strings will be surrounded by parenthesis to set them apart from surrounding text. For example, you might see a double quote indicated like this (").

Special function keys are marked with left arrow and right arrow brackets like this: <Shift>.

Simultaneous Keys

Often, one key must be held down while another key is pressed. In such cases, the keys are shown side-by-side like this: <Ctrl><Break>. In this example, <Ctrl> is pressed first and held down while <Break> is pressed.

If three keys are shown together, the first two in the row are held down while the third is pressed. A common example of this is the reboot keystroke sequence <Ctrl><Alt><Del>. <Ctrl> and <Alt> are held down while <Del> is pressed.

Lines Which You Type

Examples of lines of type that you enter are shown indented and in a different typeface, like this:

```
DEL MYLETTER.DOC
```

Syntax descriptions and other examples in this manual display either literal terms or representative terms.

Literal Terms

Character strings to be typed in exactly as written are shown in all capital letters in the typeface shown here:

```
DEL
```

Representative Terms

Expressions which are not typed in literally but rather represent information you supply are indicated with italicized lower case letters. For example, where you see the word *filename*, you type in your own file specification information. So, if the instruction says:

DEL *filename*

you might actually type (where MYLETTER.DOC is the name of the file to be deleted):

```
DEL MYLETTER.DOC
```

All such italicized terms are fully explained in this manual.

Command Line Options

Anything shown on a command line between left and right square brackets ([ ]) is an optional entry. For example, in this line:

PAUSE [*message*]

the *message* portion may be left off and the command is still complete.

Sometimes there may be a limited set of mutually-exclusive options from which to choose. In such cases, the possible choices are given, separated by a vertical bar (|). For example:

```
BREAK [ON|OFF]
```

In this case, if you elect to include anything more than the word BREAK, it must be either the word ON or the word OFF.

<Enter> Keystroke Assumed

Although not noted, one keystroke is assumed at the end of each command described. <Enter> must be pressed to tell ROM-DOS to accept your line of input. Any exception to this rule is made clear in this documentation.

To illustrate, rather than showing the line:

```
DIR <Enter>
```

we simply list the command:

```
DIR
```

with the understanding that you type DIR and then press <Enter> to execute the command.

USING ROM-DOS 6.22

2

About DOS and ROM-DOS

What is DOS?

DOS is an acronym for Disk Operating System. Several brands of DOS have been created by various companies. Each brand of DOS (PC-DOS, MS-DOS, or ROM-DOS) is a set of commands or code which tells the computer how to process information.

DOS runs programs, manages files, controls information processing, directs input and output, and performs many other related functions.

What is ROM-DOS?

ROM-DOS is an operating system which can be embedded in ROM (Read Only Memory) and can run entirely from within ROM. ROM-DOS is functionally equivalent to other brands of DOS and can run programs executable under a standard DOS (which executes out of RAM). With ROM-DOS, the executable program may reside in RAM or may be placed in ROM along with ROM-DOS.

The topics in this manual may or may not apply to your printer, depending on the options it has.

Files

What is a File?

A file is a defined set of related information electronically stored for use by your computer. Examples of files include a word-processed letter, a database full of accounting information, or a computer program. ROM-DOS itself is actually a collection of files.

A file may be stored on a diskette (also called a floppy), on a hard drive, or in computer memory (RAM or ROM).

Naming Your Files

To control interaction between various computer files, each file must have its own name. A file's name gives it an identity which is recognizable by both you and the computer.

The File Name

Computer file names in the DOS environment consist of two parts similar to a first name and last name.

The first part can be thought of as the primary file name:

name

You can give a file any name you want. The name can range from 1 to 8 characters in length and consist of any combination of letters, numbers, and the following symbols: the underscore(_), the caret (^), the dollar sign ($), the tilde (~), the exclamation point (!), the number sign (#), the percent sign (%), the ampersand (&), the hyphen (-), the braces ({ }), parenthesis (( )), the at sign (@), and the apostrophe (‘).

The Extension

The second part of the file name is called the extension:

ext

The extension can be from 1 to 3 characters long, with the same character limitations as the *name.* An extension is not required.

When used together, the *name* and *ext* are separated by a period:

name.ext

This manual uses *filename* to refer to the *name* plus the *ext* if there is one.

Extensions can be helpful in identifying the type of file. Commonly used file name extensions include DOC for documents, DAT for data, and TXT for text files.

You may use any extension you choose. However, certain extensions have a special meaning to ROM-DOS and should only be used when appropriate. These include .COM, .EXE, and .BAT. .EXE is used for executable files. The .COM extension is for command files. The .BAT extension is used for DOS instruction batch files.

Some application programs, such as word processors, may use or require particular file extensions for output or input files. It is best to follow the application instructions regarding proper file name extensions for that particular program.

Examples

A file could be named simply:

```
LETTER1
```

The same file name could have an extension included:

```
LETTER1.DOC
```

or

```
LETTER.XYZ
```

An executable program might have the file name:

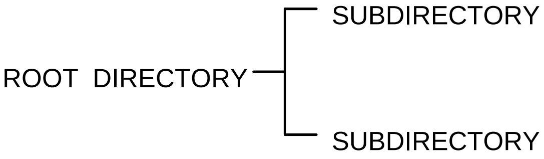
```
RUNME.EXE
```

It is possible to have several files with the same file name but different extensions. ROM-DOS searches for and accesses the file extensions in order of .COM first, .EXE second, and .BAT third, then all others. For example, you could have an executable file called MYPROG.EXE and a batch file call MYPROG.BAT. If you typed MYPROG at the command line, the file MYPROG.EXE would be executed. If you wanted to execute the batch file MYPROG.BAT, you would have to specify the .BAT extension on the command line.

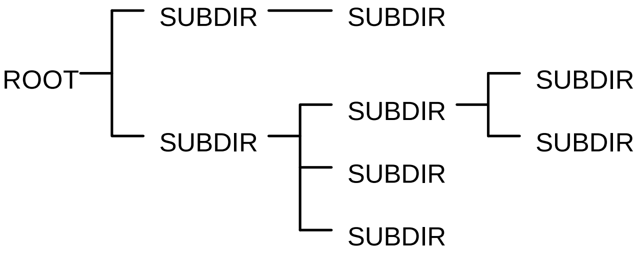
Tree-Structured Directory System

DOS stores files in what is called a tree-structured filing system. The different **places** where files are stored can be thought of as branches on a tree. Each branch, actually called a subdirectory, is either attached to the **root** directory or is attached to another branch.

This concept is often represented with a sideways tree when shown in graphic form. Here is a simple two-branch, **tree-structured** system of subdirectories:



The system may be as simple or complicated as necessary. Here is a more complex system (ROOT and SUBDIR are used to save space).



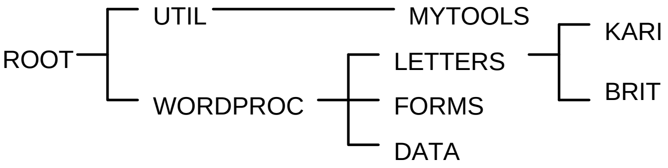
Each subdirectory has only one higher-level directory from which it stems. This single directory is called its **parent**.

Also note that each one of the subdirectories within this system can contain files. A subdirectory containing no files is **empty**.

Naming Subdirectories

You create subdirectories. You can create any structure you choose (within the tree system) giving each a subdirectory name you want. The naming of subdirectories is similar to the naming of files. There is an 8-character limit with the same character choice limitations as for file names (letters, numbers, and symbols). A subdirectory name can also have an extension. Commonly the extension is not used so that subdirectory names look different from file names.

Here is an example of the same directory structure used in the previous section. This time some actual subdirectory names are used:



Subdirectories are created and deleted with the MKDIR and RMDIR commands. For more information on creating and deleting subdirectories, see the MKDIR and RMDIR in Chapter 3.

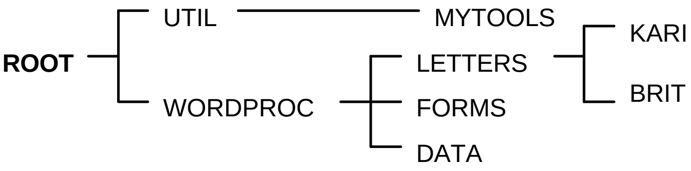
Moving Around the Tree

At any one time, ROM-DOS works from one particular point in the tree-structured filing system. This one particular point or **place** is called the default directory. When the computer is first powered on, operations begin in the root directory. Thereafter, you can change the default directory to be any subdirectory on your system. It is also possible to have the computer automatically move to a different directory upon system startup by adding commands to your AUTOEXEC.BAT file.

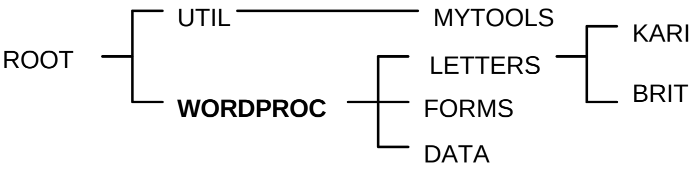
You move between directories (changing your default directory with each move) with the CHDIR (Change Directory) command. See Chapter 3. An explanation of CHDIR and its use can be found in the command description section of this manual. For now, we will refer to changing directories without explaining specific command usage.

Example

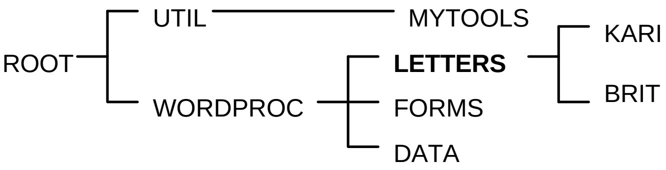
Let's use a piece of the example from the previous section. Boldface type marks the ROOT directory as our default directory upon starting up the system.



From the ROOT, we can move, or change directories, to the subdirectory called WORDPROC. WORDPROC is now the default directory.



Another move could take us to LETTERS.



By using the CHDIR command, we can move to any directory, without having to step through neighboring directories. We could have gone straight to the LETTERS subdirectory from the ROOT by specifying complete directions for reaching the desired directory.

The Path

Path refers to the set of subdirectories between the root and the default directory. In the tree analogy this is a sequence of the names of the branches leading to the branch you are currently on. The names in the path are separated by a backslash (\).

In the preceding example, the path we ended with was called \WORDPROC\LETTERS. The word **ROOT** doesn't appear in the path. It is unnecessary to include it since all paths must begin with the root. The root is represented by the first backslash (\) in the path.

By using the path description, we know where the default directory sits in relation to its parent, and its parent . . . and so on back to the root.

The Drive Specification

Since ROM-DOS can store and retrieve information from more than one disk drive, you should understand how drives are identified.

Disk drives are given letter names. By convention, floppy disk drives are identified as drive A and drive B. Even on systems which have only one physical floppy drive, ROM-DOS can treat the one drive as either A or B.

The hard drive, if your system has one, is identified as drive C. It can be partitioned, or divided, into smaller sections. Disks exceeding 4 gigabytes in sizes must be partitioned into two or more areas, with a maximum size of 4 gigabytes per partition. Each partition of a drive is identified by a separate drive letter. The first partition is drive C, the next drive D, and so on. The highest available drive is Z.

Standard notation for the drive name in this manual is an italicized small d and colon: *d:*

To refer to the C drive, you would type: c:

The identifier letter may be entered in upper or lower case.

Complete File Identification

The full identity of each computer file includes

- the name of the drive on which the file is stored.
- the path of subdirectories to the location of the file in the tree-structure system.
- the file name, including extension, if any.

In the standard notation of this manual, the full file specification may be indicated as follows:

d:\path\filename.ext

Note that just as the names of subdirectories are separated by a backslash (\) in the path specification, a backslash is also used between the drive name and path and between the path and the file name.

To simplify notation, the full file specification may simply be called

filespec

Using Wildcard Characters

Wildcard characters can be used to reference groups of files without typing the complete file name for each file. A wildcard is a substitute for all or part of a file name or extension. Performing a task for a large group of files can be simplified by the use of wildcards.

The two wildcard characters are the asterisk (*) and the question mark(?). The asterisk represents an entire name or a group of characters found within a name. The question mark represents a single character.

Examples

DIR outputs a list of all of the files in the \TEST directory with the extension .EXE:

```
DIR c:\TEST\*.exe
```

Searches for a file and only knows that it starts with a particular letter, for example D:

```
DIR D*.*
```

This displays all of the files in the current default directory that start with the letter D and have any extension.

Copies an entire group of files so that with one simple command you had reserve copies:

```
COPY c:\*.BAT b:\*.BAK
```

This copies all of the files with a .BAT extension from the c: drive root directory onto the b: drive. The files on the b: drive all have a new extension of .BAK.

The question mark only substitutes for a single character at a time. To locate all of the files that had four character file:

```
DIR b:\????.*
```

DIR displays a list of all of the files on the b: drive that have exactly four characters in the file name and any extension.

The question mark can also be used to match a single specific character in a file name:

```
REN TEST?.BAT TEST?.OLD
```

This command renames all files that have **TEST** as the first four characters in the file name, followed by any single character and the .BAT extension. The files retain the same TEST? file name, but have a new extension, .OLD.

The System Prompt

After execution of each operation, ROM-DOS waits for your next instruction. In this state, you will see the **system prompt** on your monitor.

By using the PROMPT command you can specify the information to be displayed in the prompt. Until you define the prompt otherwise, it includes only the letter name of the default disk drive and a right arrow (>). For example:

```
A>
```

One common choice for prompt line configuration is to include the default path in addition to the drive name and right arrow (>). After customization, your prompt looks like this:

```
a:\UTIL>
```

For more information, see PROMPT in Chapter 3.

The Command Line

Your keystrokes begin just to the right of the system prompt. The line of instruction you type at the prompt is called the **command line**.

Editing the Command Line

The characters of the last command entered on the command line are stored in a command line **buffer** and can be edited for re-entry.

Here are the keys you can use to edit the command line:

<F1> Displays one character at a time from the command line buffer. The right-pointing direction key works in the same way.

<F3> Displays entire contents of command buffer

<Ins> Allows insertion of one or more characters in the command line.

<Del> Allows deletion of a character from the command line buffer.

<Esc> Cancels the current command line and returns you to a new, empty line.

<Backspace> Deletes to the left. Allows "backing up" on the command line. The left-pointing direction key works in the same way.

Examples of Command Line Editing

Suppose you intended to enter the CHKDSK command, but accidentally entered:

```
CHKDSI
```

You receive a message indicating that CHKDSI is a nonexistent command or file name. Rather than re-keying in the entire string, you could press <F3> which re-displays the entire string you just entered:

```
CHKDSI_
```

At this point your cursor is located just after the last character displayed, indicated here with the underscore (_). By pressing <Backspace> you remove the letter (I) from the end of the line, putting the cursor in its place:

```
CHKDS_
```

At this point you can type the correct character:

```
CHKDSK_
```

and press <Enter> to execute the command.

As another example, suppose you enter a long string:

```
COP DATA1.DAT a:DATA1BAK.DAT
```

The command does not work. You realize that the first word, COPY, is missing a letter. To correct the command line, press <F1> three times to display the first three characters you just entered:

```
COP_
```

Now press <Ins> followed by a letter (Y):

```
COPY_
```

You can now fill in the rest of the command by simply pressing <F3> once and you will see this:

```
COPY DATA1.DAT a:DATA1BAK.DAT_
```

which is the correct command, ready to be executed with <Enter>.

Redirecting Input and Output

There are certain conventions indicating where each ROM-DOS function receives input and sends output. However, the input and output can also be redirected. This is accomplished with the right arrow (>) and the left arrow (<).

Input Redirection

The syntax for changing standard input source from keyboard input to file input is:

<*filespec*

When this is added at the end of the command line, ROM-DOS begins to receive its instructions from the named file.

CAUTION

If input redirection is used and the input file is incomplete, the system hangs waiting for instructions from the input file and does not accept keystroke information from the keyboard (except for <Ctrl><Alt><Del> to reboot).

Output Redirection

The syntax for redirecting output to a file is:

>*filespec*

When this is added to the end of the command line, standard output is temporarily directed to the file named in the *filespec*. If the named file already exists, its contents are replaced with the ROM-DOS function's output. Otherwise, a new file is created to hold the output. Output can also be redirected to a device such as PRN (the printer).

To append output to the receiving file, rather than replace its contents, the double right arrow (>>) is used:

>>*filespec*

This adds the ROM-DOS function's output after the existing contents of the named file. If the named file does not exist, ROM-DOS creates the file.

Example

Suppose you wanted to save a list of the current directory into a file. Normally, the command:

```
DIR
```

displays the directory list on your monitor.

You can, however, redirect this command's output to a file by entering:

```
DIR > MYDIR.TXT
```

This redirects the output of the DIR command into your file called MYDIR.TXT. The redirection only holds for this one occurrence; subsequent running of DIR returns to normal on-screen listing.

Batch Files

A batch file is a standard text file containing a list of commands which can be submitted to ROM-DOS for automatic sequential execution. Use of batch files helps you avoid unnecessary retyping of command sequences which are commonly repeated, complex, or difficult to remember.

The ROM-DOS command processor provides full batch file processing, compatible with standard DOS version 6.22. Batch files can include internal DOS commands, external DOS commands, special batch subcommands, names of other executable files or programs, or even the names of other batch files to which control will be transferred.

Batch File Names

When naming batch files, use the .BAT extension on the file name. This extension tells ROM-DOS to execute the batch file when its name is entered from the command line. The name of the batch file cannot be the name of other internal commands. For example, COPY.BAT is an invalid batch file name.

To execute the batch file, type the file name on the command line. You need not include the .BAT extension. When you press <Enter>, batch file execution begins.

Creating a Batch File

A batch file can be created using any word processor or text editor that will save output as unformatted text (pure ASCII). Batch files can also be created by typing directly from the keyboard into a file with the command

```
COPY CON filename.BAT
```

This copies the output from the console (keyboard) to the specified file. Once you have entered the above command, you may begin to type in the contents of your batch file.

At the completion of each line, press <Enter>. As you type each line, you can make corrections by using <Backspace> and retyping. If an incorrect line has been entered, or if you wish to discontinue for any other reason, press <Ctrl><C> to return to the command line. Your work is not saved.

When you have finished entering all the lines in your batch file, press <Ctrl><Z> and then <Enter> to complete the creation of the file and return you to the command line prompt.

Batch File Parameters

A batch file may use parameters placed on the command line. These parameters are inserted as command line arguments for commands or instructions within the batch file prior to their execution.

A batch file called ARCHIVE could be set up as follows:

```
PRINT %1
COPY %1 \ARCHIVE\*.*
DEL %1
```

The batch file could be executed by entering:

```
ARCHIVE THISFILE.DAT
```

The parameter %1 takes on the name THISFILE.DAT and the batch file actually executes the following:

```
PRINT THISFILE.DAT
COPY THISFILE.DAT \ARCHIVE\*.*
DEL THISFILE.DAT
```

Special Batch Subcommands

In addition to standard ROM-DOS commands that run in batch files, there are also special batch subcommands, for use within batch files. See CALL, ECHO, FOR, GOTO, IF, PAUSE, REM, and SHIFT in Chapter 3.

Example

Runs a program called MY_INFO1 followed by a program called MY_INFO2, each of which displays a screen full of information. After running each of these programs, you clear the screen before proceeding. Your normal keystroke sequence is:

```
MY_INFO1 <Enter>
CLS <Enter>
MY_INFO2 <Enter>
CLS <Enter>
```

You could create a batch file called INFO.BAT containing these lines:

```
MY_INFO1
PAUSE
CLS
MY_INFO2
PAUSE
CLS
```

Which is executed by simply typing:

```
INFO <Enter>
```

After execution of MY_INFO1, the system pauses. When you press any key, it clears the screen and proceeds to execute MY_INFO2, after which it pauses again. When you press a key, it clears the screen and returns control at the prompt line.

Bypassing AUTOEXEC.BAT Commands

ROM-DOS offers the ability to bypass some or all of the commands in your AUTOEXEC.BAT and CONFIG.SYS files during bootup. This is a useful feature for tracking system problems that may be related commands in either of these two files. For a complete description of this feature, please refer to "Bypassing CONFIG.SYS and AUTOEXEC.BAT Commands" in the section describing configuring ROM-DOS with CONFIG.SYS.

File Storage

The sections that follow provide a basic introduction to computer memory and disk drives.

Basic Terminology

Computer files are stored either in the computer's internal memory (RAM and ROM) or on magnetic media, typically disks (diskettes and hard drives).

Regardless of where it resides, computer information is stored in **bytes**. A byte can be thought of as the amount of space it takes to store one character. Amounts of computer memory are measured in three terms:

KB stands for Kilobyte. One Kilobyte is thought of as one thousand bytes, although it is actually 1024 bytes in computer memory.

MB stands for Megabyte. One Megabyte is thought of as one million bytes; the real number is 1,048,576 bytes.

GB stands for Gigabyte. One Gigabyte is thought of as one billion bytes; the real number is 1,073,741,824 bytes.

Computer Memory: RAM and ROM

There are two basic types of internal memory in the computer: RAM and ROM.

RAM

RAM stands for Random Access Memory. This is the working memory in the system. Random access memory can be written to, read from, erased, rewritten, and so on. RAM is an electronic workspace for your computer's use during operation.

RAM is also called volatile memory. Its storage ability is temporary in nature, only holding information while the system's power is on. If the power is turned off or interrupted for any reason, everything stored in RAM is lost.

Within limits, the amount of RAM in a system can be changed. Typically, embedded systems have 512KB, 640KB or 1MB of RAM. Greater amounts of RAM are also possible. In desktop systems 4 MB and 8 MB are common.

ROM

ROM stands for Read Only Memory. ROM is far more permanent in nature than RAM; information is electronically **burned** into a ROM chip before the chip is installed in the computer. Information stored in ROM remains intact whether the system power is on or off.

Disks and Disk Drives

Computer disks can be classified into two basic groups: diskettes (the portable 5.25" **floppies** and 3.5" disks) and hard drives (also called fixed drives).

Diskettes

A diskette is a circular disk-shaped piece of magnetic material much like audio recording tape.

Information is stored on disks in concentric spirals, called tracks. These are subdivided into sectors. Each side of a 5.25" floppy could contain 40 tracks, each divided into 9 sectors. Standard diskette storage capacities range from about 180KB to 1.44MB, but capacity is increasing with technology and time.

There are two primary floppy drive types: 3.5" or 5.25". Typical storage space for the 3.5-inch drive is 1.44 MB, while the 5.25-inch stores up to 1.2 MB.

Hard Drives

Hard drives work much like diskettes but are fixed in the computer chassis and have a much higher storage capacity. 250 MB to 500 MB hard drives are not uncommon. ROM-DOS is capable of utilizing hard drives up to 4GB. Larger drives need to be partitioned into two or more drives.

Other Drives

Portions of RAM and ROM can behave like disk drives, complete with tree-structured directories.

RAM Disk

When a disk drive is created in RAM, it is called a virtual drive. Virtual drives can be read from and written to in the same manner as physical disk media. However, if system power is interrupted, all information on such a drive is lost.

ROM Disks

ROM drives are different from RAM drives in that they are written to only upon original creation. Thereafter, they can only be read from, much like a write-protected floppy disk. If you attempt to write to a ROM drive, the system will display an error message.

Configuring ROM-DOS (CONFIG.SYS)

Certain standard settings for your system's operation can be stored in a file called CONFIG.SYS. You may create or edit your own CONFIG.SYS file using a word processor or the COPY CON command. (see “Creating a Batch File”).

ROM-DOS offers three levels of CONFIG.SYS processing: DOS 3.31 compatible, DOS 5.0 compatible, and DOS 6.22 compatible. The level of processing available is determined when ROM-DOS is configured.

DOS 3.31 compatible commands include BREAK, BUFFERS, COUNTRY, DEVICE, FCBS, FILES, LASTDRIVE, NEWFILE, REM, and SHELL.

DOS 5.0 level processing includes all of the commands available with DOS 3.31 plus DOS, INSTALL, and STACKS.

DOS 6.22 commands include the commands from both the DOS 3.31 and 5.0 levels, with the addition of INCLUDE, MENUCOLOR, MENUDEFAULT, MENUITEM, NUMLOCK, SET, SUBMENU, and SWITCHES.

The CONFIG.SYS file must be placed in the root directory of the drive which is used for system startup or **bootup**. If a CONFIG.SYS file is not found, the listed default values are used for the following listed commands:

```
BREAK = OFF
BUFFERS = 2
COUNTRY = 001
FCBS = 4
FILES = 8
LASTDRIVE = E
NUMLOCK = ON
SHELL = COMMAND.COM /P /E:128
STACKS = 0,0
```

Examples

A typical CONFIG.SYS file might look like this:

```
DEVICE = RAMDISK.SYS 64
BREAK = ON
FILES = 15
BUFFERS = 15
```

Using Multiple Configurations

Your CONFIG.SYS file can be used to define multiple system configurations. This is handy when several people share a computer and require a different working environment. It is also useful for booting your own computer using different device drivers, paths, or settings depending on the intended computer tasks.

To define multiple configurations within the CONFIG.SYS file, you first have to define a startup menu. Each menu item represents a different system configuration option. Then, for each item on the menu, a configuration block is defined. Each configuration block contains the specific commands to be implemented as the system completes its bootup.

Configuration Blocks

The menu item definition and all configuration blocks are marked with a block header. A block header is a descriptive label enclosed in square brackets ([]). The start of the menu items must be marked with the block header [MENU]. Each configuration block may have a unique label of your choice. This label can be up to 70 characters long and can contain most printable characters, including spaces, backslashes (\), forward slashes (/), commas(,), semicolons (;), and equal signs(=). Square brackets ([]) cannot be used in block names.

The menu block (or submenu block) can contain only the following commands, described in the next section:

Menuitem
Menudefault
Menucolor
Submenu
Numlock

Although numlock can be used outside of a menu/submenu block, it is typically used to enable the use of the keypad for menu choice selections in the menu block.

A sample menu block might look as follows:

```
[MENU]
menuitem=Research, Research and Development
menuitem=WP, Word Processing
menuitem=Games, Games
menucolor=8,5
menudefault=WP, 10
```

When the system boots, you will see the defined menu on your screen.

```
ROM-DOS 6.22 Startup Menu

1. Research and Development
2. Word Processing
3. Games

Enter a choice: 1
```

Each menu item has its own configuration block. Items that are common to all menu choices are placed in a configuration block labeled [COMMON]. All instructions in the common block are carried out along with the specific instructions for any menu item. The [COMMON] block can also be placed at the end of your CONFIG.SYS file so that applications can append commands into this area as the application installs. You may have as many common blocks as you want. The instructions found in the common block(s) are processed in the order listed in the CONFIG.SYS file.

When the CONFIG.SYS file is processed by ROM-DOS, it displays the Startup menu defined in the [MENU] configuration block and then waits for your response. The choice made from the menu determines the configuration block whose commands are implemented. After the menu selection, processing starts with any instructions in CONFIG.SYS prior to the menu block. Then, instructions in the configuration block (including instructions added via an INCLUDE statement) and all common blocks are processed in the order listed in CONFIG.SYS. ROM-DOS ignores instructions in any nonselected configuration blocks or submenus.

To continue the above example, the configuration blocks might appear as follows:

```
[COMMON]
device=c:\romdos\himem.sys
dos=high
break=on

[RESEARCH]
files=20
buffers=50
device=vdisk.sys 128 /e

[WP]
files=10
buffers=10
lastdrive=m
device=c:\network\loadnet.sys

[GAMES]
include=wp
device=mouse.sys

[COMMON]
```

If choice 3 [GAMES] is made, the instructions in the [COMMON] configuration block are processed, followed by the instructions in the [GAMES]. [GAMES] uses the INCLUDE command. All instructions provided for the WP menu choice also apply to [GAMES]. If any instructions are in the final [COMMON], they are processed last.

Extending Menu Items To AUTOEXEC.BAT

The defined name of the menu item you have chosen becomes the value of the Environment Variable CONFIG. For example, if you choose 3, GAMES, from the preceding menu, the variable CONFIG is set to GAMES. The CONFIG environment variable is then used in your AUTOEXEC.BAT file to customize the startup sequence. This environment variable is referenced by "%CONFIG%" in your AUTOEXEC.BAT file. For more on Batch Files, please refer to preceding sections.

An example of an AUTOEXEC.BAT file that continues the customization process from the preceding MENU, could look like this:

```
prompt $p$g
set temp=c:\mystuff\temp
c:\virus\scanit.com

rem Go to section that matches menu
rem choice made in CONFIG.SYS

goto %config%

:RESEARCH
path c:\bin;c:\ROMDOS;c:\ROMDOS\utils;c:\BORLANDC
cd \ROMDOS
rem Skip other sections and move to end
goto end

:WP
path c:\bin;c:\ROMDOS;c:\wp
wp
rem Skip next section and move to end
goto end

:GAMES
path c:\bin;c:\ROMDOS;c:\gamedir
cd \gamedir
gamelist.bat
goto end

:end
```

Bypassing CONFIG.SYS And AUTOEXEC.BAT

ROM-DOS can bypass some or all of the commands in your AUTOEXEC.BAT and CONFIG.SYS files during bootup. This is a useful feature for tracking system problems that may be related to commands in either of these two files.

To bypass the instructions in both your AUTOEXEC.BAT and CONFIG.SYS files, follow these steps:

Turn on or re-start your computer by pressing <Ctrl><Alt>and <Del>, and wait for the following screen message:

```
Starting ROM-DOS...
```

As the above message is displayed, press <F5> or hold down <SHIFT>. The following message is displayed:

```
ROM-DOS is bypassing your CONFIG.SYS and AUTOEXEC.BAT files.
```

Your system then continues the boot process using the basic default configurations. You will notice differences in the way your system behaves. For instance, installable device drivers and memory device drivers are not loaded, and system prompts and path have default values. If COMMAND.COM is not in the root directory, ROM-DOS may not be able to locate it without assistance.

If you suspect that only one or two CONFIG.SYS commands are causing problems in your system, ROM-DOS can prompt for a confirmation before running each individual command. For this option, follow these steps:

Turn on or re-start your computer by pressing <Ctrl><Alt>and <Del>, and wait for the following screen message:

```
Starting ROM-DOS...
```

While the above message is displayed, press the <F8>. The following message is displayed:

```
ROM-DOS will prompt you to confirm each CONFIG.SYS command.
```

For each command, ROM-DOS displays the command and a Yes/No prompt:

```
FILES=40 [Y,N]?
```

To carry out the FILES=40 instruction, press (Y) for Yes. To bypass the instruction select (N) for No. ROM-DOS executes the command or bypasses it according to your instruction and then moves on to the next command in the CONFIG.SYS file. To bypass the confirmation prompt for the remaining instructions in the CONFIG.SYS file and skip the AUTOEXEC.BAT file, press <ESC> at any [Y,N]? prompt. To bypass all the remaining commands in both startup files, press <F5> at the [Y,N]? prompt.

When ROM-DOS finishes all of the commands in the CONFIG.SYS file, you are prompted for processing the AUTOEXEC.BAT file:

```
Process AUTOEXEC.BAT [Y,N]?
```

If you select Yes (Y), all of the commands in AUTOEXEC.BAT are processed. You are not prompted for each instruction. If you select No (N), the AUTOEXEC.BAT file is completely bypassed.

You can also be prompted for a single CONFIG.SYS command by using the question mark (?) prior to the equals sign (=) in the command line. Refer to the next section for a full description of the (?) command.

CONFIG.SYS Command Descriptions

For a complete description of each CONFIG.SYS command, see Chapter 3.

Installable Device Drivers

The device drivers described in this section can be installed using the CONFIG.SYS device command. In addition to the following device drivers, you can use device drivers that have been designed for DOS and included with your hardware and custom-made special-purpose device drivers.

DISPLAY

Purpose

Allows you to view international letters and symbols (code pages) on EGA and VGA displays.

Syntax

```
device =[d:][path]DISPLAY.SYS codepage[fontfilespec]
```

Remarks

DISPLAY immediately re-configures your video adapter to display characters from the selected code page instead of those characters built into the hardware.

codepage specifies the code page you want to display. ROM-DOS supports code pages 437, 850, 852, 860, 863, 865, and 866.

fontfilespec, when included, tells ROM-DOS where to find the EGA.CPI or EGA3.CPI font file. EGA.CPI is the default. If EGA.CPI is in the same directory as CONFIG.SYS, this can be omitted.

Currently, all font information for countries other than Russia and Czechoslovakia can be found in EGA.CPI. Information for Russia and Czechoslovakia is contained in EGA3.CPI.

For more information on DISPLAY, see “Configuring ROM-DOS for International Use.”

Examples

```
device=DISPLAY.SYS 850
device=c:\DOS\DISPLAY.SYS 850 c:\DOS\EGA.CPI
```

Configures the video adapter to display code page 850. The second example is used if the DISPLAY.SYS and EGA.CPI files are in the c:\DOS directory instead of the same directory as CONFIG.SYS.

EMM386

Overview

Enables expanded memory support for systems capable of such support expanded memory, typically 386 and higher CPU's.

Syntax

device=[*d:*] [*path*] HIMEM.SYS
device=[*d*:] [*path*] EMM386.EXE {xxxx-yyyy}
[Frame=*seg*[,*memK*]][ROM=*xxxx-yyyy*][D=*xxx*]

or

install=[*d*:] [*path*] EMM386.EXE {xxxx-yyyy}
[Frame=*seg*[,*memK*]][ROM=*xxxx-yyyy*][D=*xxx*]

Remarks

The HIMEM.SYS driver must be loaded for the EMM386 utility to function properly.

Datalight's EMM386 driver is a subset of a typical DOS EMM Manager. EMS and Upper Memory Block (UMB), VCPI and VDS support is included. EMM386 loads itself into the UMB area (between 640K and 1MB) if the first UMB range has enough room (about 8K required). EMM386 loads itself into the first UMB if there is sufficient room. This minimizes the amount of conventional RAM used.

This utility is limited to ROM-DOS 6.22 and greater versions. It will not function properly with older versions of ROM-DOS. It works with some older versions of ROM-DOS as an EMS manager only with UMB support. Testing indicates that it performs properly with MS-DOS 6.22 operating systems. No other operating systems have been tested.

EMM386 can be installed from CONFIG.SYS using a device statement or from AUTOEXEC.BAT with an install. In all cases, HIMEM.SYS must be installed prior to EMM386 in the CONFIG.SYS file.

xxxx-yyyy defines a range of memory to **fill-in** with extended memory. Up to eight ranges can be specified. The range can be specified as an exact range, for example D000 to DFFF; or as the starting and ending marks for the range, D000 to E000. In either of these examples, 64K has been defined.

Frame defines the starting segment for four 16K pages which can be mapped in and out at will. memK specifies how much memory to reserve for EMS, the default allocates all XMS memory for use as EMS.

D= specifies the amount of RAM in Kbytes (specified as a base 10 number) to be reserved for a VDS buffer (Virtual DMA Services). The default value is zero.

ROM= specifies a region of ROM to be supported by shadow RAM. This uses write protected RAM in place of ROM and can increase the speed of some systems.

The int 67H calls supported by this EMM386 memory manager include:

	Status
40H	Get Manager Status
41H	Get Page Frame Segment
42H	Get Number of Pages
43H	Get Handle and Allocate Memory
44H	Map Memory
45H	Release Handle and Memory
46H	Get EMM Version
47H	Save Mapping context
48H	Restore Mapping context
4CH	Get Pages Owned by Handle
4DH	Get Pages for All Handles
51H	Reallocate Pages

	Status
580xH	Get Mappable Physical Addresses
5900H	Get EMS Hardware Information
5A0xH	Allocate standard/raw pages
DexxH	Select VCPI calls

Examples

```
Device = EMM386.EXE C800-F000
```

Maps RAM from extended memory into the address C800:0 to just under F000:0 and defines an Upper Memory Block region there. The range could also have been specified as C800-EFFF.

```
Install = EMM386.EXE C800-D800 F000-F800
FRAME = E000,1024
```

Maps in RAM from extended memory into C800-D7FF and into F000-F7FF and defines Upper Memory Block Regions there. Also, EMS support is allowed with four 16K pages starting at E000:0 with 1024K (1 MB) worth of 16K pages.

```
Device = EMM386.EXE ROM=F000-FFFF
```

Specifies an address range for EMM386 to use for shadow RAM. The ROM represented by the address range is copied into RAM. The RAM area is re-mapped into the ROM address space. In this example, the BIOS normally occupying the 64k block at F000:0h is copied down into RAM and run from there. This may speed up your system if it does not already make use of shadow RAM.

```
Device =  EMM386.EXE D=20
```

Sets aside 20 kilobytes of memory for DMA. Another client, such as a network TSR, then utilizes this memory buffer for disk I/O.

HIMEM

Overview

Manages extended memory and the High Memory Area (HMA) in a 286, 386, or greater or PS/2 system. HIMEM prevents programs from simultaneously using the same area of memory for two different purposes. HIMEM supports the eXtended Memory Specification (XMS) 2.0. HIMEM is installed as a device driver in CONFIG.SYS.

Syntax

device = [*d*:] [*path*] HIMEM.SYS [/machine:*n*] [/A20[+]] [/PS2]

Remarks

HIMEM can be used to allow ROM-DOS to run in High Memory.

HIMEM supports a default of 32 handles.

HIMEM should not be used with older versions of Datalight VDISK. Current versions of VDISK uses XMS memory if it is available.

HIMEM now recognizes PS/2 style machines A20 line control. HIMEM determines whether to use the PS/2 A20 control or the AT A20 control method automatically by calling Int 15h, function C0h (get system configuration).

The automatic detection can be overridden with the “/Machine:n”, /A20 /A20+, or /PS2 command line switch if the autodetection on a given system is failing.

/Machine:1 and /A20 are equivalent and designate the PC AT A20 control method. These switches instruct HIMEM not to wait for the A20 line to settle.

/Machine:2 and /PS2 are equivalent and designate the PS/2 control method.

/A20+ is similar to the standard /A20 option; however it instructs HIMEM to wait for the A20 line to settle.

Examples

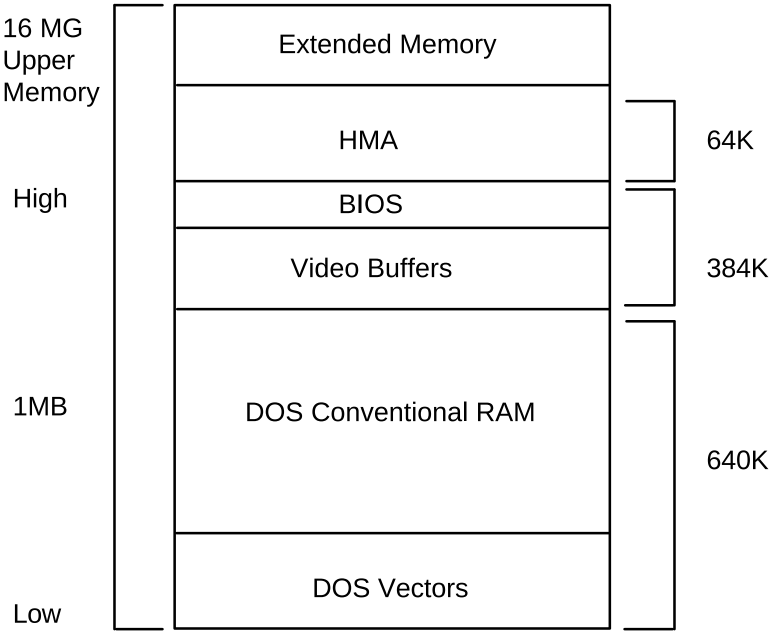
```
Device = HIMEM.SYS
```

Installs the XMS device driver. Once this driver is installed, accessing the HMA and Extended Memory (XMS) memory areas is legal. The Extended Memory area can contain up to 2 GB's of memory. Typical systems have 4, 8, or 16 MB's XMS memory installed.

```
Device = HIMEM.SYS /machine:1
```

Forces the use of the AT style A20 line control.

The HIMEM driver fails to load if the machine does not have memory above the 1 MB boundary or the BIOS does not provide support for it. It also fails to load if another XMS manager has been installed previously.



POWER.EXE

Purpose

Conserves power on a system that has APM (Advanced Power Management) by shutting down various subsystems (screen, disk drives, etc.) not being used. POWER.EXE can be used in an INSTALL command in CONFIG.SYS, in AUTOEXEC.BAT, or on the command line.

Syntax

From command line or from AUTOEXEC.BAT:

```
[path]power[/C#][/D#][/P#][/S#][/H]
```

From CONFIG.SYS:

```
Install=[path]power[/C#][/D#][/P#][/S#][/H]
```

Remarks

To work, the APM functions required to be supported by the system BIOS are:

Function	Description
00	APM Installed
01	Connect
04	Disconnect
05	CPU idle
06	CPU busy
07	Set Power State
08	Get PM Event

For each device, you can specify the length of time the device must be inactive before it is turned off. If you specify a length of zero for a device, POWER.EXE disables power management for that device.

# defines the number of seconds the device can remain inactive before it powers down.

/C# allows you to set the inactive time for the COM Ports. The default is two seconds.

/D# allows you to set the inactive time for the Disks. The default is 30 seconds. Currently all disk drives are treated as a single device.

/P# sets the inactive time for printers. The default time is 2 seconds.

/S# sets the inactive time for the screen. The default time is 9 seconds.

/H displays a user help screen.

Examples

```
POWER /S20 /C0 /P0
```

Runs POWER, turning off the screen after 20 seconds of inactivity, never turning off the COM and printer ports, and uses the default inactivity period for the disk drives (30 seconds).

VDISK

Purpose

Allows you to use memory as a disk.

Syntax

```
device = VDISK [size [secs[dirs]]] [/E]
```

Remarks

VDISK allows you to partition some of your computer’s memory as a disk. This disk, called a RAM disk or Virtual Disk, is much faster than either a floppy or hard disk. The RAM drive can use either standard DOS program memory or extended memory (above 1 Megabyte) for the disk.

Any data on the VDISK is lost when the system power is turned off.

size specifies the size of the VDISK in kilobytes. The default is 64K bytes. The memory selected is allocated from the DOS memory pool, decreasing the amount of memory available for programs, unless the extended memory switch is used.

secs specifies the sector size in bytes. The default is 512 bytes per sector. This value must be 128, 256, 512 or 1024. All other values are not valid and the default of 512 bytes are used.

dirs specifies the number of root directory entries. The default is 64 directory entries. There may be any number of root directory entries between 2 and 1024. If an odd number is given, it rounds up to the nearest multiple of 16, in order to fill the entire sector.

/E causes VDISK to use extended memory (memory above the 1 Megabyte boundary) instead of DOS program memory for the disk.

Interrupts are turned off during the transfer of data from extended memory to conventional memory.

The VDISK increases the resident size of DOS.

Examples

```
device = VDISK.SYS
```

Builds a 64K byte RAM disk in DOS memory.

```
device = c:\DOS\VDISK.SYS 220 /E
```

Builds a 220K byte RAM disk in extended memory. The VDISK device driver is loaded from the c: drive and the \DOS directory. VDISK assumes the default 512 byte sector size and 64 directory entries.

```
device = VDISK.SYS 45 128 18
```

Builds a 45K byte RAM disk in DOS memory. There are 128 byte sectors and 18 root directory entries.

Environment Variables

A block of memory is reserved for the definition of certain strings, called **environment variables,** used in the command processor environment. These include the settings you may establish with PATH and PROMPT as well as COMSPEC which is automatically defined by ROM-DOS at system startup.

Environment variables may be defined using SET, explained in Chapter 3.

Configuring ROM-DOS for International Use

You can configure ROM-DOS to conform to local conventions for date, time, and currency formats by using COUNTRY=. You can also use COUNTRY to select an international character set **code page** to determine the sort order. Any code page can also be shown on EGA and VGA displays by loading DISPLAY.SYS. You can re-map a keyboard to provide support for various languages and layouts by running KEYB.COM.

ROM-DOS uses a country code to identify which country's conventions are to be used. In most cases, the country code is the same as the international long distance telephone dialing code.

A code page is a set of 256 symbols, including letters, digits, punctuation, and graphic characters. The first 128 symbols in a code page are the standard ASCII characters and are identical in all code pages. The last 128 symbols vary depending on the code page. These symbols include the graphic line-drawing characters, plus many international letters, currency symbols, and other assorted symbols. Each code page is identified by a number, such as 437.

A computer display has one hardware code page built into it. Typically this is code page 437, the standard US code page. CGA and Monochrome monitors can only display the hardware code page in text mode. EGA and VGA monitors display the hardware code page unless you load special software (like DISPLAY.SYS).

Each country supports a default code page and an alternate code page. Here are the valid combinations:

Country	Code	Code Page	Alternative Code Page
Australia	061	437	850
Belgium	032	850	437
Brazil	055	850	437
Canadian-French	002	863	850
Czech Republic	042	852	850
Denmark	045	850	437
Finland	358	850	437
France	033	850	437
Germany	049	850	437

Country	Code	Code Page	Alternative Code Page
Hungary	036	852	850
Italy	039	850	437
Japan	081	932	---
Latin America	003	850	437
Netherlands	031	850	437
Norway	047	850	865
Poland	048	852	850
Portugal	351	850	860
Russia	007	866	437
Spain	034	850	437
Sweden	046	437	850
Switzerland	041	850	437
United Kingdom	044	437	850
United States	001	437	850
Yugoslavia	038	852	850

Changing Conventions

```
COUNTRY=049
```

Instructs ROM-DOS to use German conventions.

COUNTRY= requires COUNTRY.SYS to be present in the root directory of the boot drive.

Setting a country code affects date and time formats and the symbol used to denote currency

If you only specify a country code, ROM-DOS uses the default code page for that country. You can choose the alternate code page by including it in COUNTRY=. This command would tell ROM-DOS to use German conventions, for things like date and time, but use code page 437 instead of 850, the default code page.

```
COUNTRY=049,437
```

Setting a system code page affects the sort order for alphabetizing and the rules for converting international letters to upper case.

It is up to individual application programs to make use of these conventions. For example, DOS itself uses the date format for displaying directories and for showing and getting the current date and time. Some programs may choose to ignore the country information and continue to display dates in a specific format.

Displaying Different Code Pages

To display a code page other than the hardware code page, you must load DISPLAY.SYS in CONFIG.SYS. The following command sets the display to show code page 850, assuming both the DISPLAY.SYS driver and the EGA.CPI font file were located in the c:\DOS directory:

```
DEVICE=c:\DOS\DISPLAY.SYS 850 c:\DOS\EGA.CPI
```

The available font files are named EGA.CPI and EGA3.CPI. They are both used for EGA, VGA, and CGA systems. Refer to the section on Display.sys for the supported countries within each EGA file.

If you have an EGA or VGA system, the character font immediately switches to the requested code page. Some characters may look different after you load DISPLAY.SYS because ROM-DOS uses its own font for all 256 characters. For example, your hardware font might use a square zero character, but ROM-DOS might use a round zero character. The differences should be minor.

Printing Different Code Pages

At this time, ROM-DOS does not support printing code pages other than those stored in the printer hardware.

Changing the Keyboard Layout

To alter the keyboard layout, use KEYB from within DOS. This can be done by running KEYB in AUTOEXEC.BAT or directly from a DOS prompt. You might use this command to switch to a German keyboard layout:

```
KEYB GR
```

For each country, there are two valid code pages. If you do not specify a code page, the default code page is used.

Country	Country Identifier	Code Page	Alternative Code Page
Australia	us	437	850
Belgium	be	850	437
Brazil	br	850	437
Canadian-French	cf	863	850
Czech Republic	cz	852	850
Denmark	dk	850	437
Finland	su	850	437
France	fr	850	437
Germany	gr	850	437
Hungary	hu	852	850
Italy	it	850	437
Japan	---	932	---
Latin America	la	850	437

Country	Country Identifier	Code Page	Alternative Code Page
Netherlands	nl	850	437
Norway	no	850	865
Poland	pl	852	850
Portugal	po	850	860
Russia	ru	866	437
Spain	sp	850	437
Sweden	sv	437	850
Switzerland	sf	850	437
United Kingdom	uk	437	850
United States	us	437	850
Yugoslavia	yu	852	850

After you have loaded KEYB, your keyboard layout reflects the country you chose. You can switch back to the US keyboard layout at any time by pressing <Ctrl><Alt><F1>. You can return to the modified keyboard layout by pressing <Ctrl><Alt><F2>.

You can also switch to a completely different layout by running KEYB again and specifying another country identifier.

Native-language keyboards tend to have different layouts from US keyboards. KEYB does its best to map the available hardware keys to the desired layout. Some symbols may not be available when using a US keyboard and a non-US layout.

The “AltGR” key is not found on a standard U.S. keyboard.

Some keys are prefix keys which don't generate any symbol by themselves but modify the following keystroke. For example, on most European keyboards the apostrophe key (') causes the next letter to be accented. To produce an apostrophe by itself, press the apostrophe key followed by the space bar. Other keys which may behave as prefixes, depending on the current keyboard layout, are the backward apostrophe (`), tilde (~), and caret (^).

Some keys represent symbols which are not available in all code pages. For example, the German keyboard can produce a capital A with a caret above it. In the default German code page 850, that symbol is represented by the code 182. However, in the alternate German code page 437, there is no such symbol. If you are using the German layout and code page 437 to produce a capital A with a caret above it, you get a caret character followed by a capital A (^A).

Note that the keyboard code page could be set *not* to match the display code page. This can lead to confusion, as the keyboard may produce characters which appear on screen as other symbols. Continuing the above example, if you are using the German layout with keyboard code page 850 and display code page 437, to produce a capital A with a caret above it, you see a graphic box drawing character on your screen.

Putting It All Together

To configure your system completely, you need to include commands in your CONFIG.SYS and AUTOEXEC.BAT files. The following sample files set up a computer to use German conventions, to switch the keyboard layout to German, and to use code page 850 for sorting, upper-case conversions, and the display. The commands relevant to international issues are shown in bold type; other typical commands are shown in normal type.

COUNTRY.SYS is assumed to be in the root directory of the boot drive, and DISPLAY.SYS, EGA.CPI, KEYB.COM and KEYBOARD.SYS are assumed to be in the c:\DOS directory.

CONFIG.SYS

```
BUFFERS=20
FILES=20
COUNTRY=049
DEVICE=c:\DOS\DISPLAY.SYS 850 c:\DOS\EGA.CPI
```

AUTOEXEC.BAT

```
@ECHO OFF
PROMPT $P$G
PATH c:\DOS;c:\UTILITY
KEYB GR
```

ROM-DOS 6.22 COMMAND DESCRIPTIONS

3

Brief Description of Commands

Following are brief descriptions of all ROM-DOS commands, including batch file commands.

Command	Description
?	CONFIG.SYS command. It directs ROM-DOS to pause for confirmation before processing a command.
@	Used to suppress the display of a single batch file command line.
;	Identifies non-executing lines (same as REMark).
ATTRIB	Displays or modifies the attributes associated with a file.
BREAK	Turns on or off the ability to stop program execution at a non-I/O point.
BUFFERS	Sets the number of internal data buffers.
CALL	Batch subcommand. Invokes execution of a secondary batch file.
CDEX	Device driver. Allows access to the cd-rom drives on your system.
CHDIR	CHange DIRectory (also CD). Changes the default directory.
CHKDSK	CHecK DiSK. Checks integrity of data on a **DISK**. Displays information.
CHOICE	Allows a user to make a processing choice during the execution of a batch file.
CLS	CLear Screen. Clears all information from the monitor.
COMMAND	Spawns a second DOS command processor.

Command	Description
COPY	Copies files from one storage location to another.
COUNTRY	Designates the country code for displays
CTTY	Change TeleTYpe. Changes the default terminal interacting with ROM-DOS.
DATE	Displays the date from the system's internal calendar. Allows revision.
DEL	DELete. Deletes specified files
DEVICE	Installs a device driver into ROM-DOS.
DEVICEHIGH	CONFIG.SYS command. Loads a device into the upper memory area if available.
DIR	DIRectory. Lists contents of a specified directory.
DISKCOPY	Copies the contents of one floppy disk to another of the same type.
DOS	Installs ROM-DOS into High Memory Area (HMA).
ECHO	Batch subcommand. Turns on or off display of batch execution on the monitor.
ERASE	Erases specified files (same as DEL).
EXIT	Used to exit "nested" running of ROM-DOS within another program.
FCBS	Specifies the number of File Control Blocks (FCBS) open at one time.
FDISK	Initializes and partitions a hard disk for DOS.
FILES	Sets the maximum number of files that can be open at one time on the system.

Command	Description
FIND	Works as a filter to display only lines that contain a specified string.
FOR	Batch subcommand. Performs one DOS command on a set of files.
FORMAT	Initializes a disk so that ROM-DOS can access files on that disk.
GOTO	Batch subcommand. Moves control to a specified line in the batch file.
HELP	Lists all available ROM-DOS commands along with brief descriptions.
IF	Batch subcommand. Performs a command based on a specified condition.
INCLUDE	CONFIG.SYS menu configuration command. Allows instructions in one configuration block to be included with instructions in another configuration block.
INSTALL	Loads terminate and Stay Resident (TSR) programs during CONFIG.SYS processing.
KEYB	Allows altering of the keyboard layout for a different language or nationality.
LABEL	Creates, changes or deletes a disk volume label.
LASTDRIVE	Sets the maximum number of drives.
LOADHIGH	Batch subcommand or command line command. Loads a program into the upper memory area if available.
MENUCOLOR	CONFIG.SYS menu configuration command. Allows setting of text and background colors for the startup menu.

Command	Description
MENUDEFAULT	CONFIG.SYS menu configuration command. Sets the default menu item choice and time-out value for making a selection.
MENUITEM	CONFIG.SYS menu configuration command. Specifies an item to be placed on the startup menu displayed during system boot.
MKDIR	MaKe DIRectory (also MD). Creates a new subdirectory.
MODE	Modifies the operation of the printer, serial port and active video display.
MORE	Displays a text file one screen at a time.
NEWFILE	Allows continuation of CONFIG.SYS processing from a new file.
NUMLOCK	Sets the NUMLOCK keyboard key to on or off when your computer starts.
PATH	Displays current command search path(s). A new path line can be specified.
PAUSE	Batch subcommand. Halts execution until a key is pressed.
PRINT	Prints a list of files, up to ten files.
PROMPT	Resets the appearance of the system prompt line.
REM	REMark batch subcommand. Identifies non-executing lines.
REN	REName. Renames files.
RMDIR	ReMove DIRectory (also RM). Deletes a specified subdirectory.
SET	Sets environment variables and command processor strings.
SHARE	Installs the capabilities for file sharing and file locking on your hard disk.

Command	**Description**
SHELL	Allows selections of an alternate boot program other than the default COMMAND.COM command processor.
SHIFT	Batch subcommand. Shifts replaceable parameters one position "to the left."
SORT	Sorts a text file and displays the output to the standard device.
STACKS	Allows for the use of dynamic data stacks to handle interrupts. *Note: ROM-DOS does not utilize this command.*
SUBMENU	CONFIG.SYS menu configuration command. Defines a menu item that represents a secondary menu.
SWITCHES	Allows special CONFIG.SYS file options.
SYS	Transfers the hidden system files to a specified drive.
TIME	Displays current time from the system's internal clock. Allows revision.
TREE	Displays the path of each directory on a specified drive.
TYPE	Displays the contents of a text file on the monitor.
VER	Displays current version of ROM-DOS on the monitor.
VERIFY	Displays the current VERIFY state or set the VERIFY state to on or off.
VERSION.SYS	Modifies the version # ROM-DOS reports.
VOL	Displays the VOLume label on a disk.
XCOPY	Copies multiple files and optionally subdirectories

Full Description of Commands

The following pages provide complete descriptions of each ROM-DOS command -- including batch subcommands.

The entries are in alphabetical order. Each entry includes a description of the command's purpose, command entry syntax, remarks, and examples as appropriate.

Each command also has a label to designate whether it is an Internal or External command. Internal commands are part of the command processor program COMMAND.COM. These functions are only available while COMMAND.COM is running. External commands are actually stand-alone utility programs. They are independent from COMMAND.COM. A special notation is also made for those Internal commands that are unique to CONFIG.SYS processing. These commands, labeled Internal/CONFIG.SYS, can only be used inside a CONFIG.SYS file.

For an explanation of typing conventions, see “Conventions Used in This Manual” in the introductory materials of this book.

For Online Help information and syntax descriptions, use the /? option with any command.

The file COMMAND.HLP must be available in the root directory on the boot drive to get help information for Internal commands.

Example

```
DIR /?
```

Displays information and syntax for the DIR command.

?

Internal

Purpose

The question mark (?) command is placed on a command line in the CONFIG.SYS file following the actual command. It directs ROM-DOS to pause and ask for confirmation before processing the command.

Syntax

```
[command]? = command_arguments
```

Remarks

The *command* can be any of the standard CONFIG.SYS commands in the following list:

BREAK=
DEVICE=
DOS=
FILES=
STACKS=
BUFFERS=
FCBS=
INSTALL=
LASTDRIVE=
SWITCHES=

The *command_argument* can be any of the available options defined for the command. Please refer to the individual command description for complete instructions.

The *question mark (?)* should be placed just before the equal sign (=) in the command line.

Examples

```
DEVICE?=VDISK.SYS 64 /E
```

Causes ROM-DOS to pause and ask for confirmation before installing the VDISK. If Yes (Y) is answered, the installation continues. If No (N) is answered, the device is not loaded and processing moves on to the External CONFIG.SYS command line.

Internal

Purpose

The AT sign @ is used to prevent a single command in a batch file from being echoed to the screen as the batch file is being run. The @ sign is placed in front of the command whose display is to be suppressed.

Format

@ [*batch file command*]

Remarks

batch file command can be any executable line in your batch file.

Examples

```
@COPY FILE1.BAT FILE1.SAV
```

Executes the COPY instruction, which are not echoed to the screen as the batch file runs.

```
@ECHO OFF
```

The ECHO OFF command differs from the @ sign in that it causes all subsequent commands *not* to be displayed on the screen. To prevent the ECHO OFF command from displaying itself, place the @ sign in front of the command.

;

Internal

Purpose

The semicolon (;) command has two purposes: to allow comments in a batch or CONFIG.SYS file, and to temporarily disable a command without physically deleting the command from the file. See also the REM command.

Syntax

```
;any text here.
```

Remarks

The (;) command is used to functionally remove a command from the CONFIG.SYS file without actually deleting it from the CONFIG.SYS file.

Examples

```
;c:\BIN\VDISK.SYS 64 /E
```

Causes the VDISK command not to execute until the (;) command is removed.

ATTRIB

External

Purpose

Displays or modifies the attribute of a file.

Syntax

ATTRIB [+ | -][*option*][*filespec*]

Remarks

ATTRIB defines the characteristics of a file. It determine if a file can be deleted or modified, or if it is archived. The ATTRIB command is used to manage these file attributes.

Wildcard characters may be used in the ATTRIB *filespec*.

The ATTRIB command modifies file attributes if modify commands are given to ATTRIB. The modify commands are:

Option	Description
+/-	Add(+) or remove(-) attribute
A	Archive attribute
C	Clear all attributes
H	Hidden file attribute
R	Read Only attribute
S	System file attribute

If no modify commands are found by ATTRIB, then the files are displayed along with the file names and their current attributes.

ATTRIB (cont.)

Examples

```
ATTRIB +r myfile.dat
```

Adds the Read Only attribute to the file myfile.dat.

```
ATTRIB -a -r *.dat
```

Removes the Read Only attribute and the Archive attribute of all files with the DAT extension.

```
ATTRIB *.dat
```

Displays the attributes of all files with the DAT extension.

BREAK

Internal

Purpose

Expands the list of operations that can be stopped by pressing <Ctrl><C> or <Ctrl><Break>. Alternatively, returns to the default setting of a limited number of "break-able" operations.

Syntax

```
BREAK [ON|OFF]
```

Remarks

In the normal default condition, the BREAK switch is off. In the off mode, the "stop" commands, <Ctrl><C> and <Ctrl><Break>, affect activities that read from or write to the keyboard, the screen, or the printer. ROM-DOS does not look for these stop commands during any other activities.

With the BREAK switch set to ON, ROM-DOS looks for <Ctrl><C> and <Ctrl><Break> during other activities such as disk reads and writes.

Examples

```
BREAK ON
```

Expands the BREAK list.

```
BREAK OFF
```

Returns to limited BREAK list.

```
BREAK
```

Displays the current BREAK setting.

BUFFERS

CONFIG.SYS

Purpose

ROM-DOS has internal buffers to temporarily hold data read from the disk. Increasing the number of internal buffers will speed system performance.

Syntax

BUFFERS = *number*

Remarks

Each buffer used by ROM-DOS requires 512 bytes of RAM. The BUFFERS command increases or decreases the amount of RAM used by the operating system.

The minimum number of buffers is 2 and the maximum number is 40. If a number less than 2 is given, the number of BUFFERS is set to 2. If a number larger than 40 is given, BUFFERS is set to 40.

Examples

```
BUFFERS = 10
```

Causes ROM-DOS to have 10 buffers. These 10 buffers use 5120 bytes of RAM.

CALL Batch Subcommand

Internal

Purpose

CALL invokes execution of a secondary batch file without exiting the primary batch file. When the secondary batch file is done executing, control is returned to the primary batch file.

Syntax

CALL *batchfile* [*batchfile arguments*]

Remarks

Parameters for the secondary batch file may also be included, if appropriate.

Examples

```
CALL BATCH2
```

Executes the batch file BATCH2.BAT.

```
CALL MYBATCH FILEX FILEZ
```

Executes the batch file MYBATCH.BAT.

The arguments passed to MYBATCH.BAT are:

%1 = FILEX
%2 = FILEZ

CDEX

External

Purpose

CDEX is a device driver allowing access to the CD-ROM drives on your system. This driver is loaded in autoexec.bat or from the command line. The driver provided with your CD-ROM hardware must be loaded in config.sys prior to running CDEX.EXE.

Syntax

[*d:*][*path*]CDEX.EXE [/d:<*name*>] [/H|?][/L:<*letter*>][/M:<*number*>] [/X]

Remarks

/D:<Name> specifies the CD-ROM device name. The default name is MSC002. This name must match the name used for the config.sys installed device driver provided with the CD-ROM hardware.

/H or /? displays the help screen with syntax and examples.

/L:*letter* specifies the drive letter to use for the CD-ROM drive. The next available drive letter is assigned if the /L option is not used.

/M:*number* specifies the number of sector buffers to use while accessing the data on the CD-ROM drive. The default value is 4.

/X Use XMS memory for CD-ROM buffers

Examples

```
CDEX
```

Installs the CD-ROM using default values and names

CDEX (cont.)

```
CDEX /d:MSC003
```

Uses an alternate name for the device

```
CDEX /L:G
```

Assigns a specific drive letter, for example G rather than the next available drive letter

```
CDEX /M:6
```

Increases the number of buffers used by CDEX from 4 to 6

CHDIR (CHange DIRectory)

Internal

Purpose

Changes the default directory.

Syntax

CHDIR [*d:*][*path*]*subdir*

CD *[d:][path*]*subdir*

Remarks

S*ubdir* is the name of the new default subdirectory. Note that CD may be used instead of CHDIR.

The new directory which is to become the default must already exist. See MKDIR for creation of subdirectories.

A series of two periods (..) can be used to indicate a move back to the parent directory.

Specifying only the backslash (\) for the subdir argument moves you to the root directory of the current default drive.

Examples

CHDIR \TOOLS

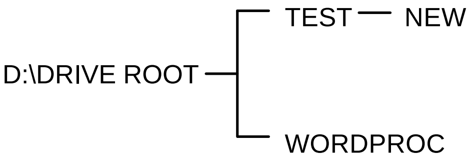
Moves you into the subdirectory called TOOLS, whose parent directory is the root of the current default drive.

CHDIR (cont.)

```
CD a:
```

Displays the current directory on drive a:. Any valid drive letter can be substituted to get the current directory on that drive.

The following examples all use this directory tree structure:



```
CD d:\TEST\NEW
```

Moves you into the subdirectory called NEW, located on the d: drive, under the parent directory TEST.

```
CHDIR ..
```

Moves you back to the parent directory of the current subdirectory. If you were in the directory d:\TEST\NEW (from the previous example), this CHDIR command moves you from NEW back into the TEST directory.

```
CD ..\WORDPROC
```

Moves you back to the parent directory and then into a subdirectory called WORDPROC. If we started in the TEST directory, we move back to the ROOT directory and then into the WORDPROC subdirectory.

```
CD \
```

Moves you back to the root directory, from any starting point in the directory tree.

CHKDSK (CHecK DiSK)

External

Purpose

The CHKDSK command checks the disk directories and File Allocation Table (FAT) and displays a disk and memory report.

Syntax

```
CHKDSK [d:][path][filespec][/C] [/F] [/V]
```

Remarks

CHKDSK examines a disk, determines if the disk has any errors in the File Allocation Table(FAT), and optionally fixes errors.

Options

/F causes CHKDSK to fix errors on the disk if any were found. The errors that can be found are directory or FAT errors. If the /F is not specified, then CHKDSK acts as if it will fix the disk, but the corrections are not written out to the disk.

If errors are detected, you are prompted with a message similar to the following:

```
15 lost allocation units found in 5 chains.
Convert lost chains to files?
```

If you answer Y for Yes, each lost chain is written to a file in the root directory of the current default drive. Each file has the name FILE*nnnn*.CHK. *nnnn* will be a sequential number. The first chain will be in FILE000.CHK. These files can be verified to see if they contain valuable information and then deleted if desired. If you answer N for No to the above prompt, CHKDSK still makes the corrections; however the lost chains are not saved to the disk.

CHKDSK (cont.)

/C allows CHKDSK to correct errors without user confirmation. This option must be used along with the /F option for corrections to be made.

/V causes CHKDSK to display each path and file as it is processed.

If a file specification is entered, then CHKDSK displays all matching files that have non-contiguous data areas on the disk. Files stored in non-contiguous areas, especially .EXE files, have slower disk access times. If CHKDSK reports a large number of files with this problem, use a utility program to optimize the files and free space on your disk.

After checking the disk, CHKDSK displays any error messages followed by a report on the state of the disk that was checked.

An example of the report is shown below:

```
Volume ROM-DOS created June 1,1990 1:00a
Volume Serial Number is 190E-4AA2

 362496          bytes total disk space
      0          bytes in 1 hidden files
   6144          bytes in 2 user files
 356352          bytes available on disk


 655360          bytes total memory
 595360          bytes free
```

CHKDSK does not wait for a disk to be inserted before the checking is initiated, nor does it repair any errors.

Examples

```
CHKDSK a:
```

Checks the integrity of drive A and prints the report to the console.

CHKDSK (cont.)

```
CHKDSK d: >drive_d.rpt
```

Checks the integrity of hard drive disk D and saves the report in a file called DRIVE_D.RPT.

CHOICE

External

Purpose

CHOICE allows a user to select between different options during the processing of a batch file.

Syntax

[*d:*][*path*]CHOICE [/C[:]*keys*] [/N] [/S] [/T[:]*c,nn*] [*Prompt Text*]

Remarks

/C[:]*keys* specifies the allowed keys for the user prompt. The colon is optional for the command syntax. The default selection for keys is “YN”. More than two key choices may be entered.

/N causes CHOICE not to display the user prompt. The selected keys are still valid, but they do not display in the prompt message.

/S causes CHOICE to be case sensitive regarding the prompt. With the /S switch, you must enter the response in the exact case used with the /C option.

/T[:]*c,nn* is used to set a time delay. CHOICE pauses for the specified number of seconds waiting for a response. If none is given, the default key choice is used. C is the default key choice character. nn specifies the number of seconds to pause. Valid number selections are from 0 to 99. With 0, there is no pause.

Text for the prompt is optional. You can display different output by using or not using the text field and also by using or removing the /N switch along with the text.

CHOICE (cont.)

Examples

```
CHOICE /c:ync
```

Displays `[Y,N,C]?`

```
CHOICE /c:ync Please select Yes, No, or Continue
```

Displays `Please select Yes, No, or Continue [Y,N,C]?`

```
CHOICE /n Continue reading file
```

Displays (note key choice of YN is default) `Continue reading file?`

CHOICE (cont.)

A complete batch file example follows. Each prompt selection returns an error level that can be trapped. The error level corresponds to the order of the key choice. For example, with /c:teo, t returns 1, e returns 2, and o returns 3.

```
@echo off
cls
echo.
echo T  Run TIME-IN Program
echo E  Run Employee Update
echo O Run TIME-OUT Program
echo.
choice /teo /t:t,5 Please select option
if errorlevel 3 goto timeout
if errorlevel 2 goto update
if errorlevel 1 goto timein



:timeout
tmout.exe
goto end
:update
updat.exe
goto end
:timein
tmin.exe
goto end
:end
```

CLS (CLear Screen)

Internal

Purpose

Clears the monitor to display a blank screen.

Syntax

```
CLS
```

Remarks

CLS clears the screen and then re-displays the DOS prompt and cursor in the upper left-hand corner. There are no additional options for CLS.

COMMAND (Command Processor)

External

Purpose

Starts a new Command Processor.

Syntax

COMMAND[*device*][*/e:number*][*/k:filename*][*/p*]
[*/c string*][*/msg*]

Remarks

This command starts a new copy of the ROM-DOS Command Processor. The Command Processor is the program having all the internal DOS commands.

Starting a new Command Processor also produces a new Environment. The size of the environment is 128 bytes by default, but can be changed using the /e switch.

Command and its arguments can also be used in a SHELL= statement in your CONFIG.SYS file. See the full description of SHELL for more details.

Options

device specifies COMMAND.COM to use a different device, such as AUX, for input and output.

/e:number sets the environment size. *Number* represents the size of the environment in bytes. The range for *Number* must be from 160 to 32768. All other values are ignored and the default value of 256 is used. ROM-DOS rounds the value entered up to the nearest multiple of 16.

COMMAND (cont.)

/k:filename tells the command processor to run the specified file name and then display the ROM-DOS command prompt. It is not recommended that this option be used in a CONFIG.SYS SHELL= statement.

/p causes COMMAND not to exit, but to remain permanent. */p* should only be used when command is used in a CONFIG.SYS SHELL statement.

/c string causes COMMAND to execute the command in *string* and then terminate. The string-command can be any internal or external command.

/msg indicates that all error messages be stored in memory. This option is recommended only for diskette-based systems. ROM-DOS keeps many of its error messages in the non-resident portion of COMMAND.COM rather than using valuable memory to store them. If an error message is needed and you have loaded ROM-DOS from a diskette, the message is only available if the boot disk is still in the drive. By using */msg*, the messages are available in memory at all times. */p* must be used along with the */msg* option.

Examples

```
COMMAND /C DIR c:
```

Executes a new copy of COMMAND. It performs a DIR command on the C drive and then exits back to the previous Command Processor..

```
SHELL=c:\COMMAND.COM /P /E:256
```

Shows loading of a permanent copy of command with an environment size of 256 bytes.

COPY

Internal

Purpose

Copies a file or set of files to a specified destination on another disk, or on another subdirectory on the current disk, or on a completely different drive. COPY may also be used to alter the *filename* within the current directory. In addition, this command can be used to direct communication between files and devices (for example, file contents to a printer, keyboard input to a file).

Syntax

This command has several possible formats. The essential structure of each command is:

COPY source target option

The *source* is the **copy from** *filespec* or *device*, and the *target* is the **copy to** *filespec* or *device*. Following are various configurations of the COPY command format;

The first *filespec* indicates the source file(s) to be copied and the second *filespec* indicates the target area on which to copy.

COPY [*d:*][*path*]*filename*[*/option*]
[*d:*][*path*]*filename*[*/option*]

As shown, several source *filespecs* can be copied into the target *filespec* listed last. The source files concatenates one after the other into the target file.

COPY [*d:*][*path*]*filename*[*/option*] + ...
... + [*d:*][*path*]*filename*[*/option*]...
...[*d:*][*path*]*filename*[*/option*]

The target *device* is a console or printer (PRN).

COPY [*d:*][*path*]*filename*[*/option*] *device*

The *device* is the source such as a keyboard or console, the output of which is directed to the target *filespec*.

COPY *device* [*d:*][*path*]*filename*[*/option*]

Options

/A and /B stand for ASCII and binary, respectively. They act as switches that allow each of these file types to be copied. When a switch is used, it applies the preceding *filename*, and remains in effect for any subsequent *filenames* until superseded by another /A or /B.

Usually /A and /B are not necessary unless you are combining ASCII and binary files.

/A indicates that the file is treated as an ASCII file (text file). When /A is used with the source file, everything is copied up to, but not including, the first <CTRL><Z> end-of-file marker. When /A is used on the target file, a <Ctrl><Z> is added as the last character in the file.

/B indicates that the file is to be treated as a binary file. When /B is used with the source file, the entire file is copied regardless of any <Ctrl><Z> characters. When /B is used with the target filespec, no <Ctrl><Z> end-of-file mark is added.

/H copies hidden files.

/V (Verify) is not implemented in ROM-DOS, for code size reasons. It is included to maintain command-line compatibility, but has no functional impact.

/Y copies the current file(s) over the existing file(s) of the same name(s) without confirmation. Using this option overrides the setting made by the COPYCMD environment variable.

COPY (cont.)

/-Y requests confirmation for the copy of one file over the existing file of the same name. Using this option overrides the setting made by the COPYCMD environment variable.

The COPYCMD environment variable is set using the SET command. This allows you to set confirmation on or off for the COPY command. If you always want to be prompted for confirmation when a file will copy over an existing file, set COPYCMD= /-Y. To automatically overwrite without confirmation during a copy instruction, set COPYCMD= /Y. Refer to the SET command for proper usage.

Remarks

When no file name is specified for the target, the new copy is given the same name as the source file name.

When no *drive* or *path* is specified for the source, the current drive and directory is assumed. When no *drive* or *path* is specified for the target, the current drive and directory are assumed.

If a drivename only is specified without a *path*, the default directory for that drive is assumed.

Both source and target *filespecs* may include wildcard characters (*) and (?) to specify a set of several files.

Examples

```
COPY LETTER.TXT a:
```

Copies the file LETTER.TXT (in your current default drive and path) to the default directory on the disk in drive A.

```
COPY *.DOC a:
```

Copies all files in the default directory with an extension of .DOC to the default directory of drive A.

COPY (cont.)

```
COPY DATAORIG.DOC DATABACK.DOC/V
```

Creates a backup copy, DATABACK.DOC, from the file DATAORIG.DOC, verifies that the file was correctly copied. The new file is located in the current default directory.

```
COPY JAN.DAT + FEB.DAT + MAR.DAT QTR1.DAT
```

Copies the files JAN.DAT, FEB.DAT, and MAR.DAT in sequence into the single file, QTR1.DAT.

```
COPY CON NEWFILE.TXT
```

Sets up your console (keyboard) to input directly to NEWFILE.TXT. <Ctrl><Z> followed by <Enter> closes the file and returns to normal command line operation.

COUNTRY

CONFIG.SYS

Purpose

ROM-DOS supports multiple country formats for time, date, and currency, as well as other basic country-specific information. A country is identified by a three digit international telephone country code.

Syntax

COUNTRY = *countrynumber* [*codepage*]

Remarks

The file COUNTRY.SYS must be present in the same directory as your CONFIG.SYS file.

If you do not specify a code page, ROM-DOS uses the default code page for the chosen country. If a code page is specified, it must be either the default or alternate code page for the chosen country.

The ROM-DOS DATE and TIME commands are affected by this command. Applications using DOS functions to determine the date, time or currency format, or requesting that DOS provide character sort order or uppercase information are affected as well.

The following table shows the currently supported countries:

Country	Code	Code Page	Alternative Code Page
Australia	061	437	850
Belgium	032	850	437
Brazil	055	850	437
Canadian-French	002	863	850
Czech Republic	042	852	850
Denmark	045	850	437
Finland	358	850	437
France	033	850	437
Germany	049	850	437
Hungary	036	852	850
Italy	039	850	437
Japan	081	932	---
Latin America	003	850	437
Netherlands	031	850	437
Norway	047	850	865
Poland	048	852	850
Portugal	351	850	860

COUNTRY (cont.)

Country	Code	Code Page	Alternative Code Page
Russia	007	866	437
Spain	034	850	437
Sweden	046	437	850
Switzerland	041	850	437
United Kingdom	044	437	850
United States	001	437	850
Yugoslavia	038	852	850

For more information on COUNTRY, see the section on Configuring ROM-DOS for International Use. The default country code is 001 (U.S.A.), code page 437.

Examples

```
COUNTRY= 049
COUNTRY = 049, 437
```

The next time you start ROM-DOS with either of these COUNTRY commands, the DATE and TIME will be displayed as follows:

```
DATE
    Current date is Wed 20.06.1990
    Enter new date (dd.mm.yy):

TIME
    Current time is 16:39:54,45
    Enter new time:
```

The first COUNTRY command above would use code page 850, by default, for sorting and case conversion. The second COUNTRY command example would use the specified code page 437 instead.

CTTY (Change TeleTYpe)

Internal

Purpose

Allows you to direct input and output to a device other than your computer's standard keyboard and monitor.

Syntax

CTTY *device*

Remarks

CTTY is used in any situation requiring interaction with an alternate console.

CAUTION

The CTTY command only affects communication with ROM-DOS and with programs that work through ROM-DOS for input and output. For example, if you go into BASIC, CTTY uses standard keyboard input regardless of previous command usage.

Examples

```
CTTY COM2
```

Sets the device on COM2 as the input/output device.

```
CTTY CON
```

Returns control to the standard keyboard.

DATE

Internal

Purpose

Displays the current date (day, month, year) known to ROM-DOS and also allows you to change it.

Syntax

```
DATE [mm-dd-yy]
```

Remarks

The date set by this command is used for **date stamping** your file revision dates. This information displays when you execute a directory listing of your files.

If your computer does not have an internal battery-operated clock, you may want to include the DATE command in your AUTOEXEC.BAT file to set the date at boot-up.

The format of the date command is also dependent on the Country specified in CONFIG.SYS. The date is displayed according to local standards for the specified country.

Also see the TIME command.

Examples

If the command is entered without the specification of *mm-dd-yy*, the current date known to ROM-DOS displays and you are prompted to enter a new date.

```
Current date is Sat 6-10-1989
Enter new date (mm-dd-yy):
```

If you do not want to change the date, just press <Enter>. Otherwise, key in the current date and press <Enter>.

DATE (cont.)

Alternatively, you may skip the display and prompting by simply entering the current date on the command line. To enter June 10, 1989, type the DATE command as follows (the century number 19 is assumed):

```
DATE 6-10-89
```

Valid entries for months, days, and years are:

mm = 1-12   *dd* = 1-31   *yy* = 80-99
(for years with 19 as the century)

ROM-DOS figures out the day of the week; do not include it in your entry.

The earliest year that ROM-DOS can accept is 1980. ROM-DOS assumes that the century is 19 unless something different is entered. For example, February 18, 2014 is:

```
DATE 02-18-2014
```

ROM-DOS adopts your entry as the current date.

DEL (DELete)

Internal

Purpose

Deletes a specified file or set of files.

Syntax

DEL [*d:*][*path*]*filename* [/P]

Remarks

DEL and ERASE are functionally identical.

When no drive is specified, the default is assumed. When no path is specified, the default path is assumed.

The wildcards characters ? and * can be used in the *filespec*. This should be done with caution to avoid deleting multiple files unintentionally.

When the *filespec* *.* is used to delete all files in the specified subdirectory, a verification message displays:

```
Are you sure (Y/N) ?
```

Type Y only if you want all files in the specified subdirectory to be deleted.

CAUTION

There is no ROM-DOS command to "undelete" a file. Although utilities exist which can attempt an undelete, certain factors can cause the deleted file to be partially or totally lost. The DEL command should be treated as a permanent delete.

DEL (cont.)

DEL deletes files within a subdirectory, not the subdirectory itself. For subdirectory removal, see the RMDIR command.

Options

/P causes DEL to pause and prompt the user before each file is deleted. This option is most useful when deleting files with wildcards. A sample prompt is shown below:

```
Filename, Delete (Y/N) ?
```

Examples

```
DEL a:*.DOC /P
```

Deletes all files on the a: drive with a .DOC extension. Before each file is deleted the user is prompted to determine if that file should be deleted.

```
DEL MYLETTER.DOC
```

Deletes the file MYLETTER.DOC from the current default subdirectory.

```
DEL *.DOC
```

Deletes all files in the current subdirectory with a .DOC file extension.

DEVICE

CONFIG.SYS

Purpose

A device driver may be installed into ROM-DOS via the DEVICE command.

Syntax

```
DEVICE = device_driver  arguments
```

Remarks

A device driver allows ROM-DOS to access hardware that is not common in all PC's.

The full drive path and file name of the device must be specified. The arguments are different depending on the device driver.

Examples

```
DEVICE=C\BIN\VDISK.SYS  120 /e
```

Installs the ROM-DOS RAM disk driver, VDISK.SYS, via the DEVICE command. There are 120K bytes of extended memory dedicated to the RAM disk in this example.

DEVICEHIGH

CONFIG.SYS

Purpose

Loads an installable device driver into the upper memory area if available.

Syntax

```
DEVICEHIGH = device_driver  arguments
```

Remarks

A device driver allows ROM-DOS to access hardware that is not common in all PC's. A device can be loaded into the upper memory areas if they are available and there is enough free upper memory to accommodate the driver's needs.   To make high memory available, the EMM386.EXE and HIMEM.SYS utilities must be loaded. If these utilities are not loaded or there is not enough upper memory available, the device loads into conventional memory.

The full drive path and file name of the device must be specified. The arguments are different depending on the device driver.

Examples

```
DEVICEHIGH=c:\BIN\MYDEVICE.SYS  /20 /M
```

Installs a driver called MYDEVICE with its command line arguments as specified. The device loads into upper memory if available.

DIR (DIRectory)

Internal

Purpose

Displays a list of the files that are in a directory.

Syntax

DIR [*d:*][*path*][*filename*][*/option*]

Remarks

The DIR command lists all the files in a directory or shows the directory entries of specific files. The standard directory display includes columns for file names, file name extensions, file sizes, and the dates and times the files were created.

Options

/A causes the DIR command to display only the files that match the specified *filespec* and have the given attribute. The following list shows the legal attribute descriptions:

Letter	**Description**
A	Archive ready for archiving
D	Directories
H	Hidden files
R	Read only files
S	System files
X	Show attributes
-	Negates listed attributes with dash “-“ symbol. For example, to select all files that do not have the archive bit set, use /A-A option.

/B (or bare) displays the directory without Volume label, date, time, or size information.

/L displays file names in lowercase.

/P selects page mode, pausing the display each time the screen is full. Press any key for the next screen.

/O displays file names in sorted order. The sort order can contain one or more of the following letters:

Letter	Description
D	By date and time, newest first
E	Alphabetic order by extension
G	Directories grouped before files.
N	Alphabetic order by name
S	Size, smallest first
-	Reverses the sort order with the dash "-" symbol. For example, to sort all files in the directory in reverse alphabetic order, use /O-N option.

/S displays files in subdirectories.

/W displays a list in wide format without date, time, or size.

The DIRCMD environment variable sets the default preferences for the DIR command. The SET command assigns values to an environment variable. Refer to the SET command section for proper usage. For example, if you always want /P set for DIR, use SET DIRCMD=/P. The default settings in DIRCMD can be overridden using the minus sign (-) preceding the option. If you want to cancel the paging for a single use of the DIR command, enter DIR /-P.

The DIRSIZE environment variable is useful for non-standard screen sizes. As with the DIRCMD variable, use the SET command to assign the values. The syntax is:

SET DIRSIZE *rows*[,*columns*]

The values for *rows* and *columns* only have an affect when you use /P or /W with DIR. /P uses *rows* to display the correct number of lines before pausing. /W uses *columns* to display the correct number of columns across the width of the screen.

DIR (cont.)

Examples

```
DIR
```

Displays the directory entries of all files in the current drive and directory.

```
DIR b:\MEMOS
```

Displays all files in the subdirectory MEMOS on drive B.

```
DIR a:\*.RPT /AD/P
```

Displays directory entries for all files in drive A with the extension RPT one screen at a time.

```
DIR /ON
```

Displays all files sorted by file name order.

```
DIR /AH
```

Displays all hidden files.

```
DIR *.DOC /B
```

Displays all files with a .DOC extension without file sizes, or volume labels.

DISKCOPY

External

Purpose

Copies the entire contents of one disk to another.

Syntax

DISKCOPY *<drive1:>* *<drive2:>* [*/option*]

Remarks

The first drive specifies the source disk, and the second drive specifies the target disk.

Disks that may be copied are 360KB, 720KB, 1.2MB, and 1.44MB disks. Both the source disk and the target disk must be of the same type. If the target disk is not correctly formatted, it is reformatted before copying.

If any problem occurs during the copy process, DISKCOPY indicates the side, track, and sector where the problem occurred.

The source or the target disk may not be a RAM or virtual disk.

DISKCOPY only copies floppy disks; hard disks are not copied.

Options

/V verifies the copy after completion.

Examples

```
DISKCOPY a: b:
```

Makes a duplicate of the contents of the disk in drive A onto drive B. Drive B must support the same type of disk as drive A.

DISKCOPY (cont.)

```
DISKCOPY a: a:
```

Makes a duplicate of the contents of the disk in drive A. The user is prompted to swap the source and target disk as needed to perform the entire copy.

DOS

CONFIG.SYS

Purpose

ROM-DOS can be loaded in special memory called the High Memory Area or HMA to free more of the conventional (lower 640K) DOS memory for use by applications.

Syntax

```
DOS=HIGH
```

Remarks

DOS=HIGH frees up more of the standard DOS memory for use by applications.

This command only works on 286, 386, and 486 CPU's with extended memory and Datalight's HIMEM.SYS High Memory Manager, or equivalent. It does not work on standard XT class PC's. Setting DOS=HIGH is ignored when ROM-DOS is in ROM.

DOS=HIGH also works on Chips and Technology's PC/Chip, if it has extended memory and Chips and Technology's HIDOS.SYS device driver, or another memory driver is loaded.

See the HIMEM.SYS device driver description in this manual. See also 386MAX by Qualitas for more information.

Examples

```
DEVICE=HIMEM.SYS
DOS=HIGH
```

Loads the High Memory Area device driver and then loads ROM-DOS into the HMA for increased conventional memory. The high memory area device driver must be loaded first, before DOS=HIGH.

ECHO Batch Subcommand

Internal

Purpose

Controls whether ROM-DOS commands and other messages are displayed during batch file execution.

ECHO also allows you to create your own messages for display.

Syntax

ECHO [ON|OFF]
ECHO *message*

Remarks

ON is the default ECHO setting. It displays commands in a batch file as they are executed by ROM-DOS. Typing ECHO OFF turns off such display. ON switches it back on again.

ECHO alone, typed without ON or OFF, displays the current ECHO setting.

message is a string of characters, such as a warning or a reminder, that you want ROM-DOS to display. Although your message is displayed whether ECHO is on or off, the message display is useful only when ECHO is off.

To create a message, type ECHO followed by your message. If your message is more than one line long, begin each line of the message with the ECHO command

The @ symbol can be used to suppress the echoing of a single command when ECHO is off. Place the @ symbol first on the command line.

Examples

```
ECHO  This batch file moves files
ECHO  to another directory.
```

Displays batch file messages of more than one line.

```
ECHO  OFF
```

Sets ECHO to off.

ERASE

Internal

Purpose

Deletes a specified file or set of files.

Syntax

ERASE [d:][*path*]*filename* [/P]

Remarks

DEL and ERASE are functionally identical.

When no drive is specified, the default drive is assumed. When no path is specified, the default path is assumed.

The wildcards ? and * can be used in the *filespec*. This should be done with caution to avoid deleting multiple files unintentionally.

When the *filespec* *.* is used to delete all files in the specified subdirectory, a verification message is displayed:

```
Are you sure (Y/N) ?
```

Type Y only if you want all files in the specified sub directory to be erased (deleted).

CAUTION

There is no ROM-DOS command to undelete a file. Although utilities exist which can attempt an undelete certain factors can cause the erased file to be partially or totally lost. The ERASE command should be treated as a permanent erase.

ERASE deletes files within a subdirectory, not the subdirectory itself. For subdirectory removal, see the RMDIR command.

Options

/P causes ERASE to pause and prompt the user before each file is deleted. This option is most useful when deleting files with wildcards. A sample prompt is shown below:

```
Filename, Erase (Y/N) ?
```

Examples

```
ERASE MYLETTER.DOC
```

Erases the file MYLETTER.DOC from the current default subdirectory.

```
ERASE *.DOC
```

Erases all files in the current subdirectory with a .DOC file extension.

```
ERASE a:*.DOC /P
```

Erases all files on the a: drive with a .DOC extension. Before each file is erased the user is prompted to determine if that file should be erased.

EXIT

Internal

Purpose

Exits a secondary "nested" ROM-DOS operation, and returns control of the system to the primary program.

Syntax

```
EXIT
```

Remarks

EXIT has no effect if a secondary COMMAND.COM command processor has not been loaded since the primary COMMAND.COM is always loaded in a **permanent** mode. A secondary COMMAND.COM is effected if it is loaded without the /P permanent option.

FCBS

CONFIG.SYS

Purpose

Allows you to specify the number of File Control Blocks (FCBs) open at one time.

Syntax

FCBS = *number* [,*minimum number*]

Remarks

Number specifies the maximum number of FCBs open at any given time. The default for this value is 4. The value for *number* must be in the range from 1 to 255. *Minimum number* specifies the minimum number of FCBs to be open at all times. *Minimum number* has the same default and range value as *number*.

Examples

FCBS = 8, 4

Sets the maximum number of FCBs to 8 and leaves at least 4 open at all times.

FDISK

External

Purpose

Prepares a hard disk for use by ROM-DOS. FDISK allows you to determine the number of logical drives that are available using a single or multiple physical hard disks.

Syntax

FDISK [/*option*]

Options

/D enables debug messages.

/T disables writing to disk, for test purposes.

Remarks

FDISK prompts the user with a command menu as shown below:

```
FDISK v6.22(revision 2.10)
(C) Copyright 1989 - 1995 Datalight

V) View partition(s)
R) Raw display of partition sectors
C) Create DOS partition(s)
N) Create non-DOS partition(s)
D) Delete a partition
A) Delete All partitions
T) Toggle boot status
M) Write Master Boot Code
Q) Quit without saving
S) Save changes (and reboot)

Enter command [V]:
```

Enter the letter that describes the choice you wish to make. If you make a mistake, press ESC and you will move back to the previous menu.

Most users will only want to initialize a hard disk the first time it is set up for use. This is done by using choice C (or possibly N). Choice C displays a description of available disk space and a recommendation for using that available space. Following these recommendations initializes a hard disk with as many 32 Megabyte partitions as possible.

If choice C indicates that the hard disk already has partitions, they may be deleted if needed. The only reason for deleting partitions is to build more smaller-sized partitions, or fewer larger-sized partitions.

The M choice writes out the code for the Master Boot Record. By default, FDISK only writes out the disk partition information. If FDISK finds that the Boot Code is invalid, it automatically writes the boot record. If the existing Boot Code appears valid, it is re-written unless forced by the use of the M option.

After completing all the modifications and/or additions you want, select S to save changes. Once FDISK has modified the disk, the following message appears.

```
Press any key and ROM-DOS will reboot
```

After rebooting, each newly FDISKed drive needs to be formatted prior to use.

Format only those drives that are new or have been resized.
Format destroys any existing data.

If no changes were made with FDISK, or you wish to abandon the changes made, simply select Q to quit without saving.

FILES

CONFIG.SYS

Purpose

The maximum number of files that may be open at one time can be modified using the FILES command.

Syntax

```
FILES = number
```

Remarks

The number of files includes the standard files, *stdin*, *stdout*, *stderr*, *stdprn*, and *stdaux*. The minimum value is 8, the maximum is 255. All other values are ignored.

Example

```
FILES = 10
```

Causes the maximum for open files to be ten.

FIND

External

Purpose

FIND is a filter to display only lines that contain a specified *string*.

The input to FIND may come from a file, or it may be piped in from another filter or a DOS command.

Syntax

FIND [*/option*] *match-string* [*filename*]

Options

/C only displays the count of lines found with the specified *string*.

/N displays the line number of the line found containing the *string*.

/V displays the lines that do not contain the string.

match-string specifies the word or group of characters to search for.

filename specifies the file or group of files to search. The complete drive and path can be specified. Wildcard characters can be used in the file name.

Examples

```
FIND printf junk.c
```

Shows each line in the file JUNK.C containing the *match-string* **printf**.

FIND (cont.)

```
dir | FIND DIR
```

Shows each line in a directory listing containing a DIR. The command first command. The FIND then displays each line containing the DIR *string*.

```
FIND /C ROM-DOS MANUAL.TXT
     MANUAL.TXT: 105
```

Gives a count of the lines in the file MANUAL.TXT containing the string ROM-DOS.

FOR Batch Subcommand

Internal

Purpose

Allows repeated execution of a ROM-DOS command applied to a set of files.

Syntax

FOR *%%variable* IN (set) DO command *%%variable*

Remarks

In execution, this command attaches the *variable* as an identifier to each file in the *set* of files described. It then applies the *command* to each of these identified files. The *set* may be an exact list of complete file names, or a global file specification using wildcard characters.

The FOR subcommand can be used directly on the command line as well as within a batch file. To use on the command line, substitute a single percent (%) symbol for the double percent signs (%%).

Examples

```
FOR %%N IN (Q1.TXT Q2.TXT) DOPRINT %%N
```

Prints only the files Q1.TXT and Q2.TXT.

```
FOR %%N IN (*.TXT) DO PRINT %%N
```

Print all files, in the current default directory, with a .TXT extension.

FORMAT

External

Purpose

Initializes a disk so ROM-DOS can access files on that disk. A disk *must* be formatted before it is usable by ROM-DOS.

Syntax

FORMAT [*drive:*] [*/options*]

Remarks

FORMAT initializes the disk and directory of the specified drive. The size of the formatted disk will be the largest possible size the specified drive supports, unless a different size is specified via a command line option.

Options

/4 formats as a 360 K byte disk even if the drive is a 1.44 or 1.2 Megabyte drive.

/7 formats as a 720 K byte disk even if the drive is a 1.44 Megabyte drive.

/B uses BIOS int 13h calls. By default, FORMAT checks the DOS version and if it is DOS 5.0 or higher, it uses the floppy device driver to do the format. Using /B forces FORMAT to bypass the floppy or hard disk controller and use BIOS calls. /B makes FORMAT device independent.

/C formats one disk without operator input. The disk is assumed to be in the specified drive, and FORMAT exits immediately when the format is complete. This switch is useful in batch files or programs that require a formatted disk without user input.

FORMAT (cont.)

/F*:#* specifies the size of the diskette to be formatted. Available size values are 360, 720, 1.2, 1.44 and 2.88, and are entered as /F:*size*. For example, /F:1.2.

/H causes the system files not to be hidden. This can be used along with /S.

/I forces FORMAT to use IOCTL calls and never use BIOS calls. Normally FORMAT first tries to access the device driver IOCTL calls to format the diskette. If this failed, BIOS calls are used (unless /B is specified, see above). BIOS calls are always used for DOS 3.3 and earlier.

/Q performs a quick FORMAT. A quick format reinitializes the disk, deleting each file and subdirectory from the disk. A quick format can only be performed on a previously fully-formatted disk.

/S copies the ROM-DOS system files (ROM-DOS.SYS and COMMAND.COM) onto the disk. You do not see ROM-DOS.SYS because it is hidden from view.

/V places a volume label on the disk. The user is prompted for the volume label once the format is complete.

If FORMAT encounters an error, the exit code returned to DOS indicates the type of error. The error codes are listed in the following table:

Error Level	**Type of Error**
0	No error encountered
1	Invalid drive
2	Unsupported drive format
3	Attempted hard-drive format (unsupported)
4	Write protect error

Examples

```
FORMAT a:
```

FORMATs the disk in drive A using the default drive parameters.

```
FORMAT b: /S /V
```

FORMATs the disk in drive B and places the system and a volume label on the disk.

```
FORMAT a: /4 /C
```

FORMATs 360K diskette in drive A, without prompting for user confirmation.

GOTO Batch Subcommand

Internal

Purpose

Transfers control to another line of the batch file.

Syntax

```
GOTO label
```

Remarks

The *label* is another line in the batch file consisting of a string up to eight characters long. The label may be an environment variable.

If the specified *label* is not found, then the batch file terminates with the error message **Label not found**.

Examples

```
GOTO MESSAGE
```

Moves the control of execution within the batch file to a line that says:

```
:MESSAGE
```

> A batch file label must be preceded by a colon (:).

HELP

Internal

Purpose

Provides Online Help of each ROM-DOS command.

Syntax

```
HELP <command>
```

Remarks

COMMAND.HLP must be available (i.e. in the path) in order to use this command. If it is not available, an error message occurs, indicating that COMMAND.HLP cannot be found.

HELP serves as a memory aid. For complete information about ROM-DOS commands, always consult this manual.

HELP for each command can also be displayed by entering a "/?" following the command name.

Examples

```
HELP DIR
```

Lists the help of the DIR command.

or

```
DIR /?
```

All available batch file commands are also listed by HELP.

IF Batch Subcommand

Internal

Purpose

Allows conditional execution of commands.

Syntax

IF [NOT] *condition command*

Remarks

The *condition* may be any one of the following:

ERRORLEVEL *number*
string1 == string2
EXIST [*d:*][*path*]*filename*

If the *condition* is true, the *command* is executed; otherwise, the *command* is bypassed and the next command in the batch file is executed. The [NOT] option tests the opposite of any of the conditions.

The ERRORLEVEL *number* is true if the last program to execute had an exit code equal or greater than *number*. Using the [NOT] option with this condition tests whether or not the exit code is less than the *number* argument.

The condition *string1 == string2* is only true when *string1* and *string2* are identical. The strings must match exactly; uppercase/lowercase mismatches are not allowed. Applying the [NOT] option creates a condition that is true only when the strings are not identical.

The EXIST *condition* is true if the specified *filename* is found. Wildcard characters are allowed in the *filename*. The [NOT] EXIST condition is true if the *filename* cannot be found.

IF (cont.)

Examples

```
IF ERRORLEVEL 15 GOTO :EXIT
```

Goes to the :EXIT label if the ERRORLEVEL was equal to or greater than 15.

```
IF %1 == CONFIG.SYS PRINT %1
```

Prints the file stored as the %1 parameter only if its exact name is CONFIG.SYS.

```
IF NOT EXIST OLD COPY CONFIG.SYS OLD
```

If there is not a file named OLD, this command copies CONFIG.SYS to OLD.

INCLUDE

CONFIG.SYS

Purpose

Includes the contents of one configuration block into another. The instructions from the originating instruction block, as well as the included block, are carried out. This command can only be used within a CONFIG.SYS configuration block.

INCLUDE = *blockname*

Remarks

INCLUDE is useful for sets of instructions common to several system configurations. The commands can be defined once in a single configuration block and then inserted into other configuration blocks via the INSERT command. See the “Using Multiple Configurations” for more details.

Examples

Carries out instructions in the configuration block labeled [WORDPROC] if **WORDPROC** was chosen from a CONFIG.SYS. The instructions in the INCLUDED block labeled [MISC] is also implemented as part of the [WORDPROC] block of instructions.

```
.
.
[MISC]
device=mouse.sys
device=c:\netword\loadnet.sys

[WORDPROC]
files=20
buffers=10
set path=c:\bin;c:\wp;c:\dict
INCLUDE=MISC
.
.
```

INSTALL

CONFIG.SYS

Purpose

Loads Terminate and Stay Resident (TSR) programs during CONFIG.SYS processing.

Syntax

INSTALL = [*d:\*][*path*] *TSR_Program* *TSR_Arguments*

Remarks

The TSR program is loaded much the same as if loaded from AUTOEXEC.BAT, except that an environment is not created. The lack of an environment may cause some programs to execute incorrectly. These programs must be loaded from the AUTOEXEC.BAT file.

Examples

```
INSTALL = c:\BIN\MOUSE.COM
```

Loads a mouse driver from CONFIG.SYS using INSTALL. Command line arguments can be included.

KEYB

External

Purpose

Alters the keyboard layout for a different language or nationality.

Syntax

KEYB

KEYB *countryid*

KEYB *countryid,* [*codepage*] [*,keyboard filespec*]

Remarks

KEYB is a resident program (TSR). Running KEYB with no arguments simply shows the current settings of a resident copy of KEYB, if there is one.

countryid is a two-letter code that specifies which country, region, or language should become current.

If no *codepage* is included, KEYB uses the default codepage for the *countryid*. You can specify either the default or the alternate code page for any *countryid*.

keyboardfilespec tells KEYB where to find its data file (KEYBOARD.SYS or KEYBRD2.SYS). If no *keyboardfilespec* is given, KEYB first looks for KEYBOARD.SYS in the current directory and then in the directory containing KEYB.COM.

Currently, keyboard data for all countries except Russia and Czechoslovakia are found in KEYBOARD.SYS. The file KEYBRD2.SYS contains the data for Russia and Czechoslovakia.

KEYB (cont.)

Here is a list of the supported *countryid* codes, along with their default and alternate code pages:

Country	Country Identifier	Code Page	Alternative Code Page
Australia	us	437	850
Belgium	be	850	437
Brazil	br	850	437
Canadian-French	cf	863	850
Czech Republic	cz	852	850
Denmark	dk	850	437
Finland	su	850	437
France	fr	850	437
Germany	gr	850	437
Hungary	hu	852	850
Italy	it	850	437
Japan	---	932	---
Latin America	la	850	437
Netherlands	nl	850	437
Norway	no	850	865
Poland	pl	852	850
Portugal	po	850	860

KEYB (cont.)

Country	Country Identifier	Code Page	Alternative Code Page
Russia	ru	866	437
Spain	sp	850	437
Sweden	sv	437	850
Switzerland	sf	850	437
United Kingdom	uk	437	850
United States	us	437	850
Yugoslavia	yu	852	850

If a copy of KEYB has already been run, it is reconfigured to the new specifications. While KEYB is active, you can switch back to a US layout at any time by pressing <Ctrl><Alt><F1>. You can toggle back to the alternate layout by pressing <Ctrl><Alt><F2>

See “Configuring ROM-DOS for International Use” for more information on KEYB.

Examples

```
KEYB GR
KEYB GR,437
KEYB GR,,c:\TOOLS\KEYBOARD.SYS
```

Establishes a German keyboard layout. The first and third uses code page 850, while the second uses code page 437. In the third case, the KEYBOARD.SYS file is located in the c:\TOOLS directory.

LABEL

External

Purpose

Sets or deletes a disk volume label.

Format

```
LABEL [d:] [volume string]
```

Remarks

The volume label may be up to 11 characters in length. LABEL only uses the first 11 characters of a volume label.

The characters used in a volume label are the same ones used for a file name.

When Label prompts you, enter the new label and press enter to modify the existing label:

```
Volume in drive C is xxxxxxxxxxx

Volume label (11 characters, ENTER for none)?
```

If a volume label has previously been assigned to a disk, and you do not enter a new volume label, the following message prints:

```
Delete current volume label (Y/N)?
```

If the disk did not have a volume label prior to running the LABEL command, the above message does not appear.

Examples

```
LABEL a:
```

Displays the volume label of drive A, if one exists, and allows it to be modified or deleted.

LASTDRIVE

CONFIG.SYS

Purpose

Set the maximum number of drives.

Syntax

```
LASTDRIVE = letter
```

Remarks

letter may be any character between A and Z. It stands for the last drive letter that ROM-DOS can access. The default value for *letter* is E.

The minimum number LASTDRIVE is the number of drives in your computer. If *letter* is less than number of drives in your computer, then the LASTDRIVE command is ignored.

LASTDRIVE is often used to cause ROM-DOS to make more space for non-standard drives that are not in your system. These drives may be CD-ROM drives, Flash disk drives, or network drives.

Example

```
LASTDRIVE = H
```

Sets H as the last drive ROM-DOS can access. If the computer has five drives installed, there is room for three additional non-standard drives.

LOADHIGH

Internal

Purpose

Loads an executable or TSR into the upper memory area if available. LOADHIGH can be run as a batch subcommand or from the DOS command line.

Syntax

LOADHIGH = *executable* [*arguments*]

or

LH = *executable* [*arguments*]

Remarks

An executable or TSR program can be loaded into the upper memory areas if they are available and there is enough free upper memory to accommodate the program's needs. To make high memory available, the EMM386.EXE and HIMEM.SYS utilities must be loaded. If these utilities are not loaded or there is not enough upper memory available, the program loads into conventional memory.

The full drive path and file name of the device must be specified. The arguments are different depending on the device driver.

Examples

```
LOADHIGH=c:\apps\checkit.exe /p
```

Installs an executable called CHECKIT as specified. The program loads into upper memory if available.

MENUCOLOR

CONFIG.SYS

Purpose

Sets the text and background colors for the startup menu. This command can only be used in a menu block within your CONFIG.SYS file.

Syntax

MENUCOLOR = *text_color* [*,background_color*]

Remarks

text_color selects the display color for the screen text. The color numbers 0 to 15 can be selected from the list below for the text color.

background_color is optional. If a value is not entered, the default color 0 (Black) is used. Be sure to specify different colors for background and text, and separate the numbers with a comma. For best results, choose contrasting colors. Only the color numbers 0 to 7 can be used as the background color designation.

For systems whose BIOS does not directly support a video display, such as Datalight's miniBIOS, the standard CONFIG.SYS menu commands are unusable. In order to use these commands, the color number sequence of "0" for *text_color* and the default *background_color*(black), or "0,0" for text and background colors can be selected. These numbers represent a color choice of black text with a black background, an unusable choice for screen viewing. Using the black/black combination in the MENUCOLOR command line tells ROM-DOS to display the startup menu in TTY mode without using BIOS screen/cursor positioning or color changing commands.

Value	Color	Value	Color
0	Black	8	Gray
1	Blue	9	Bright Blue
2	Green	10	Bright Green
3	Cyan	11	Bright Cyan
4	Red	12	Bright Red
5	Magenta	13	Bright Magenta
6	Brown	14	Bright Yellow
7	White	15	Bright White

Examples

```
MENUCOLOR=14,1
```

Displays the menu text in Bright Yellow on a Blue background.

```
MENUCOLOR=5
```

Displays the menu text in Magenta with a default background of black.

MENUDEFAULT

CONFIG.SYS

Purpose

Sets the default menu item choice and a time-out value for making a menu selection. This command can only be used with a menu configuration block in the CONFIG.SYS file.

Syntax

MENUDEFAULT = *blockname*[,*timeout*]

Remarks

blockname specifies the default menu item. The value for *blockname* must match a configuration block name defined elsewhere in your CONFIG.SYS file.

timeout (optional) represents the number of seconds ROM-DOS waits for a user input selection before initializing your system with the default configuration. The timeout period can be set to a value between 0 and 90. If you select 0, the default menu item automatically implements without a wait. If you do not enter *timeout* -, ROM-DOS does not continue until the Enter key is pressed.

If your system BIOS does not support a Video display directly, such as Datalight's miniBIOS, please refer to the MENUCOLOR command for special instructions.

MENUDEFAULT (cont.)

Examples

```
[MENU]
menuitem=Word_Proc, Word Processing
menuitem=Network, Network
menuitem=Research, Research and Development
menucolor=15,1
menudefault=Word_Proc,20
```

Makes the Word_Proc configuration block the default menu item. If the user fails to make a selection within 20 seconds, the Word_Proc block is processed.

MENUITEM

CONFIG.SYS

Purpose

Specifies an item on the startup menu. This command can only be used within a menu configuration block in the CONFIG.SYS file.

Syntax

MENUITEM = *blockname* [*,menu_text*]

Remarks

blockname is a user defined label given to a configuration block defined elsewhere in the CONFIG.SYS file. If a user selects the menuitem, all commands in the selected configuration block are processed, along with the instructions common to all menu choices (denoted by block header [COMMON]). The blockname can be up to 70 characters long and may contain most printable characters, including spaces, backslashes (\), forward slashes (/), commas, semicolons (;), equal signs (=). Square brackets ([]) cannot be used in block names.

menu_text is a descriptive statement that defines the blockname. The menu_text is displayed on the screen as a line item in the startup menu. *menu_text* can be up to 70 characters long and can contain any characters. If *menu_text* is left off, the blockname is used for the startup menu display.

If your system BIOS does not support a Video display directly, such as Datalight's miniBIOS, please refer to the MENUCOLOR command for special instructions.

MENUITEM (cont.)

Examples

```
[MENU]
menuitem=Word_Proc, Word Processing
menuitem=Network, Network
menuitem=Research, Research and Development
menudefault=Word_Proc,20
```

Defines three menu items, Word_Proc, Network, and Research. Each of these has descriptive text and a set of commands defined later in the CONFIG.SYS file.

```
ROM-DOS 6.22 STARTUP MENU


1. Word Processing
2. Network
3. Research and Development

Enter a choice: 1
```

Displays these menuitems in the startup menu at boot time.

MKDIR (Make DIRectory)

Internal

Purpose

Creates a new subdirectory.

Syntax

MKDIR [*d:*][*path*]*subdir*

or

MD [*d:*][*path*]*subdir*

Where *subdir* is the name of the new subdirectory to be created. Note that MD may be used instead of the full MKDIR.

Remarks

If no drive or path is specified, the new subdirectory is created one level below the current default directory.

If drive and/or path is specified, everything specified must exist or the command displays an error message.

Examples

```
MKDIR TEMPDIR1
```

Creates a new subdirectory called TEMPDIR1 within the current default directory.

```
MKDIR c:\UTIL\TOOLS
```

Creates a new subdirectory called TOOLS within the UTIL subdirectory, assuming UTIL already exists.

MODE

External

Purpose

Modifies the operation of the printer, serial port, and active video display.

Syntax

MODE LPT#[:]=COM#[:]

or

MODE COM#:*baud*[,*parity*[,*databits*[,*stopbits*[,P]]]]

or

MODE <*video mode*>

Remarks

The first format above causes line printer output to be redirected to the serial port.

The second format above changes the operation of the specified Communications port. The options that can be modified are listed below. Invalid values are flagged with an error message.

Parameter	**Values**
baud	110, 150, 300, 600, 1200, 2400, 4800, 9600
parity	N - None, O - Odd, E - Even
databits	Either 7 or 8
stopbits	Either 1 or 2 stop bits
P	Printer Port

MODE (cont.)

The final format changes the active video mode for the display terminal. The valid choices for this version of the MODE command are as follows:

Value	Description
40	Indicates 40 characters per line.
80	Indicates 80 characters per line.
bw40	For a color graphics adapter with color disabled and 40 characters per line.
bw80	For a color graphics adapter with color disabled and 80 characters per line.
co40	Indicates a color monitor with color enabled and 40 characters per line.
co80	Indicates a color monitor with color enabled and 80 characters per line.
mono	For a monochrome display. Assumes 80 characters per line.

Using P as the last argument repeatedly sends output to the printer port until successfully received. Without P, output is sent only once causing a critical error if unsuccessful.

A serial port should be initialized before a LPT device is redirected to it.

Examples

```
MODE COM1:9600,n,8,1
```

Modifies the settings for the COM1 device to a baud rate of 9600, no parity, eight data bits, and one stop bit.

```
MODE LPT2:=COM2
```

Redirects the output from LPT2 to the COM2 serial port. All following output to LPT2 actually goes to the COM2 device.

MODE (cont.)

```
MODE mono
```

Indicates a monochrome display adapter.

MORE

External

Purpose

Displays a text file one screen at a time.

Syntax

MORE [*filename*]

or

<*command*> | MORE

Remarks

The input to MORE may come from a file, or it may be piped in from another filter or a DOS command. If the *filename* is present, then the file is viewed; otherwise, MORE reads from the Standard Input.

Once a screen is viewed, a line is displayed on the bottom of the screen indicating the percent of the file that has been viewed. At this point, there are several options for the next lines of text to be viewed:

Option	Description
B	Display the previous full page.
<enter>	Display just one more line.
T	Display starting at the top of the file.
Spacebar	Display the next full page of text.
Q	Quit MORE

MORE (cont.)

Examples

```
DIR | MORE
```

Displays a directory one screen at a time.

```
MORE READ.ME
```

Displays the file READ.ME one page at a time.

NEWFILE

CONFIG.SYS

Purpose

Continues CONFIG.SYS file processing from a new file. The file can be located on in another directory or even on a different drive.

Syntax

NEWFILE=*filename*

Remarks

NEWFILE command is especially useful when the CONFIG.SYS file is located on an inaccessible drive or in ROM. Additional device drivers or instructions can be added easily to the new file and are processed along with the main CONFIG.SYS file upon starting the system.

When the NEWFILE= instruction is processed, control is passed from the present file (the one containing the NEWFILE instruction) to the file specified in the command. Any commands placed after the NEWFILE instruction in the original file are not processed. If the specified *filename* cannot be located, CONFIG.SYS processing is terminated (even if instructions were to be placed after the NEWFILE command) and the remainder of the startup process is completed.

NEWFILE commands can be nested. Your original CONFIG.SYS can call a second set of instructions via the NEWFILE command. The second file can in turn call a third file by using the NEWFILE command, and so on. Be sure that each file name in the successive steps has a unique name; otherwise, an infinite loop is created as control is passed back to the same file repeatedly.

When NEWFILE is used, it is also created as an environment variable, retaining the value assigned by the NEWFILE= statement. This can be used in an AUTOEXEC.BAT file for further boot-up decision-making.

NEWFILE (cont.)

Examples

```
NEWFILE=c:\BIN\NEWCFG.SYS
```

Causes instructions in the file NEWCFG.SYS, located in the c:\BIN directory, to execute as part of the CONFIG.SYS file. The contents of NEWCFG.SYS may include any of the commands listed in this section.

Also, the environment contains the following entry:

```
NEWFILE=c:\BIN\NEWCFG.SYS
```

This can be verified by running the SET command with no parameters.

NUMLOCK

CONFIG.SYS

Purpose

Sets the Num Lock key on the keyboard to ON or OFF when your computer starts.

Syntax

```
NUMLOCK=[on|off]
```

Remarks

Selecting ON designates that the Num Lock key is set to *on* when DOS boots. Selecting OFF designates that the Num Lock will be *off* when DOS boots. In either case, you still have the ability to manually turn the Num Lock key on and off after boot up with the NUMLOCK command.

Examples

```
NUMLOCK=on
```

Sets the Num Lock key to on when the system boots.

PATH

Internal

Purpose

Sets the search path for command files that are not in the current directory.

Syntax

PATH [*d:*][*path*] [;[*d:*][*path*]] ...

Remarks

Without a specified search path, ROM-DOS looks for an external command file (i.e., one with an extension of BAT, COM, or EXE) only in the current directory. The PATH command tells ROM-DOS which other directories to search after searching the current directory.

Typing only the word:

```
PATH
```

by itself displays the current path.

If you want to cancel the command paths you set previously, type:

```
PATH ; or PATH =;
```

Examples

```
PATH c:\UTIL;c:\WP;c:\123
```

Starts application programs on a fixed disk from any drive or directory. Accesses utilities, a word processor, and a spreadsheet in subdirectories c:\UTIL, c:\WP, and c:\123, set the path command.

PAUSE Batch Subcommand

Internal

Purpose

Suspends the execution of a batch file and resumes operation when any key is pressed.

Syntax

```
PAUSE [message]
```

Remarks

A batch job may require that you perform some action such as changing disks, or choosing to continue or terminate the operation. When the command processor encounters PAUSE, it suspends execution and displays the message:

```
Strike a key when ready...
```

After you perform the appropriate action striking any key other than the combinations <Ctrl><C> or <Ctrl><Break> resumes the batch job.

If you press <Ctrl><C> or <Ctrl><Break> at this point, ROM-DOS asks:

```
Terminate batch job  (Y/N)?
```

Responding "Y" ends the batch job. Strategic placement of PAUSE, combined with this query, allows you to divide the batch file into sections so you can end it at some intermediate point.

message displays a reminder on the screen during the pause. Your message precedes the "Strike a key" message. Note, however, that your message appears only if ECHO is off.

PAUSE (cont.)

Examples

```
PAUSE Place blank disk in drive a:
```

Alerts the user of the need for a disk and suspends operation until a key has been hit.

PRINT

External

Purpose

Prints a single file or a list of files.

Syntax

```
PRINT [/d:] [filename] [/options]
```

Remarks

PRINT allows you to enter between one and 32 files for spooling to the printer. The files are output to the device in a spooled manner (while the user performs other operations).

If PRINT is entered without any parameters, it displays all the files in queue.

The first time PRINT is used, the operator is prompted for the device to perform the operation:

```
Name of list device [PRN]:
```

The legal devices for printing are LPT1, LPT2, LPT3, LPT4, COM1, COM2, COM3, COM4, AUX, or PRN.

Options

/B sets the buffer size. The default buffer size is 512 bytes. A larger buffer size causes print to operate faster. The maximum buffer size is 32k bytes and the minimum size 256 bytes. This option is only allowed the first time PRINT is run.

/C cancels only the file names listed after /C.

PRINT (cont.)

/F sets the maximum number of files to be queued at one time. The default number of files is 10. The minimum is 2 and the maximum is 32. Support for more files is often useful when using wild cards in file names. This option is only allowed the first time PRINT is run (or until the next system reboot).

/P causes all files listed after this option to be submitted for printing. This is the default for file names encountered on the PRINT command line.

/T cancels all the files from the print queue. Think of this as a terminator.

/H displays the help screen.

Examples

```
PRINT FILE1.TXT FILE2.TXT FILE3.TXT
```

Puts three files into the print queue. The first file starts printing after the command ends.

```
PRINT /C FILE2.TXT
```

Removes FILE2.TXT from the print queue. All other files in the queue prints normally.

```
PRINT /T
```

Cancels all files in the print queue. Printing may continue for a short time because of the buffer in your printer.

PROMPT

Internal

Purpose

Changes the ROM-DOS command prompt.

Syntax

PROMPT [*text*] [*$character*] [*$character*...]

Remarks

The prompt ROM-DOS normally displays is the letter of the current drive followed by a right arrow (>). Using PROMPT changes the prompt to include any combination of a message, the current directory, the date, the time, and some other features.

Code	Corresponding Prompt
$T	Time
$D	Date
$P	Current Path
$V	ROM-DOS version number
$N	Current drive
$G	The > character
$L	The < character
$B	The | character
$Q	The = sign
$H	A backspace (which erases the previous character)
$E	ASCII code for Escape (X'1B')
$_	Start a new line
$$	The $ character

PROMPT (cont.)

Examples

To show this prompt:

```
Current directory is drive:\path;
Ready for command>
```

You would type in:

```
PROMPT Current directory is $P;$_Ready for command$G
```

To show on separate lines the date, time, and current directory followed by the greater-than character and a space, type:

```
PROMPT $D$_$T$_$P$G<space>
```

Where <space> refers to pressing the spacebar once. The prompt looks like:

```
Mon 6-26-1989
10:17:45.99
a:\> _
```

REM (REMark)

Internal

Purpose

REM has two purposes: to allow comments in a batch or CONFIG.SYS file, and to temporarily disable a command without physically deleting the command from the file. See also the (;) command.

Syntax

REM [*message*]

Remarks

REM provides information for the user but has no effect on the execution of the batch file.

The comment may be made up of any set of characters. A blank line can also be created by omitting the *message* portion of the line.

REM can also be used to temporarily disable a command in a batch file or CONFIG.SYS, without having to delete the line from the file. For CONFIG.SYS files, the semicolon (;) can also be used in place of the REM command.

Examples

```
REM  This batch file created by
REM  T.J. Sherrill
```

Adds at any point in a batch file as user information only.

```
DEVICE=HIMEM.SYS
DOS=HIGH
REM DEVICE=TESTDEV.SYS /P
```

Temporarily removes DEVICE=TESTDEV.SYS from CONFIG.SYS instructions. This statement is not processed again until REM is removed.

REN (REName)

Internal

Purpose

Changes the name of a file.

Syntax

REN [*d:*][*path*]*filename1 filename2*

Remarks

REN renames files within a directory; it does not move a file to a different drive or directory as part of the command.

The wildcard characters * and ? can be used to rename more than one file at a time.

ROM-DOS does not allow you to give a file a name that matches the name of an existing file in the same directory.

Examples

```
REN b:NOTES.DOC REPORT.DOC
```

Renames the file NOTES.DOC in drive B to REPORT.DOC.

```
REN *.DOC *.TXT
```

Assigns the extension TXT to all files with the current extension DOC.

RMDIR (ReMove DIRectory)

Internal

Purpose

Removes (deletes) a specified subdirectory.

Syntax

RMDIR [*d:*][*path*]*subdir*

or

RD [*d:*][*path*]*subdir*

Where *subdir* is the name of the subdirectory being deleted. Note that RD may be used instead of the full "RMDIR."

Remarks

If no drive or path is specified, RMDIR looks for the specified *subdir* one level below the current default directory.

If a drive or path is specified, everything specified must exist or ROM-DOS displays an error message.

RMDIR does not remove a subdirectory unless it is empty. An error message is displayed if you attempt to remove a subdirectory that still has files or other subdirectories within it.

Examples

```
RD TOOLS
```

Removes the TOOLS subdirectory from the current directory, assuming TOOLS is an empty directory.

SET

Internal

Purpose

Sets, displays, or removes environment variables.

Syntax

SET [*variable* = [*string*]]

Remarks

Environment variables can be used to control the behavior of programs and batch files and also the behavior of ROM-DOS. This command can be used in the AUTOEXEC.BAT and CONFIG.SYS files and on the DOS command line. The environment variables defined with the set command include, but are not limited to, PATH, COMSPEC, PROMPT, and user defined variables.

Using SET *variable* = with no argument string clears the current environment string for the named variable.

Examples

```
SET PROMPT = $p$g
```

Sets the prompt, although the prompt can also be set with the PROMPT command.

```
SET PROMPT =
```

Clears any previously set prompt settings and returns the prompt to its default state.

SHARE

External

Purpose

Installs the capabilities for file-sharing and file-locking on your hard disk.

Syntax

```
SHARE [/L:nn][/u]
```

Or from CONFIG.SYS:

```
INSTALL=[d:][path]SHARE.EXE [/options]
```

Remarks

SHARE is most commonly used in a network or multi-tasking environment where file sharing is necessary. When SHARE is loaded, DOS utilizes the SHARE utility to validate read and write requests from application programs and users.

/L:*locks* specifies the maximum number of files that can be locked at one time. The default number of files is 20.

/U unloads the share utility and frees the memory. SHARE does not unload if other TSR's have been loaded on top of it. The other TSR's must be unloaded first before trying to unload SHARE.

Examples

```
SHARE
```

Loads the SHARE program from the command line:

SHARE (cont.)

```
INSTALL=c:\UTILS\SHARE.EXE /l:30
```

Installs SHARE from the CONFIG.SYS file and changes the maximum number of locked files to 30:

```
SHARE /U
```

Unloads SHARE and frees the used memory.

SHELL

CONFIG.SYS

Purpose

Specifies a different boot program other than the default COMMAND.COM. **ROM-DOS** boots this new program, with arguments, instead of the one specified internally.

Syntax

SHELL = *boot_ program arguments*

Remarks

SHELL is most often used to start the initial copy of COMMAND with special parameters. One parameter is used for providing a larger environment than the default 128 bytes.

The *boot_program* can be any executable program. The full path, including drive letter, should be specified if the program is not in the root directory of the default drive.

Arguments are optional and program specific. They vary depending on the *boot_program* being executed by the SHELL command.

Examples

```
SHELL=c:\COMMAND.COM /E:512 /P
```

Boots the standard Command Processor but sets the environment space to 512 bytes (up from the default 128). The /P parameter tells COMMAND that it is permanent (cannot terminate).

```
SHELL = c:\TEMP\MYPROG.EXE
```

Boots a program called MYPROG.EXE, located in the directory TEMP, instead of the standard Command Processor.

SHIFT Batch Subcommand

Internal

Purpose

Moves each replaceable parameter for a batch file one position to the left. Execution of the SHIFT command allows use of more replaceable parameters in a batch file--beyond the standard set of %0 through %9.

Syntax

```
SHIFT
```

Remarks

SHIFT moves the string or value stored for each replaceable parameter one position to the left. Upon execution of SHIFT, the %0 argument assumes the value of the %1 argument, the %1 argument then assumes the value of the %2 argument, and so on.

Examples

Command line argument:
```
TYPEIT autoexec.bat config.sys net.bat
```

TYPEIT.BAT batch file:
```
     :repeat
     if EXIST %1 goto doit
     goto end

     :doit
     type %1
     pause
     shift
     goto repeat

     :end
     @echo All Done
```

Reads in a list of files (provided as arguments on the command line) and displays each one to the screen. After displaying each one, the SHIFT command copies the next file in the argument list into the %1 slot, verifies the existence of the file and continues.

SORT

External

Purpose

Sorts a text file and displays the output to the standard device.

Syntax

```
SORT [/options]  [filename]
```

Remarks

SORT normally starts its comparisons at the first character in a line.

The input to SORT may come from a file or it may be piped in from another filter or a DOS command.

Options

/+n causes SORT to begin its alphabetical sorting starting at the nth position in the string.

/r causes SORT to sort in the reverse alphabetical order.

Examples

```
SORT NAMES.LST
```

Sorts the file NAMES.LST and displays the output to the screen.

```
DIR | SORT /+14 | MORE
```

Produces a directory and then sorts the directory by file size (the file size in a directory display starts on the 14th position each line or string). The output display is shown one screen at a time by directing the output from SORT into the display utility MORE.

STACKS

CONFIG.SYS

Purpose

Allows for the use of dynamic data stacks to handle hardware interrupts. ROM-DOS does not utilize this command, although it can be added to a CONFIG.SYS file without error. Using the STACKS command has no effect on the number or size of stacks available.

Syntax

STACKS = *number,size*

SUBMENU

CONFIG.SYS

Purpose

Defines a menu item that represents a secondary menu when selected. This command can only be used within a menu configuration block in the CONFIG.SYS file.

Syntax

SUBMENU=*blockname*[,*menu_text*]

Remarks

blockname defines the name of the secondary menu block of commands. The block menu must be defined elsewhere in the CONFIG.SYS file; otherwise, ROM-DOS leaves this item off of the startup menu. The label can be up to 70 characters long and can contain most printable characters, including spaces, backslashes (\), forward slashes (/), commas, semicolons (;), and equal signs(=). Square brackets ([]) cannot be used in block names.

menu_text (optional) specifies the text that ROM-DOS displays for this menu item on the startup menu. If left out, ROM-DOS displays *blockname* as the text. *Menu_text* can be up to 70 characters long and can contain any character.

The submenu can be defined with any user-provided descriptive label. It need not have the [MENU] label.

SUBMENU (cont.)

Examples

```
[MENU]
menuitem=Word_Proc, Word Processing
menuitem=Network, Network
submenu=Research, Research and Development
menucolor=15,1
menudefault=Word_Proc,20

[WORD PROC]
    files=10
    buffers=10
    lastdrive=m
    device=c:\network\loadnet.sys

[NETWORK]
    include=Word_Proc
    numlock=off

[RESEARCH]
    menuitem=proj1, Project 1
    menuitem=proj2, Project 2
    menudefault=proj1

[PROJ1]
    files=50
    buffers=25
    numlock=on

[PROJ2]
    files=10
    buffers=20
    device=vdisk.sys 64 /e
    numlock=off
```

Defines as one of the startup menu choices. If you were to select Research and Development from the first menu, a secondary menu is displayed offering the choices of Project 1 and Project 2. The actual commands for Project 1 and Project 2 are defined in the configuration blocks labeled PROJ1 and PROJ2.

SWITCHES

CONFIG.SYS

Purpose

Allows special CONFIG.SYS file options.

Syntax

```
SWITCHES=[/k][/n][/f]
```

Remarks

/k makes an enhanced keyboard behave like a conventional style keyboard.

/n prevents the use of the F5 and F8 function keys to bypass the startup commands.

/f instructs ROM-DOS to skip the delay after displaying the "Starting ROM-DOS..." message at boot time. The delay allows the user time to use the F5 and F8 options to alter the processing of the startup files.

Examples

```
switches = /n
```

Prevents the user from using the F5 and F8 keys at boot time.

SYS (SYStem)

External

Purpose

Copies the ROM-DOS system files ROM-DOS.SYS and COMMAND.COM from the disk in the default drive to the disk in the specified drive.

Syntax

```
SYS <drive:> [/options]
```

Remarks

Use SYS to transfer the ROM-DOS system files to a floppy diskette or hard disk. The disk can be a formatted blank disk or can already contain files. With SYS, it is not necessary for the system files to be the first files on the disk. The only requirement is that there is enough contiguous free space on the disk for the new system files to be placed. The command processor, COMMAND.COM, is also transferred to the diskette. The command processor does not have to be copied into the same contiguous space as the system files. There must be enough free space available for the file somewhere on the disk.

If the disk already had system files on it, they are deleted after installing the new system files.

Options

/C tells SYS to not confirm before transferring system.

/H tells SYS to **not** hide the newly transferred system files on the destination disk.

Examples

Copies the ROM-DOS system files from drive A to drive B when you type at the A> prompt:

```
SYS b:
```

TIME

Internal

Purpose

Displays the current time as shown on the system's internal clock and allows resetting of the clock.

Syntax

TIME [*hh*:*mm*:*ss*] [pm|am]

Remarks

The time set by this command is used for time stamping your file revision dates. This information is displayed when you execute a directory listing of your files.

If your computer does not have an internal battery-operated clock, you may want to include the TIME command in your AUTOEXEC.BAT file to set the date at bootup.

The format of the time command is also dependent on the Country specified in CONFIG.SYS. The time is displayed according to local standards for the specified country.

Also see the DATE command.

If you just want to check the time maintained by ROM-DOS, type the TIME command by itself. ROM-DOS displays something like:

```
Current time is 3:00:02.48p

Enter new time:_
```

after which you press <Enter> to return to an empty command line.

If you want to change the time, you can include the time you want on the prompt line after the word TIME. Or you may type the command with no option (as you do to check the time) and enter the new time before pressing <Enter>.

TIME (cont.)

ROM-DOS displays the time according to the 24-hour clock with the a or p indicator to show AM or PM. The AM / PM indicator can be typed as “a” or “p” or as “am” or “pm”. The time may be entered in a 24 hour format or a 12 hour format with the AM or PM designator.

The allowed options for hours and minutes are:

hh = 0-24   *mm* = 0-59   *indicator* = a, p, am, or pm

ROM-DOS displays time to hundredths of seconds. When entering time, however, you do not need to enter seconds or hundredths; ROM-DOS assumes a value of zero if they are not specified.

You may skip the display and prompting by simply typing the current time after the word TIME on the command line:

```
TIME 23:24
```

ROM-DOS accepts your entry as the current time.

Examples

Enters the time 11:15 pm.

```
TIME 23:15
```

or

```
TIME 11:15 p
```

Enters the time as 9:26 am.

```
TIME 9:26
```

or

```
TIME 9:26 am
```

TREE

External

Purpose

Displays each subdirectory and optionally the files within them for a specified drive.

Syntax

```
TREE [d:] [/options]
```

Remarks

TREE displays the full path of each subdirectory on a specified disk.

[*d:*] specifies the drive that TREE displays the subdirectories from. This argument must be specified.

Options

/F causes TREE to display the files in each subdirectory.

Examples

```
c:\DATA> TREE c:
```

Displays all subdirectories on drive C.

```
c:\DATA> TREE a: /F
```

Displays all subdirectories on drive a: along with the files within each sub directory.

TYPE

Internal

Purpose

Displays the contents of a text file on the screen.

Syntax

```
TYPE [d:][path]filename
```

Remarks

If a file containing formatting codes or other non-alphanumeric characters is displayed with TYPE, you will see unintelligible characters and possibly hear beeps. This does not harm the system.

Examples

```
TYPE a:AUTOEXEC.BAT
```

Displays the AUTOEXEC.BAT file on drive A.

VER

Internal

Purpose

Displays the version number of ROM-DOS in use and allows revision of this version number.

Syntax

VER [*n.nn*] [/R]

Remarks

If a new version number is specified, two digits after the decimal are required. This command revises only the record of the DOS version number; it does not change the actual Operating System loaded in the computer.

The version command shows both the version of the VER command itself and the version of DOS in operation.

Options

/R shows the full version and release number of ROM-DOS.

Examples

VER 5.0

Changes the record of current DOS version in use to DOS 5.0. Any programs that are executed, following this command, recognize that DOS 5.0 is running.

VERIFY

Internal

Purpose

Displays or modifies the VERIFY state.

Syntax

```
VERIFY [ON | OFF]
```

Remarks

VERIFY does not perform any data verification (same as the COPY /V option). It is included to provided batch file compatibility.

VERSION.SYS

Config.sys

Purpose

Modifies the version number that ROM-DOS reports. This device performs the same function as the Internal VER command. The difference is that this one allows the change to occur during config.sys processing so that version specific device drivers can load properly.

Syntax

```
DEVICE=VERSION.SYS n.nn
```

Remarks

The new version number, *n.nn,* should be specified with two digits after the decimal. This command revises only the record of the DOS version number; it does not change the actual Operating System loaded in the computer.

The version number change can be verified after booting using the VER command.

Examples

```
DEVICE=VERSION.SYS 5.0
```

Changes the record of current DOS version in use to DOS 5.0. Any programs that are executed, following this command, recognize that DOS 5.0 is running.

VOL

Internal

Purpose

Displays the volume label on a specified disk.

Syntax

VOL <*drive:*>

Remarks

If you do not specify a Drive, then the default drive is assumed. VOL does not allow the setting of volume labels. Refer to the LABEL command for instructions on setting the volume labels.

Examples

```
A>VOL
```

Causes ROM-DOS to display the volume label on the default drive which is the a: drive.

```
A>VOL c:
```

Causes ROM-DOS to display the volume label on the c: drive.

XCOPY

External

Purpose

Copies multiple files and optionally subdirectories from one disk to another.

Syntax

XCOPY [*source*] [*target*] [*/options*]

Remarks

XCOPY is used for copying multiple files and subdirectories, if they exist.

The *source* and the *target* parameter are complete drive path and file specification descriptions. If you do not specify a path, XCOPY assumes the default path. If a file name is not specified, then *.* is assumed.

ATTRIB may be used to modify the archive bit for the various XCOPY options that check the archive status of files. Refer to the ATTRIB command description for instructions.

Options

/A copies only source files having the archive bit set in them. The archive is not reset.

/D<mm-dd-yy> causes XCOPY to copy only those files with a date later than the date specified in /D.

/E causes XCOPY to create subdirectories on the target even if they are empty.

/M causes XCOPY to copy only those source files having the archive bit set. Once the source file is copied; the archive bit is reset.

XCOPY (cont.)

/P causes XCOPY to prompt before each file is copied. The prompt appears as follows:

```
c:\COMMAND.COM (Y/N)?
```

If a Y is entered then the file is copied; otherwise, the file is not copied.

/S causes XCOPY to copy files in sub-directories of the source directory.

/V causes XCOPY to verify each write to the disk.

/W causes XCOPY to wait before starting to copy files. The following prompt is displayed.

```
Press any key to begin copying file(s)
```

Examples

```
XCOPY \bin\*.exe a: /a
```

XCOPY all files in the BIN subdirectory to the a: drive that have an EXE extension and that have the archive bit set.

INDEX