

# Datalight ROM-DOS<sup>TM</sup>

**Single User Version**

Printed: July 2003

## Datalight **ROM-DOS™** Single User Version

Copyright © 1999-2003 by **Datalight, Inc.**

All Rights Reserved.

**Datalight, Inc.** assumes no liability for the use or misuse of this software. Liability for any warranties implied or stated is limited to the original purchaser only and to the recording medium (disk) only, not the information encoded on it.

U.S. Government Restricted Rights. Use, duplication, reproduction, or transfer of this commercial product and accompanying documentation is restricted in accordance with FAR 12.212 and DFARS 227.7202 and by a license agreement.

THE SOFTWARE DESCRIBED HEREIN, TOGETHER WITH THIS DOCUMENT, ARE FURNISHED UNDER A SEPARATE SOFTWARE OEM LICENSE AGREEMENT AND MAY BE USED OR COPIED ONLY IN ACCORDANCE WITH THE TERMS AND CONDITIONS OF THAT AGREEMENT.

Datalight® and ROM-DOS™ are registered trademarks of Datalight, Inc.

FlashFX® is a trademark of Datalight, Inc.

All other product names are trademarks of their respective holders.

Part Number: 3010-0200-0513

# Contents

---

<b>Chapter 1, ROM-DOS Introduction .....</b>	<b>1</b>
About ROM-DOS .....	1
Conventions Used in this Manual.....	1
Terminology Used in this Manual .....	1
Disks and Disk Drives.....	2
Recommended Texts .....	3
<b>Chapter 2, ROM-DOS Installation .....</b>	<b>5</b>
Installing and Running ROM-DOS .....	5
Development System Installation Procedure .....	5
Installed Files .....	5
ROM-DOS Considerations .....	6
<b>Chapter 3, ROM-DOS Basics .....</b>	<b>9</b>
Files, Directories, and Command Line Entries .....	9
Naming Files .....	9
Tree-Structured Directory System .....	10
Using Wildcard Characters without Long Filename Support .....	11
Using Wildcard Characters with Long Filename Support .....	12
System Prompt .....	13
Command Line .....	13
Redirecting Input and Output .....	14
Using Batch Files .....	14
Batch Filenames .....	15
Creating a Batch File .....	15
Batch File Command Line Parameters .....	15
Batch File Commands.....	16
ROM-DOS Command Summary .....	17
<b>Chapter 4, ROM-DOS Configuration.....</b>	<b>21</b>
Basic Configuration .....	21
Using Multiple -User Configurations.....	21
Extending Menu Items to AUTOEXEC.BAT .....	23
Bypassing CONFIG.SYS and AUTOEXEC.BAT Commands .....	24
Stepping Through CONFIG.SYS and AUTOEXEC.BAT Commands.....	24
Environment Variables .....	25
Configuring ROM-DOS for International Use.....	25
Changing Country Conventions.....	26
Displaying Different Code Pages .....	26
Printing Different Code Pages .....	27
Changing the Keyboard Layout .....	27
Configuring Your System: an Example .....	28
<b>Chapter 5, ROM-DOS Internal Commands .....</b>	<b>31</b>
Internal Command Descriptions.....	31

---

<b>Chapter 6, ROM-DOS Utility Descriptions .....</b>	<b>69</b>
ROM-DOS Utilities .....	69
Command Descriptions.....	69
Mini-Command Interpreter .....	112
External Commands .....	113
Internal Commands .....	113
Configuring the Mini-Command Interpreter.....	114
<b>Chapter 7, ROM-DOS Server and Client Applications .....</b>	<b>117</b>
Serial File Server.....	117
Server Program.....	117
Client Program .....	118
Remote Disk Program.....	119
Server Program.....	119
Client Program .....	119
Using the Remote Disk.....	120
Unloading the Server Remotely .....	120
<b>Chapter 8, ROM-DOS Keyboard Layouts .....</b>	<b>121</b>
Keyboard Layouts.....	121
Canada .....	121
Denmark .....	121
Finland .....	122
France.....	122
Germany .....	122
Italy .....	123
Norway .....	123
Spain.....	123
Sweden.....	124
United Kingdom.....	124
United States.....	124
<b>ROM-DOS Glossary .....</b>	<b>125</b>
<b>Index.....</b>	<b>129</b>

# *Chapter 1, ROM-DOS Introduction*

---

## About ROM-DOS

ROM-DOS is a disk operating system that can be loaded in Read Only Memory (ROM) and can run entirely from within ROM and also from a hard or floppy disk, such as in a desktop system. ROM-DOS is functionally equivalent to other brands of DOS and can run programs that are executable under a standard DOS (which executes from RAM).

### **Conventions Used in this Manual**

This manual uses several notation conventions to denote specific actions and types of information.

- Key combinations, where two or more keys must be pressed simultaneously, are shown with a plus sign. For example, Shift+F1.
- Command line entries and displayed messages are shown in this font. For example,  
`DEL MYLETTER.DOC`
- Command line entries indicated by italicized lowercase letters represent information that you supply. For example,  
`DEL filename`  
indicates a need to enter the name of the file to be deleted.
- Command line entries enclosed within square brackets represent optional information. For example,  
`ECHO [message]`  
indicates the *message* portion may be omitted from the ECHO command.
- Command line entries that include mutually-exclusive options separate those options with a vertical bar (|). In the following example, anything more than the BREAK command must be either the ON or OFF.  
`BREAK [ON|OFF]`
- You must press the Enter key for ROM-DOS to accept your command line data. The command line entries shown in this manual do not show the Enter key.

### **Terminology Used in this Manual**

Computer files are stored in the computer's internal memory (RAM and ROM) or on magnetic media, typically disks. Regardless of where the files reside, all information is stored in bytes. A byte can store a single character of data. The following terms describe the size of memory, files, and disk space:

**KB** (kilobyte) – One kilobyte equals 1024 bytes, although the number is often rounded to one thousand bytes.

**MB** (megabyte) – One megabyte equals 1,048,576 bytes but is usually thought of as one million bytes.

**GB** (gigabyte) – One gigabyte equals 1,073,741,824 bytes but usually is thought of as one billion bytes.

**TB** (terabyte) – One terabyte equals 1,099,511,627,776 bytes but usually is thought of as one trillion bytes.

### ***Random Access Memory (RAM)***

RAM can be written to, read from, erased, and rewritten. RAM is your computer's electronic workspace during operation. RAM is also called volatile memory. Its' storage ability is temporary, only holding information while the power is on. When the power is turned off or interrupted, everything stored in RAM is lost.

Within limits, you can change the amount of RAM in a system. Typically, embedded systems have 512KB, 640KB, or 1MB of RAM. Greater amounts of RAM are also possible. In desktop systems, 16MB to 128MB are common.

### ***Read Only Memory (ROM)***

ROM is more permanent than RAM. Data is programmed into a ROM device before the device 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 are classified into two basic groups; rotating media such as floppy disks and hard disks, and memory disks such as those formatted in RAM, ROM and flash memory.

### ***Floppy Disks***

A floppy disk is a disk-shaped piece of magnetic material much like audio recording tape. The information is stored in concentric tracks that are subdivided into sectors. Typical storage space on the 3.5-inch disk is 1.44MB.

### ***Hard Drives***

Hard drives work similar to floppy disks but are fixed in the computer chassis and have a much higher storage capacity. ROM-DOS 6.22 is capable of utilizing hard drives of up to 8GB and ROM-DOS 7.1 is capable of utilizing hard drives of up to 2TB.

### ***RAM Disks***

Portions of RAM can be made to behave like disk drives, complete with tree-structured directories. When a disk drive is created in RAM, it can be read from and written to in the same manner as the physical disk media. However, when system power is interrupted, all information on the drive is lost, unless the drive is formatted on static RAM.

### ***ROM Disks***

Portions of ROM can be made to behave like disk drives, complete with tree-structured directories. ROM drives differ from RAM drives in that they are written to only once. Thereafter, they can only be read from, much like a write-protected floppy disk. In addition, ROM disks are non-volatile, in that the information is not lost when power is lost.

### ***Flash Memory Disks***

Both PCMCIA cards (PC cards) and on-board (resident) flash arrays can serve as disk drives when used with flash file system software such as Datalight's FlashFX for on-board flash memory. Unlike ROM disks, flash memory disks support both the reading and writing of data.

## **Recommended Texts**

The following are all excellent texts on the topic of DOS and its internal structures. The knowledge in them applies to Datalight's ROM-DOS as well as most other DOS compatibles. These all come highly recommended by Datalight, Inc.

1. Microsoft MS-DOS Programmer's Reference  
Microsoft Press, 1993  
ISBN 1-55615-546-8
2. DOS Programmer's Reference, 4<sup>th</sup> Edition  
by Terry Dettmann, Que Corporation, 1993  
ISBN 1-56529-150-6
3. DOS Internals  
by Geoff Chappell, Addison-Wesley 1994  
ISBN 0-201-60835-9
4. Undocumented DOS, 2<sup>nd</sup> Edition  
by Andrew Schulman, Addison-Wesley, 1993  
ISBN 0-201-63287-X
5. PC Interrupts, 2<sup>nd</sup> Edition  
by Ralf Brown & Jim Kyle, Addison-Wesley 1994  
ISBN 0-201-62485-0





## ***Chapter 2, ROM-DOS Installation***

---

### **Installing and Running ROM-DOS**

This chapter provides an overview of the process for installing ROM-DOS on a development system and running on a target platform. To create a version of ROM-DOS that runs on any system, follow these steps (described in detail in subsequent chapters of this manual).

- Select the hardware platform.
- Select the BIOS.
- BUILD (configure) ROM-DOS for that system.
- Create a ROM disk (required only for *diskless* systems).
- Program PROMs and install them (*diskless* systems) or sys the boot drive, and reboot.

### **Development System Installation Procedure**

The development system is used to install the ROM-DOS files from the distribution CD-ROM and then configure and build ROM-DOS to your target system specifications as described in the following chapters.

To install the ROM-DOS SDK software on your Windows development system:

1. Place the CD-ROM in CD drive.
2. Choose Start/Run and type **setup**
3. Follow the onscreen instructions presented by the setup program to install the required components.

We strongly recommend you install all Datalight software under its own directory branch. We suggest you accept the default branch name (DL), which corresponds with the examples in this manual.

You should also install the enclosed Datalight Software Developer's Tool Kit (SDTK) in the same manner. When the installation is complete, modify your path statement to include the C:\DL\DEVTOOLS\BIN directory. This path allows the ROM-DOS BUILD utility to locate all necessary tools.

### **Installed Files**

When installation of the ROM-DOS SDK and the SDTK software is complete, ROM-DOS and the Tool Kit software exist on your development system in the following directories/folders.

```

<root>
|
+-----<DL>
|
+--- ROM-DOS
|   +--- DEVSRC
|   +--- MEMDISK
|   +--- MINICMD
|   +--- UTILS
|   +--- RXE
+--- DEVTOOLS
|   +--- BIN
|   +--- HELP
|   +--- INCLUDE
|       +--- ...
+--- LIB
|       +--- ...

```

**Default Directory Structure**

## **ROM-DOS Considerations**

ROM-DOS runs on most x86 hardware platforms, including the 186, 286, 386, 486 or Pentium, as well as the NEC V-series and the growing number of work-alike processors. Companies such as NEC and Vadem offer all the parts of a PC in a single piece of silicon. ROM-DOS contains no hardware dependencies.

Selecting the BIOS to use depends on the availability of a BIOS for the hardware and the ROM space requirements for the BIOS. When selecting the BIOS, consider not only the availability, but the features of the BIOS. The BIOS should recognize the basic hardware you intend to have available in your system. It is generally the responsibility of the BIOS to support PCMCIA cards (at least SRAM cards) through the BIOS disk Int 13h. The BIOS should also support extended memory on those CPUs that can address RAM beyond 1MB.

Building (or more appropriately, customizing) ROM-DOS for any given system requires a minimum of effort. Building ROM-DOS can be as simple as answering a few prompts that the BUILD program displays and then placing ROM-DOS on disk or in ROM. Chapter 4 is devoted to running the BUILD program.

When building ROM-DOS, consider such issues as which drive letters you want associated with the disk drives. Will drives A: and B: be assigned to the floppy drives, as ordered on a desktop system? Or will the ROM disk be drive A:, a PCMCIA disk drive B:, followed by floppy and hard disks? ROM-DOS makes no restrictions in this area, so choose what best suits your system. You may need to adjust the file SYSGEN.ASM described under “Ordering Floppy and Hard Disk Drives” in the ROM-DOS Developer’s Guide to order the drive letters.

If a ROM disk is needed, either for booting, or as another disk on the ROM-DOS system, use the ROMDISK.EXE program to create this disk. A ROM disk driver that searches memory for the ROM disk is built into ROM-DOS.

If ROM-DOS will be placed in ROM, the final step is to program the ROMs and place them in the target machine. At this point, the system can be booted using ROM-DOS.

Systems running ROM-DOS directly from a hard or floppy disk do not need to program any ROMs. Simply run FORMAT and/or SYS on the disk and reboot.



## ***Chapter 3, ROM-DOS Basics***

---

### **Files, Directories, and Command Line Entries**

#### **Naming Files**

A file is a defined set of related information that your computer stores electronically. A file may be stored on a floppy disk (also called a floppy), on a hard drive, on a CD, or may reside in computer memory (RAM or ROM). To maintain control of interaction between various computer files, each must have its own name that both you and the computer can recognize.

#### ***Long Filenames***

Files used in the ROM-DOS environment, with long filename support enabled, can have up to 260 characters including the file extension and path. Each long filename also has a standard 8.3 filename associated with it. Long filenames can use nearly any character except the following symbols:

Output redirection	>	Question mark	?
Input redirection	<	Asterisk	*
Backslash	\	Double quotes	“
Forward slash	/	Pipe	

When using a long filename or path that contains spaces, the name and/or path must be surrounded by double quotes. For example:

```
copy "c:\my directory\my file.doc" "c:\another directory\reference"
```

#### ***Standard 8.3 Filenames***

Files used in the ROM-DOS environment, without long filenames support enabled, have two-part names separated by a period. The first part is the filename; the second part is the filename extension. For example, the command interpreter file provided with ROM-DOS is named COMMAND.COM, where COMMAND is the filename and .COM is the filename extension.

Filenames range from one to eight characters in length and consist of any combination of letters, numbers, and the following symbols:

underscore	_	Ampersand	&
Caret	^	Hyphen	-
dollar sign	\$	Braces	{ }
Tilde	~	Parenthesis	( )

exclamation point	!	At sign	@
number sign	#	Apostrophe	'
percent sign	%		

### ***Filename Extensions***

The second part of the filename is the filename extension. The filename extension has one, two, or three characters and may use the same symbols as the filename. A filename extension is not required, although filename extensions can be helpful in identifying the type of file. Commonly used filename extensions include .DOC for documents, .DAT for data, and .TXT for text files.

You may use any filename extension you choose. However, certain filename extensions have a special meaning to ROM-DOS and should only be used when appropriate. These include:

.COM – used for executable files

.EXE – used for executable files

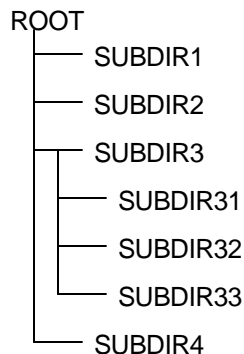
.BAT – used for batch files

Some application programs, such as word processors, may use or require particular filename extensions for output or input files. It is best to follow the application instructions regarding proper filename extensions for that particular program. For example, a file named LETTER1 may include a filename extension such as LETTER1.DOC or LETTER1.TXT.

It is possible to have several files with the same filename but different extensions. ROM-DOS searches for and accesses the filename extensions in the following order: .COM, .EXE, .BAT, and then all others. For example, you could have an executable file named MYPROG.EXE and a batch file call MYPROG.BAT in the current directory. When you enter MYPROG on the command line, the file MYPROG.EXE is executed. If you want to execute the batch file MYPROG.BAT, you must specify the .BAT extension when entering MYPROG on the command line.

### **Tree-Structured Directory System**

ROM-DOS uses a tree-structured directory system. In this system, each branch of the directory, called a subdirectory, is either attached to the main or root directory or is attached to another subdirectory. The following diagram illustrates the directory system and shows two levels of subdirectories under the root directory.



### ***Naming Subdirectories***

You can create any subdirectory structure you choose, giving each subdirectory the name of your choice. The naming of subdirectories is similar to the naming of files. There is an eight-character limit when long filename support is disabled, and you have the same character-choice limitations as for filenames (letters, numbers, and symbols). A subdirectory name can also have an extension. For more information on creating and deleting subdirectories, refer to the MD and RD descriptions later in this manual for more information on creating and removing directories.

### ***Moving around the Directory Tree***

When the computer is first turned on, ROM-DOS boots into the root directory. From the root, you can change to any other directory by means of the CD (CHDIR) command. At any given time, ROM-DOS considers you to be in a specific directory, referred to as the current directory. You can make the computer automatically move to a different directory upon system startup by adding the CD command to your AUTOEXEC.BAT file. Refer to the description of the CD command later in this manual for more information on changing the current directory.

### ***Drive Specifications***

Since ROM-DOS can store and retrieve information from more than one disk drive, disk drives are given unique names such as A:, B:, C:, and so on. By convention, floppy disk drives are identified as drive A: and drive B:. On systems having 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:. Hard drives can be partitioned (divided) into smaller sections with the FDISK utility. Under DOS 6.22, disks exceeding 8GB must be partitioned into two or more areas, with a maximum size of 2GB per partition but will never use more than the first 8GB of the drive. DOS 7.1 does not have these restrictions. A separate drive letter identifies each hard-drive partition. The first partition is drive C:, the next drive D:, and so on. The highest available drive identifier is the letter Z.

To refer to the C: drive, enter the following on the command line

C:

---

**Note:** The drive name may be entered in either uppercase or lowercase.

---

### **Using Wildcard Characters without Long Filename Support**

To simplify a task performed on a large group of similarly named files, use wildcard characters. Wildcard characters allow you to reference groups of files without entering the complete filename for each file in the group. A wildcard character can substitute all or part of a filename or extension. 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 beyond the place where the asterisk is in the search mask. The question mark represents a single character. The following table lists some examples of wildcard character usage.

Example	Description
DIR C:\TEST\*.EXE	Lists all files in the TEST directory having the extension .EXE.

Example	Description
DIR D*.*	Lists all files in the current directory that start with the letter D.
COPY C:\*.BAT B:\*.BAK	Copies all files with a .BAT extension from the C: drive root directory onto the B: drive. The files on the B: drive will have an extension of .BAK. This example backs up a group of files with a single command.
DIR B:\????.*	Lists all the files on the B: drive that have exactly four characters in the filename but have any extension. The question mark substitutes for a single character.
REN TEST?.BAT TEST?.OLD	Renames all files having TEST for the first four characters in the filename, followed by any single character and the .BAT extension. The files retain the same TEST? filename but gain the file extension OLD. The ? can also be used to match a single specific character in a filename.

### **Using Wildcard Characters with Long Filename Support**

To simplify a task performed on a large group of similarly named files, use wildcard characters. Wildcard characters allow you to reference groups of files without entering the complete filename for each file in the group. A wildcard character can substitute all or part of a filename or extension. 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 up to the next character in the search mask. The question mark represents a single character. The following table lists some examples of wildcard character usage.

Example	Description
DIR C:\TEST\*.EXE	Lists all files in the TEST directory having the extension .EXE.
DIR C:\TEST\A*M*.EXE	Lists all files in the TEST directory having a name that begins with an A, contains an M, and has the extension .EXE.
DIR D*.*	Lists all files in the current directory that start with the letter D.
DIR D*	Lists all files in the current directory that start with the letter D.
DIR *D	Lists all files in the current directory that end with the letter D.
COPY C:\*.BAT B:\*.BAK	Copies all files with a .BAT extension from the C: drive root directory onto the B: drive. The files on the B: drive will have an extension of .BAK. This example backs up a group of files with a single command.
DIR B:\????.*	Lists all the files on the B: drive that have exactly four characters in the filename but have any extension. The question mark substitutes for a single character.
REN TEST?.BAT TEST?.OLD	Renames all files having TEST for the first four characters in the filename, followed by any single character and the .BAT extension. The files retain the same TEST? filename but gain the file extension OLD. The ? can also be used to match a single specific character in a filename.



## **System Prompt**

After the execution of each command, ROM-DOS displays the system prompt indicating that it is ready for the next instruction. Unless you define the system prompt otherwise with the **PROMPT** command, the prompt includes only the current disk drive designation followed by a right angle bracket. For example,

```
A:>
```

One common choice for prompt line configuration is to include the current path in addition to the drive designation. For example,

```
A:\MY_FILES>
```

For more information on configuring the system prompt, refer to the **PROMPT** command description later in this manual.

## **Command Line**

Your keystrokes appear to the right of the system prompt on the command line. You can use the following keys to edit the contents of the command line:

<b>Key</b>	<b>Editing Function</b>
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 last command entered on the command line is stored in a command line buffer. You can recall and edit the contents of this buffer as a way of reentering the command to repeat it or make changes to it. For example, if you intended to enter **CHKDSK** but accidentally entered **CHKDSI**, a message appears indicating that **CHKDSI** is a nonexistent command or filename. Rather than reentering the entire string of characters, press **F3** to recall the string and use the **Backspace** key to back up and make the correction.

You may also make corrections to the beginning of long command line entries. For example, suppose you enter the command

```
CPY DATA1.DAT A:DATA1BAK.DAT
```

where the command is misspelled (**COPY** is missing the **O**). To correct the command line, press **F1** to display the first character. Then press **Ins** followed by **O**.

You can now enter the rest of the command by pressing **F3** once.

## Redirecting Input and Output

Certain conventions dictate where each ROM-DOS command receives input data and where it sends output data. However, by using the right angle bracket (>) and the left angle bracket (<), you can redirect the input and output.

### Input Redirection

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

< *filename*

When this is added at the end of the command line, ROM-DOS receives its instructions from the file named *filename*.

---

**Caution:** If input redirection is used and the input file is incomplete, the system will hang and refuse to accept information from the keyboard (except for Ctrl+Alt+Del to reboot).

---

### Output Redirection

The syntax for redirecting output to a file is

> *filename*

When this is added to the end of the command line, standard output is temporarily directed to the file named *filename*. 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, use >>.

>> *filename*

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

For example, if you want to save a list of the current directory contents to a file, you can redirect the DIR command's output to a file by entering:

DIR > MYDIR.TXT

where MYDIR.TXT is the name of the file you want to contain the directory listing.

## Using Batch Files

A batch file is a standard text file containing a list of commands that can be submitted to ROM-DOS for automatic sequential execution. Using batch files helps you avoid unnecessary retyping of command sequences that 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, DOS version 7.1, and Windows 98 DOS box long filename support. Batch files can include internal DOS commands, external DOS commands, batch file commands, names of other executable files or programs, or even the names of other batch files.

### **Batch Filenames**

When naming batch files, use the .BAT extension on the filename. This extension tells ROM-DOS to execute the batch file when its name is entered on 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 filename.

To execute the batch file, enter the filename on the command line. You need not include the .BAT extension unless a file with the same filename and a .EXE or .COM extension is present in the same directory. Batch file execution begins when you press the Enter key.

### **Creating a Batch File**

You can create a batch file by using any word processor or text editor that saves output as unformatted (ASCII) text. Or you can create them by typing directly from the keyboard into a file. This is done with the command

```
COPY CON filename .BAT
```

This tells ROM-DOS to copy the output from the console (keyboard) to the specified file. Once you have entered the above command, you may enter the contents of your batch file.

At the completion of each line, press Enter. As you enter each line, you can make corrections using Backspace and retyping. If you enter an incorrect line or wish to discontinue without saving your work, press Ctrl+C.

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

### **Batch File Command Line Parameters**

A batch file may use parameters placed on the command line. Insert these parameters as arguments for commands or instructions within the batch file. For example, the following batch file, named ARCHIVE, accepts a command line parameter:

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

Execute the ARCHIVE batch file by entering

```
ARCHIVE THISFILE.DAT
```

The %1 parameter takes on the name THISFILE.DAT, and the batch file executes the following to make a copy of THISFILE.DAT in the ARCHIVE subdirectory:

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

---

**Note:** %1 represents the first parameter in a batch file command. If a command has multiple parameters, they are represented by %2, %3, and so on.

---

### **Batch File Commands**

In addition to the standard ROM-DOS commands, there are other commands specifically for batch files. Refer to CALL, CHOICE, ECHO, FOR, GOTO, IF, PAUSE, REM, and SHIFT later in this manual for more information on these commands.

For example, you may often run a program named MY\_INFO1 followed by a program named MY\_INFO2, both of which display a screen of information. After running each of these programs, you always clear the screen before proceeding. Your normal keystroke sequence is:

```
MY_INFO1
CLS
MY_INFO2
CLS
```

You could create a batch file named INFO containing the following commands:

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

After executing MY\_INFO1 (by entering INFO on the command line), the system pauses. When you press a key, the batch file clears the screen and executes MY\_INFO2, then pauses again. Press a key to return to the command line.

**Note:** You can bypass some or all of the commands in your AUTOEXEC.BAT files during system boot. Refer to “**Bypassing CONFIG.SYS and AUTOEXEC.BAT Commands**” on page 24

## ROM-DOS Command Summary

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

Command	Description	Location
?	CONFIG.SYS command. It directs ROM-DOS to pause for confirmation before processing a command.	Page 31
@	Used to suppress the display of a single batch-file command line.	Page 32
;	Identifies nonexecuting lines. The same as the REM command.	Page 32
ANSI.SYS	A console device driver that allows you to support ANSI codes on the local screen.	Page 69
ATA.SYS	A PCMCIA ATA disk device driver.	Page 70
ATTRIB	Displays or modifies the attributes associated with a file.	Page 70
BACKUP	Backs up a single directory tree to a floppy drive, hard disk, or network drive.	Page 71
BREAK	Turns on or off the ability to stop program execution at a non-I/O point.	Page 33
BUFFERS	Sets the number of internal data buffers.	Page 33
CALL	Batch file command. Invokes execution of a secondary batch file.	Page 34
CHDIR (also CD)	Changes the current directory.	Page 34
CHKDSK	Checks the integrity of data on a disk. Displays information.	Page 72
CHOICE	Allows a user to make a processing choice during the execution of a batch file.	Page 73
CLS	Clears all information from the monitor's screen.	Page 35
COMM	ANSI terminal program.	Page 75
COMMAND	Starts a second DOS command processor.	Page 77
COPY	Copies files from one storage location to another.	Page 36
COUNTRY	Designates the country code for displays.	Page 25
CTTY	Changes the default terminal interacting with ROM-DOS.	Page 38
DATE	Displays and allows you to change the date from the system's internal calendar.	Page 39
DEL	Deletes specified files.	Page 39
DELTREE	Deletes one or more directory trees or individual files.	Page 78
DEVICE	Installs a device driver into ROM-DOS.	Page 40
DEVICEHIGH	Loads a device into the upper memory area, if available.	Page 41
DIR	DIRectory. Lists contents of a specified directory.	Page 41
DISK2IMG	Creates an image of a drive or disk.	Page 79

Command	Description	Location
DISKCOPY	Copies the contents of one floppy disk to another of the same type.	Page 80
DISPLAY	Displays international letters and symbols.	Page 80
DOS	Installs ROM -DOS into High Memory Area (HMA).	Page 43
DUMP	Shows contents of memory or a file in hex and ASCII format to the display.	Page 81
ECHO	Batch file command. Displays (on) or hides (off) commands executed from a batch file.	Page 44
EGA/EGA3.CPI:	Font data files for use with the international video display driver, DISPLAY.SYS.	Page 81
EMM386	Enables expanded memory support for capable systems.	Page 82
ERASE	Erases specified files (same as DEL).	Page 44
EXE2BIN	Converts a DOS .exe file to a .com file.	Page 83
EXIT	Used to exit nested running of ROM-DOS within another program.	Page 45
FCBS	Specifies the number of File Control Blocks (FCBS) open at one time.	Page 46
FDISK	Initializes and partitions a hard disk for DOS.	Page 84
FILES	Sets the maximum number of files that can be open at one time on the system.	Page 46
FIND	Works as a filter to display only lines that contain a specified string.	Page 85
FOR	Batch file command. Performs one DOS command on a set of files.	Page 46
FORMAT	Initializes a disk so that ROM-DOS can access files on that disk.	Page 86
GOTO	Batch file command. Moves control to a specified line in the batch file.	Page 47
HELP	Lists all available ROM-DOS commands along with brief descriptions.	Page 47
HIMEM	Manages extended memory and the high memory area on a system that is 286 or greater.	Page 87
IF	Batch file command. Performs a command based on a specified condition.	Page 48
INCLUDE	Allows instructions in one configuration block to be included with instructions in another configuration block.	Page 49
INSTALL	Loads Terminate and Stay Resident (TSR) programs during CONFIG.SYS processing.	Page 5
KEYB	Allows altering of the keyboard layout for a different language or nationality.	Page 88

Command	Description	Location
KEYBOARD/ KEYBRD2.SYS	Keyboard code page data files for use with the international keyboard driver, KEYB.COM.	Page 89
LABEL	Creates, changes, or deletes a disk volume label.	Page 89
LASTDRIVE	Sets the maximum number of drives.	Page 50
LOADHIGH	Loads a program into the upper memory area, if available.	Page 50
MEM	Displays the used and free memory in your system.	Page 90
MENUCOLOR	Allows setting of text and background colors for the startup menu.	Page 51
MENUDEFAULT	Sets the default menu-item choice and time-out value for making a selection.	Page 52
MENUITEM	Specifies an item to be placed on the startup menu display during system boot.	Page 53
MKDIR (also MD)	Creates a new subdirectory.	Page 53
MODE	Modifies the operation of the printer, serial port, and active video display.	Page 90
MORE	Displays a text file one screen at a time.	Page 92
MOVE	Moves files and renames files and directories.	Page 92
MSCDEX	Enables the use of CD-ROM drives.	Page 93
NED	A DOS text editor.	Page 94
NEWFILE	Allows continuation of CONFIG.SYS processing from a new file.	Page 54
NUMLOCK	Sets the NUMLOCK key to on or off when your computer starts.	Page 55
PATH	Displays current command search path(s). A new path line can be specified.	Page 55
PAUSE	Batch file command. Causes execution to halt until a key is pressed.	Page 56
POWER	Conserves power on the system that employs an APM BIOS.	Page 97
PRINT	Prints a list of up to ten files.	Page 99
PROMPT	Resets the appearance of the system prompt line.	Page 57
PROTO	PROTO creates function prototypes in C language files	Page 99
REM	Batch file command for identifying non-executing lines.	Page 58
REMDISK	Remote Disk client.	Page 119
REMSERV	Remote Disk server.	Page 119
REMQUIT	Terminates Remote Disk server.	Page 120
REN	Renames files.	Page 58

Command	Description	Location
RESTORE	Restores files previously saved with BACKUP to the hard disk.	Page 100
RMDIR (also RD)	Deletes a specified subdirectory.	Page 59
RSZ	Zmodem file transfer utility.	Page 101
SET	Sets environment variables and command processor strings.	Page 59
SHARE	Installs the capabilities for file sharing and file locking on your hard disk.	Page 103
SHELL	Allows selections of a command interpreter other than COMMAND.COM.	Page 60
SHIFT	Batch file command. Shifts replaceable parameters one position to the left.	Page 61
SMARTDRV	Disk Caching utility for hard & floppy disks, CD-ROM and other devices.	Page 104
SORT	Utility that sorts text files and displays the output to the standard device.	Page 105
STACKDEV	Increases the number of stacks available for IRQ handlers and Int13h.	Page 105
STACKS	Allows for the use of dynamic data stacks to handle interrupts.	Page 61
SUBMENU	Defines a menu item that represents a secondary menu.	Page 62
SUBST	Allows one drive to appear as another drive.	Page 106
SWITCHES	Allows special CONFIG.SYS file options.	Page 63
SYS	Transfers the hidden system files to a specified drive.	Page 107
TIME	Displays and allows you to change the current time from the system's internal clock.	Page 63
TRANSFER	File transfer utility.	Page 108
TREE	Displays the path of each directory on a specified drive.	Page 109
TYPE	Displays the contents of a text file.	Page 65
UMBLINK	A non-protected mode program that can allow the creation of Upper Memory Blocks using existing RAM areas.	Page 110
VDISK	Allows the use of memory as a simulated disk driver.	Page 111
VER	Displays current version of ROM-DOS.	Page 65
VERIFY	Displays the current VERIFY state or sets the VERIFY state to on or off.	Page 66
VOL	Displays the volume label on a disk.	Page 66
XCOPY	Copies multiple files and, optionally, subdirectories.	Page 111



## *Chapter 4, ROM-DOS Configuration*

---

### **Basic Configuration**

Certain standard settings for your system's operation can be stored in a file named 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**” on page 15.)

You must place the CONFIG.SYS file in the root directory of the drive that is used for system startup or boot. If a CONFIG.SYS file is not found, the following default values are used for the following commands:

```
BREAK = OFF
BUFFERS = 15
COUNTRY = 001
FCBS = 4
FILES = 8
NUMLOCK = ON
SHELL = COMMAND.COM /P /E:128
STACKS = 0,0
```

#### ***Example***

A typical CONFIG.SYS file might look like this.

```
BREAK = ON
FILES = 15
BUFFERS = 15
DEVICE = C:\ROMDOS\HIMEM.SYS
```

### **Using Multiple-User Configurations**

Your CONFIG.SYS file can be used to define multiple system configurations. This is handy when several people share a computer and require different working environments. 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 need to define a startup menu. Each menu item represents a different system configuration option. Then, for each item on the menu, define a configuration block. Each configuration block contains the specific commands to be implemented as the system completes booting.

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) may contain only the following commands (a full description for each command can be found in the Command Descriptions section in Chapter 5):

- MENUITEM
- MENUEFAULT
- MENUCOLOR
- SUBMENU
- NUMLOCK

---

**Note:** Although NUMLOCK may be used outside of a menu/submenu block, it is typically used to enable 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, the following menu displays

```
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 can be 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 they are listed in the CONFIG.SYS file.

When the CONFIG.SYS file is processed by ROM-DOS, it first displays the startup menu that was 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 to be executed. After the menu selection, processing starts with any instructions in CONFIG.SYS prior to the menu block. Then, instructions in the selected configuration block (including instructions added in via an INCLUDE statement) and all common blocks are processed in the order they are listed in CONFIG.SYS. ROM-DOS ignores the 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 number 3 is made, selecting [GAMES], the instructions in the [COMMON] configuration block are processed first, followed by the instructions in the [GAMES] configuration block. The [GAMES] section makes use of the INCLUDE command. All of the instructions provided for the WP menu choice also apply to [GAMES]. If any instructions are in the final [COMMON] configuration block, 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 number 3, GAMES, from the preceding menu, the variable CONFIG is set to GAMES. The CONFIG environment variable can then be used in your AUTOEXEC.BAT file to further customize the startup sequence. This environment variable is referenced by %CONFIG% in your AUTOEXEC.BAT file.

An example of an AUTOEXEC.BAT file that continues the customization process from the preceding MENU may 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 Commands

ROM-DOS offers the capability to bypass some or all of the commands in your AUTOEXEC.BAT and CONFIG.SYS files during the boot process. This feature may be useful in tracking system problems that may be related to one or more commands in either of these two files.

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

1. Turn on, or restart your computer if it is already on, and wait for the following message.  
Starting ROM-DOS...
2. As the above message is being displayed, press the F5 key or hold down the SHIFT key to display the following message.  
ROM-DOS is bypassing your CONFIG.SYS and AUTOEXEC.BAT files.

Your system then continues the boot process using the basic default configurations. You may notice a difference in the way your system behaves. For instance, installable device drivers and memory device drivers are not loaded, and system prompts and paths have default values. If the command interpreter COMMAND.COM is not in the root directory, ROM-DOS may not be able to locate it.

## Stepping Through CONFIG.SYS and AUTOEXEC.BAT Commands

If you suspect that one or more commands in either the CONFIG.SYS or AUTOEXEC.BAT files are causing problems in your system, you can choose to process or bypass each command as follows:

1. Turn on, or restart your computer if it is already on, and wait for the following message.  
Starting ROM-DOS...
2. As the above message is being displayed, press the F8 key to display the first command in the CONFIG.SYS file.
3. For each command, ROM-DOS displays a [Y,N]? prompt. To process the instruction, press Y. To bypass the instruction, press N. ROM-DOS then moves 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, you can press the F5 key at any [Y,N]? prompt.

---

**Note:** You can prompt for a single CONFIG.SYS command by using the question mark (?) command prior to the equals sign (=) in the command line. For a complete description of each CONFIG.SYS command, refer to the internal Command Descriptions starting on page 31.

---

When ROM-DOS finishes all of the commands in the CONFIG.SYS file, it prompts with:

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

Press Y to selectively process the commands in AUTOEXEC.BAT, otherwise press N to bypass all commands in the AUTOEXEC.BAT file.

## Environment Variables

A block of system memory is reserved for the definition of certain strings, called environment variables, to be used in the command processor environment. These include the settings you may establish with the PATH and PROMPT commands as well as COMSPEC, which is automatically defined by ROM-DOS at system startup. Environment variables may be defined using the SET command that is explained in the Command Descriptions section in chapter 6.

## Configuring ROM-DOS for International Use

You can configure ROM-DOS to conform to local conventions for date, time, and currency formats by giving the COUNTRY= command. You can also use the COUNTRY command to select an international character set (known as a code page) that determines the sort order. Any code page can also be shown on EGA and VGA displays by loading DISPLAY.SYS. You can remap a keyboard to provide support for various languages and layouts by running KEYB.COM.

ROM-DOS uses a country code to identify the country conventions that 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. A number, such as 437 identifies each code page.

A computer display has one hardware code page built into it. Typically, this is code page 437, which is 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. The following table lists 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	865
Finland	358	850	437
France	033	850	437
Germany	049	850	437
Hungary	036	852	850
Italy	039	850	437
Japan	081	932	---

Country	Code	Code Page	Alternative Code Page
Latin America	003	850	437
Netherlands	031	850	437
Norway	047	850	865
Poland	048	852	850
Portugal	351	850	860
Russia	007	437	866
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 Country Conventions**

The command to instruct ROM-DOS to use German conventions, for example, is

```
COUNTRY=049
```

The COUNTRY= command requires COUNTRY.SYS to be present in the root directory of the boot drive. Setting a country code affects

- Date and time formats
- 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 the COUNTRY= command. This command tells ROM-DOS to use German conventions for things such as date and time but using code page 437 instead of 850, the default code page.

```
COUNTRY=049,437
```

Setting a system code page affects

- The sort order for alphabetizing
- The rules for converting international letters to uppercase

The individual application programs determine whether they make use of these conventions. For example, DOS 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 are 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 and VGA systems.

If you have an EGA or VGA system, the character fonts are immediately switched 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-like zero character, but ROM-DOS might use a round zero character. The differences are 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, issue the KEYB command from within DOS. You can do this by running KEYB in AUTOEXEC.BAT or directly from a DOS prompt. Use this following command to switch to a German keyboard layout, for example:

```
KEYB GR
```

Most countries have two valid code pages. If you do not specify a code page, the default code page is used. The following table lists the valid combinations.

Country	Country Identifier	Code Page	Alternate Code Page
Belgium	be	850	437
Canadian-French	cf	863	850
Czech Republic	cz	850	852
Denmark	dk	850	865
France	fr	437	850
Germany	gr	437	850
Italy	it	437	850
Latin America	la	850	437
Netherlands	nl	437	850
Norway	no	850	865
Poland	pl	850	852
Portugal	po	850	---
Russia	ru	437	866
Spain	sp	850	437
Sweden	sv	437	850

Country	Country Identifier	Code Page	Alternate Code Page
Swiss French	sf	850	437
Swiss German	sg	850	437
United Kingdom	uk	437	850
United States	us	437	850
Yugoslavia	yu	850	852

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 (Alt+Left-Shift from Russian and Czech Republic keyboards). You can return to the modified keyboard layout by pressing Ctrl+Alt+F2 (Alt+Right-Shift from Russian and Czech Republic keyboards). You can also switch to a completely different layout by running KEYB again and specifying another country identifier.

Appendix G includes diagrams of the different keyboard layouts. The diagrams show native-language keyboards that 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. In the diagrams, symbols appearing in the lower right corner of a key are activated by pressing the AltGr key along with the desired key. On keyboards without a right AltGr, pressing Ctrl+Alt represents the AltGr key.

---

**Note:** The AltGr key is not found on a standard US keyboard.

---

Some keys are prefix keys that 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 alone, press the apostrophe key followed by the space bar. Other keys that may behave as prefixes, depending on the current keyboard layout, are the backward apostrophe (`), tilde (~), and caret (^).

Some keys represent symbols that 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, and you try to produce a capital A with a caret above it, you get a caret character followed by an uppercase A (^A).

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

### **Configuring Your System: an Example**

To completely configure your system, 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 use code page 850 for sorting, uppercase conversions, and the display; and to



switch the keyboard layout to German. 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
```



## Chapter 5, ROM-DOS Internal Commands

---

### Internal Command Descriptions

The following pages provide a complete description of each ROM-DOS internal command, including batch file commands. Each entry includes a description of the command's purpose, command entry syntax, remarks, and examples as appropriate. *Internal* commands are part of the command processor program COMMAND.COM. These functions are only available while COMMAND.COM is running.

For on-line help information and syntax descriptions, use the /? option with any command. For example:

```
DIR /?
```

---

**Note:** The file COMMAND.HLP must be available in the root directory on the boot drive to access help information for internal commands.

---

---

### ?

---

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

#### Syntax

[command]? = *command\_arguments*

#### Remarks

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

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

*command\_arguments* can be any of the available options defined for the command. Refer to the individual command description for complete instructions.

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

#### Example

```
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 will continue. If No (N) is answered, the device will not be loaded and processing moves on to the next CONFIG.SYS command line.

---

---

## @

---

### *Internal Command*

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

#### **Syntax**

@ [batch file command]

#### **Remarks**

The *batch file command* argument can be any executable line in your batch file.

#### **Examples**

@COPY FILE1.BAT FILE1.SAV

Executes the COPY instruction, but the instructions 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 Command*

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. Refer also to the REM command.

#### **Syntax**

;*[any text here]*

#### **Remarks**

Use the (;) command 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

Prevents the VDISK command from executing until the (;) command is removed.

---

## BREAK

---

*Internal Command*

The BREAK command 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 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 Command*

ROM-DOS has internal buffers to temporarily hold data read from the disk. Increasing the number of internal buffers speeds 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 two, and the maximum number is 40. When the number is less than two, the number of BUFFERS is set to two. When the number is larger than 40, then BUFFERS is set to 40. The default number of buffers is calculated using a scale. The ratio is 15 to 640KB. Consequently, a system having 640KB of conventional memory will have 15 buffers. If the calculated number is less than two, then two buffers is used.

**Example**

```
BUFFERS = 10
```

This command tells ROM-DOS to create ten buffers. These ten buffers use 5120 bytes of RAM.

---

---

## CALL

---

*Batch File, Internal Command*

The CALL command 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
```

---

---

## CHDIR (CHange DIRectory)

---

*Internal Command*

The CHDIR command changes the current directory.

**Syntax**

```
CHDIR [drive:][path]subdir
```

```
CD [drive:][path]subdir
```

**Remarks**

*Subdir* is the name of the new current subdirectory. You may use CD in place of CHDIR.

The new directory that is to become the current directory must already exist. Refer to MKDIR for information on the creation of subdirectories.

A series of two periods (..) may be used to indicate a move back to the next -higher or parent directory.

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

### Examples

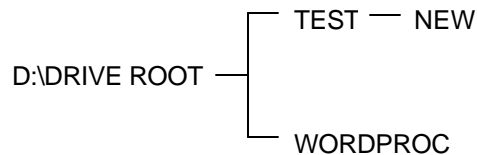
```
CHDIR \TOOLS
```

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

```
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 use this directory tree structure:



```
CD D:\TEST\NEW
```

Moves you into the subdirectory named 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 named WORDPROC. If you start in the TEST directory, you will 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.

## CLS (CLear Screen)

*Internal Command*

The CLS command clears the monitor to display a blank screen.

### Syntax

```
CLS
```

### Remarks

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

## COPY

*Internal Command*

The COPY command copies a file or set of files to a specified destination: another disk, 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 or 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 *filename* or *device*, and the *target* is the copy to *filename* or *device*. Following are various configurations of the COPY command format.

COPY [drive:] [path]filename [option] [drive:] [path]filename [option]

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

COPY [drive:] [path]filename [option] + [drive:] [path]filename [option]  
[drive:] [path]filename [option]

As shown, several source *filenames* may be listed to be copied into the target *filename* that is listed last. The source files are concatenated into the target file.

COPY [drive:] [path]filename [option] device

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

COPY device [drive:] [path]filename [option]

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

### Options

The /A and /B options represent ASCII and binary, respectively, and act as switches that allow each of these file types to be copied. When /A or /B is used, it applies to the preceding *filename*. The option remains in effect for any *filenames* following in the command line until superseded by another /A or /B.

/A and /B options are only needed when combining ASCII and binary files.

/A treats the file as an ASCII file (text file). When 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 treats the file 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 filename, no Ctrl+Z end-of-file marker is added.

/H copies hidden files.

/V is not implemented in ROM-DOS for code size reasons. It is included to maintain command-line compatibility.



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

/-Y confirms the copy of one file over the existing file of the same name. This option overrides the setting made by the COPYCMD environment variable.

Set the COPYCMD environment variable with 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. For proper usage, refer to the SET command.

### Remarks

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

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

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

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

### Examples

```
COPY LETTER.TXT XT A:
```

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

```
COPY *.DOC A:
```

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

```
COPY DATAORIG.DOC DATABACK.DOC
```

Creates a backup copy, DATABACK.DOC, from the file DATAORIG.DOC. The new file is located in the current 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 Command*

ROM-DOS supports multiple country formats for time, date, and currency, and 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.

This command affects the ROM-DOS DATE and TIME commands. Applications that use DOS functions to determine the date, time or currency format, or request that DOS provide character sort order or uppercase information, are affected as well.

Refer to the table provided on page 27 for a list of the currently supported countries.

**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 20.06.1998
Enter new date (dd.mm.yyyy):
TIME
Current time is 16:39:54,45
Enter new time:
```

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

---

## CTTY (Change TeleTYpe)

---

*Internal Command*

The CTTY command directs input and output to a different device other than your computer's standard keyboard and monitor.

**Syntax**

CTTY *device*

**Remarks**

Use CTTY for any situation requiring interaction with an alternate console.

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

The DATE command displays the current date (month, day, year) as known to ROM-DOS and also allows you to change it.

### Syntax

DATE [*mm-dd-yy*]

DATE [*mm-dd-yyyy*]

### Remarks

The date set by this command is used, among other things, for date stamping your file revision dates. This information is displayed when you execute a directory listing of your files.

You may want to include the DATE command in your AUTOEXEC.BAT file, so you can set the date during boot. If your computer has an internal, battery-operated clock, you won't need to do so.

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.

Refer also to the TIME command.

### Examples

When entering this command without specifying *mm-dd-yyyy*, the current date as known to ROM-DOS is displayed, and you are prompted to enter a new date.

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

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

Alternatively, you may skip the display and prompting by entering the current date on the command line. To enter June 10, 1999 (assuming US country support), enter the DATE command as follows:

```
DATE 6-10-1999
```

Valid entries for months, days, and years are

```
mm = 1-12 dd = 1-31 yyyy = 1980-2099
```

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

---

## DEL (DElete)

---

*Internal Command*

The DEL command deletes a specified file or set of files.

**Syntax**

```
DEL [drive:] [path] filename [/P]
```

**Remarks**

The DEL command and the ERASE command are functionally identical.

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

Global filename characters ? and \* (wildcards) can be used in the *filename*. This should be done with caution as it is possible to delete multiple files unintentionally.

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

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

Enter Y to delete all files in the specified subdirectory.

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

---

**Caution:** No ROM-DOS command can undelete a file. Although utilities exist that 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 deletion.

---

**Options**

The /P option causes DEL to pause and prompt you before each file is deleted. This option is most useful when deleting files with wildcards.

**Examples**

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

Deletes all files on the A: drive with a .DOC extension. Before each file is deleted, you are prompted to determine if that file should be deleted. A sample prompt is shown below:

```
C:\> DEL MYLETTER.DOC
MYLETTER.DOC, Delete (Y/N) ?
```

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 Command*

The DEVICE command installs a device driver.

**Syntax**

```
DEVICE = [drive] [path] driver name [arguments]
```

**Remarks**

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

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

**Example**

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

Installs the ROM-DOS RAM disk driver, VDISK.SYS, via the DEVICE command and dedicates 120KB of extended memory to the RAM disk.

---

## DEVICEHIGH

---

*CONFIG.SYS Command*

The DEVICEHIGH command loads an installable device driver into the upper memory area, if available.

**Syntax**

```
DEVICEHIGH = [drive] [path] driver name [arguments]
```

**Remarks**

A device driver allows ROM-DOS to access hardware that is not common in all PCs. 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 is loaded into conventional memory.

The full drive path and filename of the device must be specified. The arguments differ depending on the device driver.

**Example**

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

Installs a driver MYDEVICE with its command line arguments as specified. The device is loaded into upper memory, if available.

---

## DIR (DIRectory)

---

*Internal Command*

The DIR command displays a list of the files that are in a specific directory.

**Syntax**

```
DIR [drive:] [path] [filename] [/option]
```

**Remarks**

Use the DIR command to list all the files in a directory or to show the directory entries of specific files. The standard directory display format includes columns for filenames, filename extensions, file sizes, and the dates and times the files were created.

### Options

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

Attribute	Description
A	Archive-ready for archiving
D	Directories
H	Hidden files
R	Read only files
S	System files
X	Show attributes
-	The minus sign can be used to negate listed attributes. For example, to select all files that do not have the archive bit set, use /A-A option.

The /B or bare option causes the filenames to be displayed without volume label, date, time, or size information.

The /L option causes the filenames to be displayed in lowercase.

The /P option selects page mode, which makes ROM-DOS pause the display each time the screen is full. Press any key to go onto the next page of entries.

The /O option causes the filenames to be displayed in sorted order. The sort order can contain one or more of the following attributes.

Attribute	Description
D	By date and time, newest first
E	Alphabetic order by extension
G	Directories grouped before files
N	Alphabetic order by name
S	By size, smallest first
-	The minus sign can precede the sort option to reverse the sort order. For example, to sort all files in the directory in reverse alphabetic order, use /O-N option.

The /S option causes the display to include files in subdirectories also.

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

The DIRCMD environment variable can be used to set the default preferences for the DIR command. The SET command assigns the values to an environment variable. Refer to the SET command for proper usage. For example, to always have the /P option set for DIR, use the statement SET DIRCMD=/P. You can override the default settings in DIRCMD by using the minus sign (-) preceding the option. To cancel the paging for a single use of the DIR command, enter DIR /-P.

The DIRSIZE environment variable is useful for nonstandard screen sizes. As with the DIRCMD variable, the SET command assigns the values. The syntax is

SET DIRSIZE *rows*[,*columns*]

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

### Examples

DIR

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

DIR B:\MEMOS

Lists all files in the subdirectory MEMOS on drive B:.

DIR A:\*.\*RPT/P

Lists the directory entries for all files in drive A: with the extension RPT, displayed one screen at a time.

DIR /ON

Lists all files sorted by filename order.

DIR /AH

Lists all hidden files.

DIR \*.\*.DOC /B

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

## DOS

### CONFIG.SYS Command

ROM-DOS can be loaded into an upper portion of memory referred to as the High Memory Area (HMA), freeing more of conventional (lower 640KB) DOS memory for use by applications.

### Syntax

DOS=[HIGH | LOW]

### Remarks

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

This command only works on 286 and higher CPUs with extended memory and Datalight's HIMEM.SYS High Memory Manager, or equivalent, installed. It will not work on standard XT class PCs. Setting DOS=HIGH is ignored when ROM-DOS is in ROM.

Refer also to the **HIMEM** device driver description on page 87.

### Example

```
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 File, Internal Command*

The ECHO command 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*

ECHO

### Remarks

The ON option is the default ECHO setting. It causes commands in a batch file to be displayed as ROM-DOS executes them. Typing ECHO OFF turns off such display, after which the ON option switches it back on again.

The ECHO command alone, entered without the ON or OFF option, displays the current ECHO setting.

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

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

The @ symbol can be used to suppress the echoing of a single command when ECHO is on. Place the @ symbol first on the command line. Refer also to the description of the @ command for additional information.

### Examples

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

A batch file message with more than one line.

```
ECHO OFF
```

Sets the ECHO to off.

---

## ERASE

---

*Internal Command*

The ERASE command deletes a specified file or set of files.

### Syntax

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

### Remarks

The DEL command and the ERASE command are functionally identical.



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

Global filename characters ? and \* can be used in the *filename*. This should be done with caution as it is possible to delete multiple files unintentionally.

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

Are you sure (Y/N) ?

Enter Y to erase (delete) all the files in the specified subdirectory.

---

**Caution:** ROM-DOS has no command to unerase a file. Although utilities exist that can attempt an unerase, 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, refer to the RMDIR command.

### Options

The /P option causes ERASE to pause and prompt before each file is deleted. This option is most useful when deleting files with wildcards.

### Examples

```
ERASE MYLETTER.DOC
MYLETTER.DOC, Delete (Y/N)?
```

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, you are prompted to determine whether that file should be erased.

---

## EXIT

---

*Internal Command*

The EXIT command exits a secondary nested ROM-DOS operation and returns control of the system to the primary program.

### Syntax

```
EXIT
```

### Remarks

The EXIT command 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 affected if it is loaded without the /P permanent option.

## FCBS

---

*CONFIG.SYS Command*

The FCBS command 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. The *minimum number* specifies the minimum number of FCBs to be open at all times. The *minimum number* argument has the same default and range value as the *number* argument.

### Example

FCBS = 8, 4

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

---

## FILES

---

*CONFIG.SYS Command*

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

### Syntax

FILES = *number*

### Remarks

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

### Example

FILES = 10

Specifies the maximum number of open files to ten.

---

## FOR

---

*Batch File Command*

The FOR command allows repeated execution of a ROM -DOS command applied to a set of files.

### Syntax

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

**Remarks**

During 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 filenames or a global file specification using wildcard characters.

The FOR subcommand can be used directly on the command line and 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) DO PRINT %%N
```

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

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

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

---

## GOTO

*Batch File, Internal Command*

The GOTO subcommand 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.

**Example**

```
GOTO MESSAGE
```

This command moves the control of execution within the batch file to a line that says

```
:MESSAGE
```

---

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

---

---

## HELP

*Internal Command*

The HELP command provides on-line help of each ROM-DOS command.

**Syntax**

HELP *command*

**Remarks**

The *command* argument refers to any ROM-DOS internal command or external utility.

The COMMAND.HLP file must be available (i.e., in the path) 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.

All available batch file commands are also listed by HELP.

**Examples**

```
HELP DIR
DIR /?
```

Both commands list the help of the DIR command.

**IF***Batch File Command*

The IF subcommand 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 [ *drive:* ] [ *path* ] *filename*

If the *condition* is true, then 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 condition.

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

**Examples**

```
IF ERRORLEVEL 15 GOTO EXIT
```

Will GOTO 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
```

Copies CONFIG.SYS to OLD if a file named OLD does not exist.

---

## INCLUDE

---

### *CONFIG.SYS Command*

The INCLUDE command 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.

#### Syntax

```
INCLUDE = blockname
```

#### Remarks

This command is useful for sets of instructions common to several system configurations. The commands are defined once in a single configuration block and then inserted into other configuration blocks via the INSERT command. For additional details, refer to 'Using Multiple-User Configurations' on page 21.

#### Example

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

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

When you choose WORDPROC from a CONFIG.SYS menu, the instructions in the configuration block labeled [WORDPROC] are carried out. The instructions in the INCLUDED block labeled [MISC] are also implemented as part of the [WORDPROC] block of instructions.

---

## INSTALL

---

### *CONFIG.SYS Command*

The INSTALL command loads Terminate and Stay Resident (TSR) programs during CONFIG.SYS processing.

**Syntax**

INSTALL = [drive:][path] TSR\_Program TSR\_Arguments

**Remarks**

The TSR program is loaded 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.

**Example**

INSTALL = C:\BIN\MOUSE.COM

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

---

---

## LASTDRIVE

---

CONFIG.SYS Command

The LASTDRIVE command sets the maximum number of drives.

**Syntax**

LASTDRIVE = letter

**Remarks**

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

The minimum number for 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 nonstandard drives that are not in your system. These drives may be CD-ROM drives, flash disk drives, or network drives.

**Example**

LASTDRIVE = H

Causes ROM-DOS to allocate space for eight drives. If the computer has five drives installed, there is room for three additional nonstandard drives.

---

---

## LOADHIGH

---

CONFIG.SYS, Internal Command

The LOADHIGH command loads an executable or TSR program into the upper memory area, if available. LOADHIGH can be run as a batch file command or from the DOS command line.

**Syntax**

LOADHIGH = [drive:][path]executable [arguments]

-or-

$LH = [drive:][path]executable [arguments]$

### Remarks

An executable or TSR program can be loaded into the upper memory areas when they are available and have 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 filename of the device must be specified. The arguments are different depending on the device driver.

### Example

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

Installs an executable named CHECKIT with its command line arguments as specified. The program loads into upper memory, if available.

## MENUCOLOR

*CONFIG.SYS Command*

The MENUCOLOR command allows you to set 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

The *text\_color* argument 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.

The *background\_color* argument 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, which rely on BIOS screen support, are unusable. 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, which is an unusable choice for screen viewing. Using the black/black combination in the MENUCOLOR command line signifies to ROM-DOS to display the startup menu in TTY mode without using BIOS screen/cursor positioning or color changing commands.

#### Color Values:

0 – Black	1 – Blue	2 – Green
3 – Cyan	4 – Red	5 – Magenta
6 – Brown	7 – White	8 – Gray
9 – Bright Blue	10 – Bright Green	11 – Bright Cyan

12 – Bright Red      13 – Bright Magenta      14 – Bright Yellow  
 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 Command*

The MENUDEFAULT command allows you to set the default menu-item choice and a time-out value for making a menu selection. This command can only be used within a menu configuration block in the CONFIG.SYS file. Refer also to ‘**Using Multiple-User Configurations**’ on page 21.

### Syntax

```
MENUDEFAULT = blockname[,timeout]
```

### Remarks

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

The optional time-out argument represents the number of seconds ROM-DOS waits for a user input selection before initializing your system with the default configuration. The time-out period can be set to a value between 0 and 90. If you select 0, the default menu item is automatically implemented without a wait. If you do not enter a time-out value, ROM-DOS will 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.

### Example

```
[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 you fail to make a selection within 20 seconds, the Word\_Proc block is processed.



---

## MENUITEM

---

*CONFIG.SYS Command*

The MENUITEM command allows you to specify an item on the startup menu. This command can only be used within a menu configuration block in the CONFIG.SYS file. Refer also to ‘**Using Multiple-User Configurations**’ on page 21.

### Syntax

```
MENUITEM = blockname [,menu_text]
```

### Remarks

The *blockname* argument is a user-defined label given to a configuration block defined elsewhere in the CONFIG.SYS file. If a user selects the menu item, all commands in the selected configuration block are processed, along with the instructions that are 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.

The *menu\_text* option is a descriptive statement that defines the *blockname*. The *menu\_text* is displayed on the screen as a line item in the startup menu. The *menu\_text* argument can be up to 70 characters long and can contain any characters. If this argument 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.

### 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 and Development. Each of these has descriptive text and a set of commands defined later in the CONFIG.SYS file. At boot time, these menu items are displayed in the startup menu as follows:

```
ROM-DOS 6.22 STARTUP MENU
1. Word Processing
2. Network
3. Research and Development
Enter a choice: 1
```

---

## MKDIR (MaKe DiRectory)

---

*Internal Command*

The MKDIR command creates a new subdirectory.

### Syntax

```
MKDIR [drive:] [path] subdir
```

MD [*drive:*] [*path*]*subdir*

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

#### Remarks

If no drive or path is specified, the new subdirectory is created within (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 named TEMPDIR1 within the current default directory.

```
MD C:\UTIL\TOOLS
```

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

---

## NEWFILE

---

*CONFIG.SYS Command*

The NEWFILE command allows you to continue CONFIG.SYS file processing from a new file. The file can be located in another directory or even on a different drive.

#### Syntax

NEWFILE=*filename*[*,device*]

#### Remarks

The 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 is processed along with the main CONFIG.SYS file upon starting the system.

When the NEWFILE= instruction is processed, control passes 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 returns to the original CONFIG.SYS file and the next instruction is processed. The next instruction can even be a second NEWFILE= statement, allowing flexibility for systems which may have a variety of drives installed or not installed at boot time. The first successful NEWFILE statement is processed, transferring control to the specified filename. The remaining instructions in the original CONFIG.SYS file, including NEWFILE= statements, are not processed.

NEWFILE commands can be nested. That is, 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 filename in the successive steps has a unique name, otherwise, you will create an infinite loop 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.

The presence of *[,device]* on the NEWFILE command line will tell the ROM-DOS kernel to load this device driver before trying to process this NEWFILE statement. The format of this parameter is exactly the same as the parameter for a DEVICE= statement. See page 40 for a description of the DEVICE= statement.

### Examples

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

Causes instructions in the file NEWCFG.SYS, located in the C:\BIN directory, to be executed as part of the CONFIG.SYS file.

```
NEWFILE=C:\BIN\NEWCFG.SYS,ROMDRIVE.SYS C000
```

Causes the device driver, ROMDRIVE.SYS, to be executed prior to processing the instructions in the file NEWCFG.SYS as part of the CONFIG.SYS file.

Also, the environment will contain the following entry:

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

Verify this by running the SET command with no parameters.

## NUMLOCK

*CONFIG.SYS Command*

The NUMLOCK command sets the NumLock key on the keyboard to ON or OFF when your computer starts.

### Syntax

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

### Remarks

Selecting ON designates that the NumLock key is set to on when DOS boots. Selecting OFF designates that the NumLock is off when DOS boots. In either case, you still have the ability to manually turn the NumLock key on and off after booting with the NUMLOCK command.

### Example

```
NUMLOCK=on
```

Sets the NumLock key to on when the system boots.

## PATH

*Internal Command*

The PATH command sets the search path for command files that are not in the current directory.

### Syntax

```
PATH [drive:][path][;[drive:][path]] ...
```

**Remarks**

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

To append one or more additional directories to the PATH, use %PATH% for the current path. For example, to add C:\DOS to the current path, enter:

```
PATH %PATH%;C:\DOS
```

at the command prompt.

Enter PATH without parameters to display the current path.

To cancel previously-set command paths, enter:

```
PATH =
```

or

```
PATH =;
```

**Example**

If your application programs reside on a fixed disk, the PATH command enables you to start any of them from any drive or directory. To access utilities, a word processor, and a spreadsheet in subdirectories C:\UTIL, C:\WP, and C:\123, set the path command as follows:

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

## PAUSE

*Batch File, Internal Command*

The PAUSE command suspends the execution of a batch file and resumes operation when any key is pressed.

**Syntax**

```
PAUSE
```

**Remarks**

A batch program 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, or make a decision, 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 displays

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

Responding Y ends the batch job. Therefore, strategic placement of the PAUSE command allows you to divide the batch file into sections so you can end it at some intermediate point.

The *message* option allows you to display 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 on.

Example

```
PAUSE Place blank disk in drive A:
Prompts the user to insert a disk and suspends operation until a key has been hit.
```

PROMPT

Internal Command

The PROMPT command changes the ROM-DOS command prompt.

Syntax

```
PROMPT [text] [$character] [$character...]
```

Remarks

The prompt that ROM-DOS normally displays is the letter of the current drive followed by a right angle bracket (>) (the greater-than symbol). By using the PROMPT command, you can change this prompt to include any combination of a message, the current directory, the date, the time, and some other features.

Code	Corresponding Prompt
\$T	Current time
\$D	Current date
\$P	Current drive and 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 (carriage return)
\$	The \$ character

Examples

```
To show this prompt
Current directory is drive:\path;
Ready for <command>
enter
PROMPT Current directory is $P;$_Ready for $Lcommand$G
To show the date, time, and current directory on separate lines followed by the greater-
than character and a space, enter:
PROMPT $D$_$T$_$P$G<space>
```

where *<space>* refers to pressing the spacebar once. The resulting prompt is

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

---

## REM (REMark)

---

*Batch File, Internal Command*

The REM 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 (;) command.

### Syntax

```
REM [message]
```

### Remarks

The REM command provides information but has no affect on the execution of the batch file.

The comment may consist of any set of characters. You may also create a blank line 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.

### Examples

```
REM This batch file created by
REM Jane Doe
```

These lines may be added at any point in a batch file as user information only.

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

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

---

## REN (REName)

---

*Internal Command*

The REN command changes the name of a file.

### Syntax

```
REN [drive: ][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 ? may 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 Command*

The RMDIR command removes (deletes) a specified empty subdirectory.

**Syntax**

```
RMDIR [drive:][path]subdir
```

```
RD [drive:][path]subdir
```

*subdir* is the name of the subdirectory being deleted. Note that RD may also be used.

**Remarks**

If no drive or path is specified, RMDIR looks for the specified *subdir* within (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 when you attempt to remove a subdirectory that still contains files or other subdirectories.

**Example**

```
RD TOOLS
```

Removes the TOOLS subdirectory from the current directory, assuming TOOLS is an empty directory.

---

## SET

---

*Internal Command*

The SET command sets, displays, or removes environment variables.

**Syntax**

```
SET [variable = [string]]
```

**Remarks**

Use the environment variables to control the behavior of programs and batch files and also the behavior of ROM-DOS. Use this command in the AUTOEXEC.BAT and CONFIG.SYS files and on the DOS command line. The environment variables that can

be 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.

## SHELL

*CONFIG.SYS Command*

The SHELL command allows you to specify a command interpreter other than the default COMMAND.COM or to load COMMAND.COM with non-default arguments (parameters). ROM-DOS boots this new program, with arguments, instead of that specified internally.

#### Syntax

SHELL = *cmd\_interpreter arguments*

#### Remarks

The SHELL command is most often used to start the initial copy of COMMAND with special parameters. One parameter provides a larger environment than the default 128 bytes.

The *cmd\_interpreter* 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 and vary depending on the *cmd\_interpreter* 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 named MYPROG.EXE, located in the directory TEMP, instead of the standard Command Processor.



---

## SHIFT

---

*Batch File, Internal Command*

The SHIFT command 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

This command 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.

### Example

The following batch file 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, verifying the existence of the file, and continues.

```
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
```

---

## STACKS

---

*CONFIG.SYS Command*

The STACKS command enables the dynamic use of data stacks to handle hardware and software interrupts that use large amounts of stack space. You may use this command only in your CONFIG.SYS file. Use the STACKS command if the system crashes or encounters a stack overflow during the boot or at runtime.

STACKS uses more RAM for the DOS stacks, which you can calculate with the formula (number-of-stack\*stack-size). The maximum extra DOS stack size is 32KB (64\*512).

**Syntax**

STACKS = *n,s*

**Remarks**

*n* specifies the number of stacks. Valid values for *n* are 0 and integers in the range 8 through 64.

*s* specifies the size (in bytes) of each stack. Valid values for *s* are 0 and integers in the range 32 through 512.

## SUBMENU

*CONFIG.SYS Command*

The SUBMENU command defines a menu item that represents a secondary menu when selected. This command may only be used within a menu configuration block in the CONFIG.SYS file.

**Syntax**

SUBMENU=*blockname*[ ,*menu\_text* ]

**Remarks**

The *blockname* argument 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 blocknames.

The optional *menu\_text* argument specifies the text that ROM -DOS displays for this menu item on the startup menu. If this argument is left out, ROM -DOS displays the *blockname* as the text. The *menu\_text* can be up to 70 characters long and may contain any character.

The submenu can be defined with any user-provided descriptive label. It need not have the [MENU] label.

**Example**

```
[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
```

In the preceding example, a submenu is defined as one of the startup menu choices. When you 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 Command*

The SWITCHES command allows special CONFIG.SYS file options.

### Syntax

SWITCHES=[/k][/n][/f]

### Remarks

The /k argument makes an enhanced keyboard behave like a conventional keyboard.

The /n argument prevents the use of the F5 and F8 function keys to bypass the startup commands.

The /f argument instructs ROM-DOS to skip the delay after displaying the “Starting ROM-DOS...” message at boot time. The delay allows you time to use the F5 and F8 options to alter the processing of the CONFIG.SYS and AUTOEXEC.BAT files as described in page 24.

### Example

```
switches = /n
```

Prevents you from using the F5 and F8 keys at boot time.

---

## TIME

---

*Internal Command*

The TIME command displays the current time as shown on the system’s internal clock. Allows resetting of the clock.

### Syntax

TIME [hh:mm:ss] [pm|am]

**Remarks**

The time set by this command is used, among other things, for time stamping your file revision dates. This information is displayed when you execute a directory listing of your files.

You may want to include the TIME command in your AUTOEXEC.BAT file to set the date at boot time. If your computer has an internal, battery-operated clock, you won't need to do so.

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. Refer also to the DATE command.

If you just want to check the time maintained by ROM-DOS, enter the TIME command alone. ROM-DOS displays something like this

```
Current time is 3:00:02.48p
Enter new time:_
```

After which you press Enter to return to an empty command line.

When you want to change the time, you can include the desired time on the prompt line after the word TIME. Or you may enter the command with no option (as you do to check the time) and enter the new time before pressing Enter.

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 entered 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 designators.

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 needn't enter seconds or hundredths; ROM-DOS assumes a value of zero if they are not specified.

You may skip the display and prompting by 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**

To set the time to 11:15 p.m., enter

```
TIME 23:15 or TIME 11:15 p
```

To set the time to 9:26 a.m., enter

```
TIME 9:26 or TIME 9:26 am
```

---

## TRUENAME

---

*Internal Command*

The TRUENAME command displays the actual path of virtual drives created with SUBST, network drives, normal drives, and the contents of environment variables.

**Syntax**

```
TRUENAME [path [%environmentvariable%]]
```

**Remarks**

TRUENAME will display the full path to a network drive or a drive created with SUBST.

If TRUENAME is not given a path or environment variable it will display the full path of the current drive up to the current directory.

Long directory or file names will be truncated to 8.3 format.

Network drives will be expanded to show the drive letter in //SERVER/VOLUME: format.

**Option**

The %environmentvariable% option will cause TRUENAME to display the contents of the environment variable passed to truenamename.

**Examples**

```
TRUENAME C:
```

Displays the full true path of drive C:.

```
TRUENAME G: (where G: is a drive created with SUBST)
```

Displays the full true path of the actual drive and path that was SUBST'd.

---

## TYPE

*Internal Command*

The TYPE command displays the contents of a text file on the screen.

**Syntax**

```
TYPE [drive:] [path]filename
```

**Remarks**

The *drive* and/or *path* arguments must refer to a valid drive and directory combination.

The *filename* may not contain any wildcards.

If a file containing formatting codes or other nonalphanumeric characters is displayed with TYPE, unintelligible characters are displayed. This does not harm the system.

**Example**

```
TYPE A:AUTOEXEC.BAT
```

Displays the AUTOEXEC.BAT file on drive A.

---

## VER

*Internal Command*

The VER command displays the version number of the 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. Note that 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.

**Option**

The /R option shows the full version and revision number of ROM-DOS.

**Example**

VER 5.0

Changes the record of the 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 Command*

The VERIFY command displays or modifies the VERIFY state.

**Syntax**

VERIFY [ON | OFF]

**Remarks**

The VERIFY command does not perform any data verification (same as the COPY /V option). It is included to provide batch file compatibility.

---

---

## VOL

---

*Internal Command*

The VOL command displays the volume label on a specified disk.

**Syntax**

VOL [*drive:*]

**Remarks**

If you do not specify a *drive*, the current drive is assumed. VOL does not allow the setting of volume labels. Refer to the LABEL command on page 89 for instructions on setting the volume labels.

**Examples**

VOL

Causes ROM-DOS to display the volume label on the default drive, which is the A: drive.

VOL C:

Causes ROM-DOS to display the volume label on the C: drive.





## Chapter 6, ROM-DOS Utility Descriptions

---

### ROM-DOS Utilities

#### Command Descriptions

The following pages provide a complete description of each ROM-DOS external command. Each entry includes a description of the command's purpose, command entry syntax, remarks, and examples as appropriate. *External* commands are stand alone utility programs and device drivers.

For on-line help information and syntax descriptions, use the `/?` option with any command. For example:

```
XCOPY /?
```

---

### ANSI

---

ANSI.SYS is a console device driver that allows you to support ANSI codes on the local display.

#### Syntax

```
Device=ANSI.SYS [options]
```

#### Remarks

ANSI.SYS supports standard ANSI escape sequences.

ANSI.SYS writes directly the screen when using text video mode.

#### Options

The `/K` option forces use of the extended keyboard BIOS calls which sense F11 and F12.

The `/X` option lets you redefine the extended keys independently.

The `/S` option disables the keyboard redefinition feature.

The `/Tnm` option indicates that the video mode *nm* is a text mode. By default, modes 0, 1, 2, 3 and 7 are text modes.

#### Examples

```
DEVICE=ANSI.SYS
```

This example loads ANSI.SYS with default settings.

```
DEVICE=ANSI.SYS /T54 /S
```

Load ANSI.SYS with mode 54h as a video text mode and disable keyboard redefinition.

## ATA

---

ATA.SYS is a PCMCIA ATA disk device driver.

### Syntax

DEVICE = ATA.SYS [/A:xxx] [/L:xxx]

### Remarks

ATA.SYS requires an Intel 82365 or compatible controller. The PCMCIA controller card itself is always mapped to addresses 3E0h and 3E1h. These are fixed addresses and can not be changed. This driver supports only 5-volt ATA cards. Up to two drives can be supported with the driver. Use the Datalight FORMAT command to format an ATA disk, if it is not already formatted.

ATA Cards tested:

Integral 1841PA 17MB ATA card 660KB/s write, 690KB/s read

Toshiba TH6SS160201AA 20MB ATA card 900KB/s write, 1200KB/s read

SanDisk SDCFB 4MB CompactFlash card 300KB/s write, 1000KB/s read

SanDisk SDP3B 2MB ATA card 275KB/s written 1000KB/s read

Controllers tested:

Vadem VG-465

Intel 82365SL rev A

### Options

The /A:xxx option sets the memory window that the ATA card gets mapped to for use. This is a 4KB window. The default is C000. Replace xxx with the correct memory segment window for your installation.

The /L:xxx options sets the I/O address where the ATA card is mapped to by the controller. The default I/O address is 240h. The memory usage for this is 16 bytes starting at the given address.

### Examples

DEVICE = ATA.SYS

Load ATA.SYS with the default memory segment and I/O address settings.

DEVICE=ATA.SYS /AD000 /I290

Load ATA.SYS using memory segment address D000h and I/O address 290h

---

## ATTRIB

---

The ATTRIB command either displays or modifies the attributes of a file or directory.

**Syntax**

```
ATTRIB [+R | -R] [+A | -A] [+S | -S] [+H | -H] [[drive:][path]filename] [/S] [/C]
```

**Remarks**

The file attributes define the characteristics of a file. They determine if a file may be deleted or modified, or if it is archived. Use the ATTRIB command to manage these file attributes.

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

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

Option	Description
+/-	Add(+) or remove(-) attribute inserted before each option.
A	Archive attribute
-C	Clear all attributes
H	Hidden file attribute
R	Read Only attribute
S	System file attribute
/S	Recurse into subdirectories

If ATTRIB finds no modify commands, then it displays the files in the specified directory along with the filenames and their current attributes.

**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.

---

## BACKUP

---

The BACKUP command backs up a single directory tree to a floppy drive, hard disk, or network drive.

**Syntax**

```
BACKUP srcpath dstdrive [/A] [/H] [/L[:<logname>]] [/M] [/S] [/Y] [/?]
```

**Remarks**

The complement program, RESTORE, restores a backup set to hard disk. Backup creates one or more backup volumes in the form of DL970507.001, where the year, month, and day compose the name, and the volume number is the file extension.

The *<srcpath>* (source path) can be any legal DOS path, with an optional file mask, such as D:\SOURCE\\*.C.

The *<dstdrive>* (destination drive) can be any legal DOS drive.

**Note**

At the time this manual was updated, this utility did not contain long filename support.

**Options**

The A option appends the data to an existing backup file.

The /H option backs up hidden files as well as normal files.

The /L option writes a .LOG file in ASCII text form. If no log filename is specified, BACKUP creates a .LOG file in the current directory.

The /M option backs up only files that do **not** have the archive bit set. BACKUP automatically clears the archive bit for each file it backs up. Use Datalight ATTRIB to view/set the archive bit manually.

The /S option backs up the entire tree, not just the one directory.

The /Y option allows BACKUP to operate in batch files with no user input (affirmative is given for all prompts). However, if the backup set requires multiple floppies, BACKUP prompts for all floppies after the first one.

**Note:** BACKUP operates much faster while using any disk cache program.

**Example**

The following command backs up all files in the C:\DEV\ROMDOS directory, including subdirectories, to the B: drive.

```
BACKUP C:\DEV\ROMDOS B: /S
```

---

## CHKDSK (CHecK DiSK)

---

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

**Syntax**

```
CHKDSK [drive:][path][filename]/C [/F] [/V]
```

**Remarks**

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

**Options**

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

The /F option causes CHKDSK to fix FAT or directory errors on the disk if any are found. If /F is not specified, then CHKDSK acts as if fixing the disk, but no corrections are written 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 drive. Each file will have 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.

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

If a *filename* is used, then CHKDSK displays all files matching that specification that have noncontiguous data areas on the disk. Files that are stored in noncontiguous areas, especially .EXE files, have slower disk access times. If CHKDSK reports a large number of files with this problem, you should use a utility program that optimizes the files and frees 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 11-09-2001 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:. The report is printed to the console.

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

Checks the integrity of hard drive disk D. The report is saved in a file named DRIVE\_D.RPT.

---

## CHOICE

---

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

### Syntax

```
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 the keys is YN. More than two key choices may be entered.

*/N* prevents display of the user prompt. The selected keys are still valid, but they are not displayed in the prompt message.

*/S* selects case-sensitivity regarding the prompt. With the */S* switch, you must enter the response in the exact case used with the */C* option.

*/T[:]c,nn* sets 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. The argument *c* is the default-key choice character. The *nn* argument specifies the number of seconds to pause. Valid number selections are from 0 to 99. A 0 setting produces no pause.

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

**Examples**

When the following CHOICE command is used

```
CHOICE /c:ync
```

you will see:

```
[Y,N,C]?
```

If the text argument is added

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

the display will be:

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

If the key choice prompt is left off

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

only the following appears (note key choice of YN is default)

```
Continue reading file?
```

A complete batch file example follows. Each prompt selection returns an errorlevel that can be trapped. The errorlevel 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.
c:\dos\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
```

---

## COMM

---

The COMM communications program provides the ability to communicate with a remote ROM-DOS system. COMM supports Xmodem file transfer, autodialing, Zmodem, and terminal emulation and time zones.

### Command Line Options

All command line options must be separated by a space.

Option	Description
/B#	Sets the baud rate to # on startup. The available baud rates are 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, and 115200.
/COM#	Sets the communications port to COM1 or COM2; both are supported.
/8N1	Sets the serial port to 8 data bits, no parity, one stop bit.
/7E1	Sets the serial port to 7 data bits, even parity, one stop bit.
/8N2	Sets the serial port to 8 data bits, no parity, two stop bits.
/8O1	Sets the serial port to 8 data bits, odd parity, one stop bit.
/8E1	Sets the serial port to 8 data bits, even parity, one stop bit.

You can reverse the placement of the parity and data bits on the command line. For example, /8N1 is equivalent to /N81.

### Environment Variables

The environment variable COMM is supported, which may set certain communications parameters. The switches are identical to the command line options, for example:

```
SET COMM= /COM2 /B2400 /7E1
```

Causes COMM to start using COM2, at 2400 baud, with 7 data bits, even parity, and 1 stop bit. If new options are specified on the command line, they override the environment variable settings. Invalid options are ignored.

An additional environment variable for time zones is also supported. The TZ variable allows you to compensate for time zone differences when sending and receiving files and when the time stamp on a received file is critical. This variable only affects transfers done with the Z-modem protocol. X-modem transfers are not affected. By default, if no time zone is set, the Datalight utilities COMM and RSZ assume Greenwich Mean Time (GMT).

Setting the TZ variable is not necessary when using all programs. If you are transferring files between two Datalight utilities, the file time stamp is not affected. Some utilities, such as Windows Hyper-terminal, automatically convert the time stamp on the file to GMT and assumes that the receiving program will adjust the time to the correct local time upon receipt.

The syntax for the TZ environment variable is:

```
SET TZ= <abbreviation> +/- value
```

Abbreviation represents any three-letter abbreviation for the chosen time zone. The variable serves as a reminder to the user. For example, if setting the time zone for Pacific Standard Time, the variable could be set as PST; and for Eastern Standard Time as EST. The abbreviation is only a placeholder in the syntax for the TZ variable. There are no incorrect abbreviation choices as long as only three letters are used.

Value represents the number of hours this time zone varies from GMT. For example, the west Coast of the United States is -8 hours relative to GMT. This value may have to be adjusted to compensate for daylight savings time. There should be no spaces between the abbreviation, plus or minus sign, and the value. Some examples are:

```
SET TZ=PST-8
SET TZ=CMT-3
SET TZ=GMT+2
```

If an incorrect format for the time zone is entered, the default of GMT is used.

On a desktop PC running Windows programs, such as Hyper Terminal, determining the time zone is part of the setup of the operating system. However, if COMM is run from a DOS box, the TZ variable still will need to be set on the target machine.

## **COMM Commands**

Enter most commands by pressing an Alt+key combination. Some commands take effect immediately (such as changing the baud rate), while others require further information (such as a filename) before continuing.

If you do not want to execute a command, or want to stop a command while it is running (such as a file transfer), press the Esc key to return to terminal mode.

<b>Command</b>	<b>Description</b>
Alt+B	Sets the baud rate. This command toggles between all the available baud rates. Continue to press Alt+B until the desired baud rate appears on the status line at the bottom of the display.
Alt+C	Clears the display.
Alt+D	Autodial allows entry of number to be dialed. Press Enter to redial the previous number.
Alt+E	Toggle echo (duplex). Press Alt+E to toggle the duplex between full (echo off) and half (echo on).
Alt+H	Hangs up if the modem is capable of hanging up with an ATH0 command.
Alt+P	Toggles through the available parameters allowing settings to be made. Esc cancels the command.
Alt+T	When enabled, pressing the Enter key generates a CR/LF instead of just a CR.
Alt+X	Exits the program. This command does not drop the carrier, so use this command if you need to do MS-DOS operations while online. You can run COMM again without losing the carrier and continue with telecommunications.
PgUp	Sends a file to a remote computer, giving you the option of either Xmodem or ASCII file transfer protocols. Esc cancels at any time during the transfer.



Command	Description
PgDn	Receives a file using the Xmodem or ASCII file transfer protocol. Press Esc to cancel at any time during the transfer.

### **Terminal Emulation**

Currently, COMM supports a subset of the ANSI escape codes that compose the only terminal emulation available. These escape codes, A, B, C, D, H, J, and K, should meet most needs since the emulation includes such features as cursor positioning and erase to the end of the line and/or page.

### **File Transfer Recovery with Zmodem Protocol**

The Zmodem file transfer protocol has the ability to resume a set of file transfers at the point of interruption, such as in the case of a call hang-up or disconnected cable. In the event of a failed Zmodem upload or download, run COMM as follows to resume the file transfer:

1. Press the PgDn key to initiate a receive-file operation
2. Select Zmodem in the file transfer option list.
3. Select Y to enable the ZMODEM crash recovery option.

Auto downloads do not use crash recovery.

---

## **COMMAND**

---

The COMMAND command starts a new command processor.

### **Syntax**

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

### **Remarks**

Use this command to start a new instance of the ROM-DOS command processor, the program that contains the internal ROM-DOS commands.

Starting a new instance of the 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. The environment size may be larger if this is a secondary copy of COMMAND.

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

### **Options**

The /e: *number* switch sets the environment size. *Number* represents the size of the environment in bytes. *Number* must be in the range from 160 to 32,768. All other values are ignored and replaced with the default value of 256. ROM-DOS rounds up the value entered to the nearest multiple of 16.

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

The /p switch causes COMMAND not to exit but to remain permanent. Use the /p switch only when COMMAND is used in a CONFIG.SYS SHELL statement.

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

The /msg option causes all error messages to be stored in memory. This option is recommended only for floppy disk-based systems. ROM-DOS keeps many of its error messages in the nonresident 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 floppy disk, the message will only be available if the boot disk is still in the drive. By using the /msg option, the message is available in memory at all times. The /p option must be used along with the /msg option.

### Examples

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

Causes a new copy of COMMAND to be executed. 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
```

Loads a permanent copy of COMMAND with an environment size of 256 bytes.

---

## DELTREE

---

The DELTREE command deletes one or more directory trees or individual files.

### Syntax

```
DELTREE [/Y] [drive:]path [[drive:]path[...]]
```

### Remarks

/Y prevents DELTREE from prompting before deleting.

[drive:]path indicates the name(s) of the file(s) or directory tree(s) to delete. Wildcards are allowed.

### Examples

Datalight DELTREE deletes one or more directory trees. For example, to delete all files and directories in the tree C:\TEMP, enter

```
DELTREE C:\TEMP.
```

You can also use DELTREE to delete individual files, one at a time, using wildcards. For example, to delete selected files in the current directory, enter

```
DELTREE *.*.
```

DELTREE then prompts you for each file it finds, allowing you to choose whether to delete them.

---

**Caution:** Be careful when using wildcards with DELTREE. DELTREE deletes all specified files and subdirectories regardless of their attributes. Multiple files and/or subdirectories may be specified.

---

---

## DISK2IMG

---

The DISK2IMG command creates a binary image of a drive or disk.

**Syntax**

`DISK2IMG Drive: DestinationPath`

**Remarks**

The drive specifies the source disk.

The DestinationPath is the name and path for the binary image output.

**Options**

None

**Examples**

`DISK2IMG A: C:\FLOPPY.IMG`

This command creates a file FLOPPY.IMG containing a binary image of the A: drive.

---

---

## DISKCOMP

---

The DISKCOMP command compares contents of two floppy diskettes.

**Syntax**

`DISKCOMP drive1: drive2: [/option]`

**Remarks**

The drive parameters specify the diskettes to compare.

No matter which drive letters are used, the user will be prompted to insert the diskettes.

The disks that may be compared are 720KB, 1.2MB, and 1.44MB disks. Both the source disk and the target disk must be of the same type

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

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

DISKCOMP compares only floppy disks, not hard disks.

**Options**

The /1 option compares only the first side of the diskettes.

The /8 option compares only the first eight sectors of each track.

**Examples**

`DISKCOMP A: B:`

Compares the contents of the diskettes in drive A: onto drive B:.

`DISKCOMP A: A:`

Reads the contents of the diskette in drive A:, then prompts for the user to swap diskettes and compares those contents with the new target diskette.

## DISKCOPY

---

The DISKCOPY command copies the entire contents of one floppy disk to another.

### Syntax

DISKCOPY *drive1: drive2: [/option]*

### Remarks

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

The disks that may be copied are 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 unformatted, or has a different format, then 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 copies only floppy disks, not hard disks.

### Options

The /V option verifies the copy after completion.

### Examples

DISKCOPY A: B:

Duplicates the contents of the disk in drive A: onto drive B:. Drive B: must support the same type of disk as drive A:.

DISKCOPY A: A:

Duplicates the contents of the disk in drive A:. You are prompted for swapping the source and target disk as needed to perform the entire copy.

---

## DISPLAY

---

DISPLAY is a device driver that allows you to view international letters and symbols (code pages) on EGA and VGA displays.

### Syntax

device =[*drive:*][*path*]DISPLAY.SYS *codepage*[*fontfilename*]

### Remarks

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

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

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

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

For more information on DISPLAY, see the section on '**Configuring ROM-DOS for International Use**' on page 25.

**Examples**

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

These examples configure 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.

---

## DUMP

---

DUMP is a utility that dumps out the contents of memory or the contents of a file.

**Syntax**

```
dump filename [offset] [length]  
dump /address [length]
```

**Remarks**

filename	The name of the file to dump.
offset	The offset into the file in hexadecimal to start dumping from.
length	The length of bytes in hexadecimal to dump.
address	The hexadecimal address in memory to start dumping from.

**Examples**

```
Dump myfile.txt 0 20
```

This will dump the first 20 hex bytes of MYFILE.TXT

---

## EGA.CPI/EGA3.CPI

---

EGA.CPI and EGA3.CPI are font data files for use with the international video display driver, DISPLAY.SYS. These files contain alternate font sets to display in place of the hardware code page built into the EGA or VGA display monitor.

**Syntax**

Refer to the instructions for DISPLAY.SYS for usage.

**Remarks**

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

## EMM386

The EMM386 device driver enables expanded memory support for systems capable of supporting expanded memory, such as the 386 and higher CPUs. HIMEM.SYS, or another extended memory specifications (XMS) manager, must be installed prior to EMM386.

### Syntax (in CONFIG.SYS)

```
device=[drive:] [path] HIMEM.SYS
device=[drive:] [path] EMM386.EXE {I=xxxx-yyy} {X=xxxx-yyy}
[FRAME=seg[,memK]]
[ROM=xxxx-yyy] [D=xxx] [NOMOVEXBDA]

xxxx and yyy define a range of memory. I= includes UMBs (Upper Memory Blocks) in
that range. X= excludes UMBs in that range. ROM= creates ROM shadowing in that
range.
```

### Remarks

The HIMEM.SYS driver must be loaded for the EMM386 utility to function properly. Datalight's EMM386 supports Expanded Memory Services (EMS), Upper Memory Blocks (UMB), Virtual DMA Services (VDS), Virtual Control Program Interface (VCPI), and ROM shadowing.

MS-DOS checks for the presence of an XMS UMB provider after it loads any device. If such a device is found, it allocates all available UMBs, builds MCB chains within them, and records the fact that an upper memory chain is available in its SDA. However, ROM-DOS does not function in this manner. The Datalight EMM386 driver builds its own MCB chains within its allocated UMBs and sets up the previously-mentioned data structures with the help of the DOS kernel. Consequently, XMS UMBs that are made available by programs such as MS-DOS' EMM386.EXE are not recognized by ROM-DOS.

The *I=xxxx-yyy* option tells EMM386 to include UMBs in the range specified by *xxxx-yyy*. The *X=xxxx-yyy* option tells EMM386 to exclude or **not** to make UMBs in a specified range. EMM386, by default, attempts to make UMBs in the range from C800-F800, excluding areas already occupied by ROM or RAM. 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.

The *FRAME=* option defines the starting segment for four 16K pages that can be mapped in and out at will. The optional *memK* argument specifies how much memory to reserve for EMS. The default behavior is to allow all XMS memory to be used as EMS but also to share it, if not in use, with other processes such as Windows. The EMS memory is simulated using extended memory. The segment must be on an even 16K boundary. For example,

```
DEVICE = EMM386.EXE FRAME=E000,512
```

The above CONFIG.SYS statement allocates 512KB of extended memory to be used as EMS memory, with four Windows into that memory at E000:0. By default, EMM386 uses memory as requested for EMS up to the maximum in the system.

The *ROM=* option specifies a region of ROM to be supported by shadow RAM. This can speed up system performance if the BIOS resides in ROM and does not do its own ROM shadowing. As with all memory ranges, the range must begin and end on 4KB boundaries.

The *D=* option specifies the amount of RAM in kilobytes (specified as a base-10 number) that is to be reserved for a VDS buffer. The default value is zero. Values for the *D=* option must be between 16KB and 256KB and are rounded up to the nearest 4KB.

The *NOMOVEXBDA* option tells EMM386 not to move the Extended Bios Data Area from conventional memory to upper memory.

The *MAX=memk* option specifies the maximum amount of EMS memory in KB.

The *LOW* argument prevents the relocation of EMM386 into the first UMB with sufficient size to hold it. EMM386 remains in conventional memory, using approximately 20KB, but reserving UMB areas for use by other drivers.

The *PS2* argument forces EMM386 to use the PS2-style (port92h) A20 line control.

The *RAM* argument is included for compatibility and has no effect on the function or setup for EMM386.

Datalight's EMM386 contains auto detection of BIOS extensions. EMM386 automatically searches for UMBs in the range of C800-F800, similar to MS EMM386. It supports VCPI, VDS, and all Int 67H functions, shared XMS/EMS, and works with MS Windows. Datalight's EMM386.EXE is significantly smaller than other extended memory managers.

Datalight's EMM386 does not support reallocation of EMS pages. Only four map-able pages are supported. EMS handle 0 is not supported. LIM 3.2 contexts obtained through Int 67h function 47h share eight storage slots with EMS handle names.

### Examples

```
DEVICE = EMM386.EXE I=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. When no *FRAME=* option is supplied, neither EMS nor VCPI services are provided by EMM386.

```
DEVICE = EMM386.EXE I=C800-D7FF I=F000-F7FF FRAME=E000, 1024
```

Maps in RAM from extended memory into C800-D7FF and into F000-F7FF and define Upper Memory Block Regions there. Also, EMS support is allowed with four 16KB pages starting at E000:0 with 1024KB (1MG) worth of 16KB 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 remapped into the ROM address space and write protected. In this example, the BIOS that normally occupies the 64KB block at F000:0h is copied 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 20KB of memory for Direct Memory Access (DMA). Another client, such as a network TSR, can then utilize this memory buffer for disk I/O.

---

## EXE2BIN

---

EXE2BIN attempts to convert an executable file into a com file.

### Syntax

```
EXE2BIN infile outfile
```

**Remarks**

The following must be true for the conversion to be successful: CS=DS=SS (Tiny Model) and there cannot be any segment fixups. Segment fixup addresses are contained in the .exe header so that variable sin the code will be set to contain the segment address which can only be known when DOS loads the file. COM files do not have a mechanism to do fixups.

**Examples**

```
EXE2BIN test.exe test.com
```

Convert the .exe file test.exe into test.com.

## FDISK

The FDISK command initializes a hard disk for ROM-DOS to use. FDISK allows you to specify the number of logical drives that are available using a single or multiple physical hard disks. FDISK can be run from its menu or by command line arguments. FDISK supports drives that require LBA support.

**Syntax**

```
FDISK [drive number] [/3] [/C] [/I# [/N]] [/B|/R|/V] [/S#]  
[/?]
```

**Options**

The *drive number* option specifies the BIOS drive number to use with other command line arguments. For hard drives, this number is 80h, 81h, and so on, depending on the number of hard drives and partitions in the system.

The /3 option enables FAT32 mode

The /B option creates a FAT16 partition and forces the update of the master boot record code.

The /C option causes FDISK to proceed without confirmation messages.

The /I# argument creates a Super-boot partition with an identification (ID) number of #. 98h is the default value for #.

The /N option prevents writing of the master boot record code for the Super-boot partitions.

The /R option refreshes the master boot record code only

The /S# option sets the partition size in MB. The default setting creates a partition as large as possible.

The /V argument displays the current partition information then exits.

**Remarks**

The FDISK command, when entered with no command line arguments, prompts with the command menu:

```
V) View partition(s)  
R) Raw display of partition sectors  
C) Create DOS partition(s)  
P) Create Super-boot partition  
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 the letter that corresponds to your choice. If you make a mistake, press Esc to return to the previous menu.

Initialization of a hard disk is usually performed when it is first setup for use. Do this by using choice C (or possibly N). Choice C displays a description of available disk space and a recommendation for using that available space. Follow these recommendations to initialize a hard disk with as many 2GB partitions as possible.

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

The M choice writes the code for the master boot record. By default, FDISK only writes 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 not rewritten unless you select the M option.

After completing all desired modifications and/or additions, 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 initialized drive must be formatted prior to use. Format only those drives that are new or have been resized. Formatting destroys any existing data.

If no changes were made with FDISK, or you wish to abandon the changes made, select Q to quit.

### Examples

```
FDISK 81 /R /C
```

Refreshes the master boot record code on drive 81h without confirmation.

```
FDISK 80 /I98 /S5 /C
```

Creates a 5MB Super-boot partition with an ID of 98h and without confirmation.

```
FDISK 80 /S
```

Creates a partition that is as large as possible.

---

## FIND

---

The FIND command displays lines, within a disk file, that contain a specified string of characters.

### Syntax

```
FIND [/V] [/C] [/N] [/I] "string" [[drive:][path]filename[ ...]]
```

### Options

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

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

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

The /I option ignores the case of characters

The *string* argument specifies the string of characters to search for. Double quotes are required around the string.

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

### Examples

```
FIND printf junk.c
```

Displays each line in the file JUNK.C that contains the *string* printf.

```
dir | FIND DIR
```

Displays each line in a directory listing that contains a DIR. The command first executes a DOS DIR command with the output piped into the FIND command. The FIND command then displays each line that contains the DIR *string*.

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

Displays a count of the lines in the file MANUAL.TXT that contain the string ROM-DOS.

---

## FORMAT

---

The FORMAT command initializes a disk so ROM-DOS can access files on that disk. A disk must be formatted before ROM-DOS can use it.

### Syntax

```
FORMAT drive: [/V[:label]] [/Q] [/F:size] [/S] [/C]
```

```
FORMAT drive: [/V[:label]] [/Q] [/4 | /7] [/S] [/C]
```

```
FORMAT drive: [/Q] [/4 | /7] [/S] [/C]
```

### Remarks

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

### Options

The *drive*: parameter specifies the drive letter to be formatted.

The /4 switch causes the floppy disk to be formatted as a 360KB disk even if the drive is a 1.44MB, 2.88MB, or 1.2MB drive.

The /7 switch causes the floppy disk to be formatted as a 720KB disk even if the drive is a 1.44MB or 2.88MB drive.

The /C switch causes FORMAT to format 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.

The /F:size option specifies the size of the floppy disk 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.

The /H switch causes the system files not to be hidden or write-protected. This can be used along with the /S option.

The /Q option causes FORMAT to do 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.

The /S switch causes FORMAT to copy the ROM-DOS system files, ROM-DOS.SYS and COMMAND.COM, onto the disk. The file ROM-DOS.SYS is renamed and stored on the disk as files IBMBIO.COM and IBMDOS.COM, which are stored as hidden files, unless the /H option is used.

The /V:LABEL switch causes FORMAT to place a volume label on the disk. If the volume label is not provided on the command line, you are prompted for the volume label once the format is complete.

The /I switch causes FORMAT to run without display of the sign-on message.

---

## HIMEM

---

The HIMEM.SYS device driver manages extended memory and the High Memory Area (HMA) in a 286, 386 or greater, or PS/2 systems. 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 = [drive:][path] HIMEM.SYS [/machine:n] [/A20[+]] [/PS2]
[/CONTROLA20:OFF]
```

### Remarks

The HIMEM driver 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's VDISK. Current versions of VDISK use XMS memory if it is available.

HIMEM recognizes PS/2-style A20 line control and 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). This automatic detection can be overridden with the /Machine:n, /A20, A20+, or /PS2 command line switches in the event that the auto detection on a given system fails.

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

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

/A20+ is similar to /A20 but instructs HIMEM to wait for the A20 line to settle.

/Machine:3 designates support for the Phoenix Cascade BIOS A20 control methods.

Alternately, /CONTROLA20:OFF instructs HIMEM to *not* detect the control method for the A20 line and assumes the A20 line is always on.

The /BIOS switch forces the use of BIOS Int15h, Function 87h, for data transfers to and from XMS memory.

The /QUIET switch forces HIMEM to remove the sign-on message when loading.

### Error Conditions

**No Extended Memory**—An extended memory error condition can occur if the BIOS (via Int 15H, function 88H) notifies HIMEM that there is no extended memory. In this situation, HIMEM displays an appropriate error message and does not install.

**Failure to Control the A20 Line**—When HIMEM installs, it attempts to control the A20 line, which controls access to the HMA. HIMEM first attempts control via the AT method (using the 8259 keyboard control). If that fails, HIMEM then attempts control via the PS/2 method (using I/O port 60H). If both methods fail, HIMEM assumes it can't control the A20 line and displays the message

A20 Control (OFF)

If either of these errors occur, try using the /A20, /A20+, or /PS2 in the HIMEM command line.

Note also that some older programs assume that the machine is a 1MB 8086 and so require that the A20 line to be disabled (OFF) while they run. Current programs typically do not require that the A20 line be disabled.

### Examples

Device = C:\DOS\HIMEM.SYS

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

Device = C:\DOS\HIMEM.SYS /machine:1

Forces the use of the AT-style A20 line control.

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

## KEYB

The KEYB command allows you to alter the keyboard layout for a different language or nationality.

### Syntax

KEYB

KEYB *countryid*

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

### Remarks

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

The *countryid* argument is a two-letter code that specifies which country, region, or language is to become current.

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

The *keyboard filename* argument tells KEYB where to find its data file (KEYBOARD.SYS or KEYBRD2.SYS). When no *keyboard filename* is given, KEYB

first looks for KEYBOARD.SYS in the current directory, 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.

Refer to the table on page 27 for a list of the supported *countryid* codes, along with their default and alternate code pages.

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 U.S. layout at any time by pressing Ctrl+Alt+F1 (Alt+Left-Shift for Russian and Czech Republic keyboards). You can toggle back to the alternate layout by pressing Ctrl+Alt+F2 (Alt+Right-Shift for Russian and Czech Republic keyboards).

### Examples

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

Each of these commands establishes a German keyboard layout. The first and third use 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.

---

## KEYBOARD & KEYBRD2

---

KEYBOARD.SYS and KEYBRD2.SYS are keyboard code page data files for use with the international keyboard driver, KEYB.COM.

### Syntax

Refer to KEYB.COM for usage instructions.

### Remarks

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.

---

## LABEL

---

The LABEL command sets or deletes a disk volume label.

### Syntax

```
LABEL [drive:] [volume string]
```

### Remarks

The *volume string* may be up to 11 characters in length. LABEL only uses the first 11 characters of a volume label. The characters that are acceptable in a *volume string* are the same as those for a filename.

LABEL prompts as follows. Enter the new label and press enter to modify the existing label.

Volume in drive C is xxxxxxxxxxxx

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 is printed.

Delete current volume label (Y/N)?

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

### Example

LABEL a:

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

## MEM

The MEM command displays the used and free memory in your system.

### Syntax

MEM [/B | /C | /R]

### Remarks

Options	Description
/B	Displays each BIOS extension and its size.
/C	Classifies the memory usage.
/R	Does raw dump of the MCB chain.

MEM displays a list of the DOS memory contents, what free space is available, and how much memory is in conventional memory, upper memory, the HMA and extended memory. This program is useful to fine tune the system to have as much free memory as possible for applications.

### Options

The /B option displays BIOS extensions in the range from C000:0 to F800:0.

The /C option shows program, TSR, and device driver sizes.

The /R option shows a low-level DOS listing of MCBs (Memory Control Blocks).

## MODE

The MODE command modifies the operation of the printer, serial port, and active video display.

**Syntax**

```
MODE LPT#[:]=COM#[:]  
MODE COM #:baud[:,parity[:,databits[:,stopbits[:,P]]]]  
MODE video mode  
MODE display lines
```

**Remarks**

The first syntax above redirects line printer output to the serial port.

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

```
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
```

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

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

```
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.
```

The final syntax sets the number of *display lines*. Valid values included L25, L43, and L50.

---

**Note:** A serial port should be initialized before an 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 mono
```

Indicates a monochrome display adapter.

## MORE

---

The MORE command 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 any other DOS *command*. If the *filename* is present, then the file is viewed; otherwise MORE reads from the Standard Input.

Once a screen has been 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.

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	Exit MORE

### Examples

DIR | MORE

Displays a directory one screen at a time.

MORE READ.ME

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

---

## MOVE

---

The MOVE command moves files and renames files and directories.

### Syntax

To move one or more files:

MOVE [/Y | /-Y] [*drive:*][*path*]*filename1*[,...] *destination*

To rename a directory:

MOVE [/Y | /-Y] [*drive:*][*path*]*dirname1* *dirname2*

### Remarks

[*drive:*][*path*]*filename1* specifies the location and name of the file or files you want to move.



*destination* specifies the new location of the file. Destination can consist of a drive letter and colon, a directory name, or a combination. If you are moving only one file, you can also include a filename if you want to rename the file when you move it.

[*drive:*][*path*]*dirname1* specifies the directory you want to rename.

*dirname2* specifies the new name of the directory.

### Options

/Y suppresses prompting to confirm creation of a directory or overwriting of the destination.

/-Y causes prompting to confirm creation of a directory or overwriting of the destination. The /Y option may be present in the COPYCMD environment variable. This may be overridden with /-Y on the command line.

## MSCDEX

The Datalight CD-ROM driver enables CD-ROM drives.

### Syntax

MSCDEX [*options*]

### Remarks

Options	Description
/D:<Name>	CD-ROM device name (default MSC002)
/L:<Letter>	Specifies the drive letter for CD-ROM drive
/M:<Number>	Specifies the number of sector buffers (default 4)
/X	Use extended memory for CD-ROM buffers

A low-level driver from the CD-ROM drive manufacturer interfaces to the actual CD-ROM hardware while the Datalight CD-ROM driver provides the interface between ROM-DOS and that hardware driver. This interface is called the Microsoft CD-ROM extensions (or MSCDEX).

For each hardware driver loaded, use the /D: option to specify the name of that driver. The default name is MSC002.

The /L: option specifies the drive letter to use for the CD-ROM drive. The default driver letter is the next available. To increase the number of available drive letters, use the LASTDRIVE command in CONFIG.SYS.

The /M: option specifies the number of buffers to use to speed up CD-ROM drive access.

The /X option puts those buffers in extended memory rather than conventional memory.

### Examples

```
MSCDEX /D:MSC003
```

Installs the CD-ROM driver using MSCD003 as the device name.

```
MSCDEX /L:G
```

Installs the CD-ROM driver using G as drive letter.

```
MSCDEX /M:6 /X
```

Installs the CD-ROM driver using six buffers allocated in XMS.

## NED

---

The NED editor is a menu-based text editor available for use with ROM-DOS. This editor is similar to other desktop editors but has special functions designed for use in editing C-source and assembly code.

### **Starting the Editor**

To start the editor, enter

```
NED [filename] [filename]
```

NED may be initiated with or without filename arguments. Wildcard file specifications are allowed.

Up to ten files can be entered on the command line. If NED is run without arguments, it loads all files accessed during the last editing session, returning you to the exact position in the file. You can switch between the open files.

You can also enter

```
NED @errfile
```

where *errfile* is the name of your compiler error output file. NED loads all files that had errors and allows you to move between errors.

Once NED is running, you may load files into memory by using the File/Open menu command. File/Reload replaces the current file with a new file or reloads a new copy of the same file. File/Reload confirms before replacing an unsaved file.

### **Basic Editor Operation**

NED uses the standard Windows interface for cut, copy, and paste operations. Del and Shift+Del both move the selected block to the clipboard. There is no true undo command, but Ctrl+V or Shift+Ins may be used to paste the clipboard contents to the current cursor position. Table 1 lists the all the default shortcut keys.

If a search string is all lowercase, NED treats it as a case-insensitive search. If a search string contains any uppercase letters, it is case sensitive. The replacement string is inserted exactly as entered. Repeating a Search command repeats the last Forward or Backward Search operation, not the last Replace operation.

There is one bookmark for all files. Once the bookmark is set, going to the bookmark returns you to the file and position where you set it.

The Indent and Remove-indent (referred to as Undent in the Options/Do Command) commands work on tabs. Indent inserts a tab at the beginning of the current line, or if a block is active, at the beginning of each line in the block. Remove-indent removes the first tab from the current line or from each line in the block. If there are no tabs, Remove-indent has no effect.

Toggle case inverts the case of the current character if no block is active. If a block is active, Toggle case sets the entire block to uppercase if the first character was lowercase, and to lowercase if the first character was uppercase.

Tabs are currently set to 3 for .C, .H, .CPP, .HPP, and .T files. They are set to 8 for all other files.

File/Print prints the current block if there is one, otherwise it prints the current file. NED prompts for a device to print to, which may be a filename. Tabs are expanded to spaces.

The Options/Do Command is intended primarily for debugging. This command allows you to execute any editor command by choosing it from a menu list.

The macro commands (Record Macro/Play Macro) allow you to define a sequence of keystrokes that can be repeated over and over. Select Record Macro (ALT=), enter the keystrokes, then press ALT= again. The macro sequence can be played by selecting Play Macro or by pressing ALT. Keyboard bindings are saved in NED.CFG in the same directory as NED.EXE. NED.CFG also contains the list of active files and positions.

If you record and play a recursive macro, it plays continuously.

If you press an invalid key on a menu, NED operates as if you pressed enter.

If you run out of memory, such as when you have more than 300KB of files open, NED returns to DOS.

### **Remote Editing**

NED operates as a full-screen editor, even through a serial port using ANSI Escape codes. Any communication program capable of emulating an ANSI terminal will work with NED in remote mode.

NED automatically detects if the console is redirected through a serial port, either via CTTY, or when a serial console. NED does not support ANSI key codes, so the use of PC function keys and standard PC cursor keys is supported through control keys. To use the special control keys, copy the NEDREMOT.CFG to the name NED.CFG in the same directory that NED.EXE is run from on the target system. This NED configuration file was created using the “Map a key” function (under the Options menu), and can be modified in the same manner.

Always use the Esc key to get to the menus. Use Ctrl+K to enable/disable blocking mode when selecting text. The remote key mapping is provided in the following list.

Ctrl+A	Left arrow
Ctrl+B	Find backward
Ctrl+C	Copy to clipboard
Ctrl+D	Go to mark
Ctrl+E	Delete to end of line
Ctrl+F	Find forward
Ctrl+G	Go to line number
Ctrl+H	Delete previous character (same as Backspace)
Ctrl+I	Insert tab (same as Tab)
Ctrl+J	Page down

Ctrl+K	Toggle block mode (for cutting to clipboard)
Ctrl+L	Delete the entire line
Ctrl+M	Insert return (same as Enter)
Ctrl+N	Toggle insert/overwrite mode
Ctrl+O	Open a file
Ctrl+P	Toggle through previous 3 positions
Ctrl+Q	Home
Ctrl+R	Search/Replace
Ctrl+S	Right arrow
Ctrl+T	Top of document
Ctrl+U	Page up
Ctrl+V	Insert clipboard at cursor
Ctrl+W	Up arrow
Ctrl+X	Delete to clipboard
Ctrl+Y	End of document
Ctrl+Z	Down arrow
Ctrl+[	Menu/Cancel operation (same as Esc)
Ctrl+]	Brace match
Ctrl+\	Do a command (opens a menu with all NED commands)

### **Troubleshooting Remote NED**

If nothing appears on the terminal screen, check the baud rate of the terminal program, check the serial cable (it should normally be a null-modem cable), and check that the terminal program is set to emulate ANSI escape codes.

In some cases, it is possible for the remote auto-detect to fail. In this case, run the program NEDREMOT prior to running NED. NEDREMOT sets a word at 40:E8h to inform NED to operate remotely.

### **Default Hot Keys**

Many of the editor commands can be accessed directly by pressing key combinations. For example, press Alt+X to exit the editor and save any open files. The following table lists the default hot keys. You can redefine the commands and keys using the Bind HotKey command available on the Options Menu.

Key	Function	Key	Function
Alt+Q	Quit without saving	F1	Help
Alt+X	Exit, saving as needed	F7	Load file into current buffer
Ctrl+A	Search again	F9	Save file

Key	Function	Key	Function
Ctrl+B	Search backward	F10	Exit asking for save as needed
Ctrl+C	Copy the block to clipboard	Left Arrow	Left one character
Ctrl+D	Find the mark	Right Arrow	Right one character
Ctrl+E	Erase to end-of-line	Up Arrow	Up one line
Ctrl+F	Search forward	Down Arrow	Down one line
Ctrl+G	Go to a line number	Home	Beginning of line
Ctrl+I	Indent the block	End	End of line
Ctrl+K	Toggle block mode	Page Up	Up one screen
Ctrl+L	Delete line to the clipboard	Page Down	Down one screen
Ctrl+M	Set the mark	Center (5)	Center the cursor onscreen
Ctrl+N	Read a file into a new buffer	Ctrl+Left Arrow	Left one word
Ctrl+P	Move to the previous position	Ctrl+Right Arrow	Right one word
Ctrl+Q	Quote the next character	Ctrl+Up Arrow	Up one C function
Ctrl+R	Replace text	Ctrl+Down Arrow	Down one C function
Ctrl+S	Switch to the next buffer	Ctrl+Home	Scroll toward beginning of file
Ctrl+T	Toggle the case of character(s)	Ctrl+End	Scroll toward end of file
Ctrl+U	Remove indent from the block	Ctrl+Page Up	Beginning of file
Ctrl+V	Insert the clipboard	Ctrl+Page Down	End of file
Ctrl+W	Delete word to the clipboard	Ins	Toggle Insert/Overwrite mode
Ctrl+X	Delete block to the clipboard	Del	Delete character
Ctrl+Z	Cancel the selected block	Backspace	Delete character backward
Alt =	Start/end recording macro	Ctrl+Ins	Copy block to clipboard
Alt -	Playback macro	Ctrl+BackSpace	Delete word backward
Alt+F7	Previous error	Shift+Ins	Insert the clipboard
Alt+F8	Next error	Shift+Del	Delete block to clipboard

---

## POWER

---

POWER.EXE conserves power on a system that has APM (Advanced Power Management) by shutting down various subsystems (screen, disk drives, etc.) that are 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 AUTOEXEC.BAT, enter

```
[path]power[options]
From CONFIG.SYS, enter
Install=[path]power[options]
```

### Remarks

The system BIOS must support the following APM functions for POWER.EXE to work.

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 that 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.

The (#) argument for each command option defines the number of seconds the device can remain inactive before it is powered down.

### Options

The /C# option sets the inactive time for the COM ports. The default value is two seconds.

The /D# option sets the inactive time for the disks. The default value is 30 seconds. Currently, all disk drives are treated as a single device.

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

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

The /H option displays a user help screen.

The /K options sets the keyboard time.

The /ADV:MIN argument provides the minimum (most responsive) power reduction.

The /ADV:REG argument provides the standard power reduction.

The /ADV:MAX argument provides the maximum (least responsive) power reduction.

The /STD argument provides the standard power reduction.

The /OFF argument turns power management off.

### Example

```
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) and keyboard.

---

## PROTO

---

PROTO creates function prototypes in C language files.

### Syntax

```
PROTO filename [filenames]
```

### Remarks

The start of the function must begin with the first character in a new line. A simple space can cause the function prototype for this function to be overlooked. Which can be useful under certain circumstances. In order to have proto create function prototypes for any assembly files there must be a multi-line comment (see below) with a c-style function description.

```
C language file :
unsigned MultiplyTwoBytes( unsigned char ucValue1, unsigned char
ucValue2) {
    ...
}
Assembly Language (ASM)file:
comment ~
unsigned MultiplyTwoBytes( unsigned char ucValue1, unsigned
char ucValue2) {}
~
Public MultiplyTwoBytes
MultiplyTwoBytes proc
    ...
MultiplyTwoBytes endp
```

The prototypes are only displayed to the screen so the user can redirect them where they want.

Proto takes every option on the command line and assumes that it is a filename.

Therefore you can specify multiple filenames on the command line. Wildcard characters are allowed in the filenames.

### Note

At the time this manual was updated, this utility did not contain long filename support.

### Examples

```
PROTO hello.c hello.asm >hello.h
```

Create a prototype listing in the output file hello.h from the files hello.c and hello.asm.

```
PROTO test*.c >test.h
```

Create a prototype listing in the output file test.h from all of the files matching the name mask of test\*.c in the current directory.

---

## PRINT

---

The PRINT command prints a single file or a list of files.

**Syntax**

```
PRINT [/drive:] [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 you perform other operations).

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

The first time PRINT is used, the operator is prompted with this message 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.

**Note**

At the time this manual was updated, this utility did not contain long filename support.

**Options**

The /B option 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 32KB and the minimum size is 256 bytes. This option is only allowed the first time PRINT is run.

The /C option cancels only the filenames listed after the /C command.

The /F option sets the maximum number of files to be queued up at one time. The default number of files is ten. The minimum is two and the maximum is 32. Support for more files is often useful when using wildcards in filenames. This option is only allowed the first time PRINT is run (or until the next system reboot).

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

The /T option cancels all the files from the print queue (list).

The /H option displays the help screen.

**Examples**

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

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

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

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

```
PRINT /T
```

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

---

## RESTORE

---

The RESTORE command is the complement program to BACKUP. It restores files previously saved with BACKUP to the hard disk.

**Syntax**

```
RESTORE srcdrive [dstpath] [options]
```



**Remarks**

Option	Description
/D	Show directory of files in backup volume(s).
/P	Prompt only if destination file already exists and is a hidden, system, or read-only file, or is marked as changed. Do not prompt on unchanged existing files.
/S	Restore all subdirectories.
/Y	Answer Yes to all prompts.

The *srcdrive* is the drive to restore from and may include an optional file mask. For example, `RESTORE A:*.C` restores only those files ending with `.C` from the backup set in drive A:.

The *dstpath* must be a DOS destination path. If no destination path is given, `RESTORE` assumes the current directory. The destination may contain wildcards (which override wildcards placed in the *srcdrive* option).

**Note**

At the time this manual was updated, this utility did not contain long filename support.

**Options**

The `/D` option just displays a directory of what files are in the backup set, similar to `DIR`. This is useful if you are not sure what files or dates/times are in the backup set.

The `/S` option restores from subdirectories as well.

The `/Y` option is useful when running `RESTORE` from a batch file. Any entire set from another hard drive or network drive can be restored without user input.

**Examples**

```
RESTORE A: /D
```

Displays files in the backup.

```
RESTORE A: D:TEMP\fx*.fh /S
```

Completes restore, with subdirectories, of all files whose filenames begin with `fx` and have a `.fh` file extension.

```
RESTORE B:*.c /S
```

Restores only files with a `.C` file extension to the current directory.

---

## RSZ

---

`RSZ.EXE` is a Zmodem file transfer utility used to transfer files over a serial port to another machine running the Zmodem file transfer protocol. `RSZ.EXE` can be used in place of the `COMM` program and can be started from within a DOS batch file to send and receive files. In addition, `RSZ.EXE` does not require that the system have a video display as does the `COMM` program. `RSZ.EXE` uses approximately 24KB of RAM. The syntax for `RSZ` is:

```
RSZ /Pn /Bn /Hn /Fn /R|[S file1 [file2 ...]]
```

Some examples of `RSZ` usage include:

```
RSZ /R
RSZ /P3 /B115200 /H2 /C /F0 /S a.b lmn o.p z.*
```

### **RSZ Program Options**

All command line options must be separated by a space.

Option	Description
/P	Port number:n = 1, 2, 3, or 4 for COM1 to COM4 (default setting is 1 for COM1).
/B	Baud rate:n = 50,110,300,..115200.  Always N81 when changed. If this option is not specified, program uses the current port parameters.
/Inn	IRQ Numbers – valid IRQ selections are 3 – 15. Default value for COM2 and COM4 is IRQ3; the default value for COM1 and COM3 it is IRQ4.
/H	Handshaking options.  Both sides must use the same value: 0 = none (default), 1 = software, 2 = hardware.
/F	File management options (all files are binary):  0 = skip, 1 = resume, 2 = make duplicate, 3 = replace (default)
/R	Receive files specified by sender using Zmodem protocol.
/S	Send the specified files using Zmodem protocol. Wildcards are allowed.
/V	Verbose switch forces the status display to ON when the console is redirected to a second COM port, and allows the Esc key to break out of a transfer.  Esc will not work when RSZ is using the same COM port as the remote console.
/Q	Do not display/send status information while transferring files. This option is automatically selected when running CTTY.

### **File Transfer Recovery**

For Zmodem file transfer recovery to work after a failed transfer, you must use RSZ with the /F1 option on the receiver command line. This option selects the resume file management option.

### **Time Zones**

An additional environment variable for time zones is also supported. The TZ variable allows you compensate for time zone differences when sending and receiving files and when the time stamp on a received file is critical. This variable only affects transfers done with the Z-modem protocol. X-modem transfers are not affected. By default, if no time zone is set, the Datalight utilities COMM and RSZ assume Greenwich Mean Time (GMT).

Setting the TZ variable is not necessary when using all programs. If you are transferring files between two Datalight utilities, the file time stamp is not affected. Some utilities, such as Windows HyperTerminal, automatically convert the time stamp on the file to GMT and assumes that the receiving program will adjust the time to the correct local time upon receipt.

The syntax for the TZ environment variable is:

```
SET TZ= <abbreviation> +/- value
```

Abbreviation represents any three-letter abbreviation for the chosen time zone. The variable serves as a reminder to the user. For example, if setting the time zone for Pacific Standard Time, the variable could be set as PST and for Eastern Standard Time as EST. The abbreviation is only a placeholder in the syntax for the TZ variable. There are no incorrect abbreviation choices as long as only three letters are used.

Value represents the number of hours this time zone varies from GMT. For example, the west coast of the United States and Canada is -8 hours relative to GMT. This value may have to be adjusted to compensate for daylight savings time. There should be no spaces between the abbreviation, plus or minus sign, and the value. Some examples are:

```
SET TZ=PST -8
SET TZ=CMT -3
SET TZ=GMT+2
```

If an incorrect format for the time zone is entered, the default of GMT is used.

On a desktop PC running Windows programs, such as HyperTerminal, determining the time zone is part of the setup of the operating system. However, if COMM is run from a DOS box, the TZ variable still will need to be set on the target machine.

---

## SHARE

---

The SHARE command installs the capabilities for file sharing and file locking on your hard disk.

### Syntax

```
SHARE [/L:nn] [/u]
```

Or from CONFIG.SYS

```
INSTALL=[drive:][path]SHARE.EXE [/options]
```

### Remarks

The SHARE utility is most commonly used in a network or multitasking 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.

### Note

At the time this manual was updated, this utility did not contain long filename support.

SHARE should not be used under DOS 7.1. With the addition of FAT32 support to DOS 7.1, the internal DOS share tables were overwritten.

### Options

The /L:*nn* option specifies the maximum number of files that can be locked at one time. The default number is 20.

The /U option unloads the share utility and frees the memory. SHARE does not unload if other TSRs have been loaded on top of it. The other TSRs must be unloaded first before trying to unload SHARE.

### Examples

```
SHARE
```

Loads the SHARE program from the command line.

```
INSTALL=C:\UTILS\SHARE.EXE /1:30
```

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.

## SMARTDRV

The SMARTDRV command provides disk and diskette cacheing. This is a memory resident program that speeds up your system's performance by reducing the number of times that an application has to physically access the disk. Datalight's SMARTDRV is fully compatible with Microsoft's documented SMARTDRV API (application programming interface).

### Syntax

```
SMARTDRV [/X] [[drive[+|-]]...] [/U] [/C | /R] [/F | /N] [/V | /Q | /S] [InitCacheSize]
[E:ElementSize] [/B:BufferSize]
```

### Remarks

To cache MSCDEX performance, you must load that driver first.

### Options

Option	Description
drive [+ -]	Sets caching options on the specified drive, which will have write-caching disabled unless you specify a '+'. Specifying a '-' will disable all caching for that drive.
/X	Disable write-behind caching for all drives.
/U	Don't load the CD-ROM caching module.
/C	Writes all information currently in the write-cache to the disk.
/R	Clears the cache and restarts the SMARTDRV program.
/F	Writes the cached data before the command prompt returns (default).
/N	Doesn't write the cached data before the command prompt returns.
/V	Displays the Verbose status messages when loading the cache.
/Q	Does not display any messages while loading, aka Quiet mode (default).
/S	Displays information about the current status of the cache.
InitCacheSize	Specifies the amount of XMS memory (KB) for the cache.
/E: ElementSize	Specifies how many bytes of information to move at one time.
/B: BufferSize	Specifies the size of the read-ahead buffer.

**Example**

The first example enables a write-cached drive C, a read-only cached drives D and Q – the latter being a CD-ROM drive. Note that drive A is also read-cached, even though it was not specified on the command line. The second example shows the results of a status report.

```
SMARTDRV C+ D Q /V 2048 /E:2048 /B:4096
then
SMARTDRV /S
```

---

## SORT

---

The SORT command sorts a text file and displays the output to the standard device.

**Syntax**

```
SORT [/R] [/+n] [[drive1:][path1]filename1] [> [drive2:][path2]filename2]
```

or

```
command | SORT [/R] [/+n] [> [drive2:][path2]filename2]
```

**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 any other DOS *command*.

**Options**

The `/+n` option causes SORT to begin its alphabetical sorting starting at the  $n^{\text{th}}$  position in the string.

The `/r` option 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 in each line or string). The output display is then shown one screen at a time using the MORE command.

---

## STACKDEV

---

The STACKDEV.SYS command is used to increase the number of stacks available for IRQ handlers and Int13h. The standard STACKS= command for ROM-DOS increases stack space for the DOS stacks only.

**Syntax**

DEVICE=STACKDEV.SYS

**Remarks**

Under various conditions, a stack overflow error occurs while booting ROM-DOS, usually during the CONFIG.SYS or AUTOEXEC.BAT file processing. For example, some other program may service an interrupt and not provide its own stack for that purpose. If, inside its interrupt handler, the program consumes a lot of stack space by allocating many automatic variables, calling other interrupt handlers, or by deeply nesting sub-routines, then the interrupt handler may overflow the stack of the program that it interrupted. This situation usually results in a system crash.

The STACKDEV.SYS device driver handles stack support similar to MS-DOS' internal Stacks support. STACKDEV increases the stack space available for IRQ handlers and Int13h by switching to a local stack each time an interrupt occurs. STACKDEV does not allocate the stacks from a pool, instead, each interrupt on the list is given exactly one stack with a fixed stack size of 128 bytes.

STACKDEV differs from the internal STACKS= support with ROM-DOS in that the STACKS= command is used to allocate stack space for ROM-DOS' own internal stacks and does not add additional stack support for IRQs.

**Examples**

DEVICE = STACKDEV.SYS

Loads the STACKDEV.SYS device driver during CONFIG.SYS processing. This driver should be loaded early in the file so that it can handle the IRQ stack needs of other device drivers or programs.

---

## SUBST

---

The SUBST command allows one drive to appear as another drive. This is useful for creating a consistent drive letter when the drive and/or path may change.

**Syntax**

SUBST [*drive1*: [*drive2*:]*path*]

SUBST *drive1*: /D

**Options**

The /D switch disables the substituted drive letter (un-installs it).

The *drive1* argument must be currently unassigned and non-substed.

The *drive2* and *path* arguments must be a valid drive and/or directory.

**Remarks**

SUBST without any options displays the currently substituted drives. The path may be any legal DOS path, including network drives. DOS by default will not reserve room for extra drive letters. The LASTDRIVE command will be required to make available extra drive letters.

**Examples**

SUBST e: b:\subdir

SUBSTitute drive E: to use B:\SUBDIR

SUBST

Displays all SUBSTituted drives.

SUBST e: /D

Drive E: is no longer SUBSTituted.

---

## SYS

---

The SYS command 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. The file ROM-DOS.SYS is renamed and stored on the disk as files IBMBIO.COM and IBMDOS.COM, which are stored as hidden files.

### Syntax

SYS *drive*: [ /*options* ]

### Remarks

Use the SYS command to transfer the ROM-DOS system files to a floppy disk or hard disk. The disk can be a formatted blank disk or can contain files; 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. If the disk already contains system files, installing the new system files deletes the existing files.

The command processor, COMMAND.COM, is also transferred to the disk and does not need to be copied into the same contiguous space as the system files.

You can run SYS three different ways. The first is to boot and run your system with ROM-DOS. When you run the SYS command this way, SYS copies the ROM-DOS system files and COMMAND.COM from the root directory of the default/current disk drive.

The second method is to run SYS from the root directory of a disk drive that has been previously prepared with the SYS command, but isn't booted and running. For example, you can run SYS from a bootable floppy disk to copy the files to the hard disk without actually booting from the floppy disk itself.

The third method uses the file ROM-DOS.SYS, the equivalent of the hidden system files IBMBIO.COM and IBMDOS.COM. ROM-DOS.SYS should be present in the same directory with COMMAND.COM and SYS.COM. These three files can be placed in the root directory or subdirectory on a floppy disk (that need not be booted or bootable), or in a subdirectory on the hard drive. Run the SYS command from the directory where the files reside to transfer the system files to the destination drive.

### Options

The /C option prevents confirmation before transferring system files.

The /H option shows the newly transferred system files on the destination disk.

The /J option prevents display of the sign-on message.

### Example

SYS B:

Copies the ROM-DOS system files to drive B:.

---

## TRANSFER

---

TRANSFER is a file-exchange utility that allows embedded systems to upload and download files over a serial link using the XMODEM protocol. The program running on the host system may be either the COMM terminal program, another of Datalight's serial communications utilities, or another instance of TRANSFER. Refer to '**COMM**' for information regarding the COMM program.

In systems where the BIOS supplies a serial port console, TRANSFER uses BIOS Int 10h function 0Eh and Int 16h functions 0 and 1 to transfer files. In systems having a BIOS that does not supply a remote console, TRANSFER may also be used to transfer files over the same serial link as that used for the CTTY console. A specific serial port (COM1, COM2, and so on) can also be specified for TRANSFER to use for file transfers.

To move a file between systems, run TRANSFER on the target system. Either TRANSFER or COMM may be run on the host PC. If COMM is running on the host PC, press the PgUp key on the PC for COMM to send a file to the target system. COMM prompts for the filename and the protocol for TRANSFER; specify the Xmodem protocol. If you are using TRANSFER on the host PC, select the COM port, the baud rate, and specify either send or receive.

Run TRANSFER as follows:

```
A>TRANSFER [options] filename
```

The *filename* parameter specifies the file to be uploaded or downloaded. A path and drive letter may precede the filename. Wildcards are not allowed in the *filename* parameter.

The *options* for TRANSFER are listed in the following table.

Option	Description
/S	Sends a file.
/R	Receives a file.
/B#	Sets the baud rate. The <i>rate</i> number may be 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, or 115200. The default rate is 9600.  Specifying the baud rate causes TRANSFER to use a serial port rather than the default console.  The default port is COM1; it can be changed using the /COM option.
/BC	Uses the BIOS console to transfer files (instead of CTTY).
/COM#	The /COM# tells TRANSFER not to disable use of the console, and allows the user to select the COM port.  The COM number (#) may be either 1, 2, 3 or 4.  This option (as well as /B#) causes TRANSFER to use interrupt driven serial I/O.  An INS8250/16450/16550 or compatible UART is assumed at standard PC addresses.  If the baud rate option is not used along with the /COM option, the default baud rate is



Option	Description
	9600.
/Q	Prevents the display of the output to the screen.
/IRQ	Allows changing of the COM port IRQ value from the default setting. The default values are IRQ4 for COM1 and COM3; IRQ3 for COM2 and COM4.

When transferring files over the console, TRANSFER uses DOS calls, by default, to allow operation with CTTY. On some smaller systems it may be preferable to use a BIOS console interface to achieve higher throughput. The /BC option tells TRANSFER to use the BIOS' console interface rather than that of DOS.

### **TRANSFER Program Examples**

The following example receives a file via the console. The data of the file is placed on drive B: in a file named FILE.DAT.

```
A:>transfer /r B:file.dat
```

This example sends the file JUNK.ABC over COM4 at 1200 baud, using IRQ 11.

```
A:>transfer /s /B1200 /COM4 /IRQ11 junk.abc
```

The following two examples show the use of TRANSFER on both the host PC and the target system. The file ED.EXE is sent from the host PC to the target system. The file received on the target system is named VI.EXE.

Target system command:

```
A:>TRANSFER /r B:VI.EXE
```

Host PC system command:

```
A:>TRANSFER /s C:\BIN\ED.EXE
```

---

## **TREE**

---

The TREE command displays each subdirectory and, optionally, the files within them for a specified drive.

### **Syntax**

```
TREE [drive:][path] [/F] [/A]
```

### **Remarks**

The TREE command displays the full path of each subdirectory on a specified disk.

*drive*: specifies the drive that TREE displays the subdirectories from. This argument must be specified. The *path* argument can refer to any currently existing subdirectory name.

**Option**

The /F option causes TREE to display the files in each subdirectory.

The /A option will force the use of text characters instead of ASCII graphical characters.

**Examples**

```
TREE C:
```

Displays all subdirectories on drive C:.

```
TREE A: /F
```

Displays all subdirectories on drive A: along with the files within each subdirectory.

---

## UMBLINK

---

UMBLINK builds upper memory blocks (UMBs) which have a distinct MCB chain and may or may not be linked into the DOS MCB chain. UMBLINK can be used on some systems where EMM386 can not be used.

**Syntax**

```
DEVICE=UMBLINK.EXE [x=mmmm-nnnn]
```

**Remarks**

UMBLINK allows for the conversion of various RAM memory regions into DOS UMBs into which device drivers and TSRs may be loaded. UMBLINK is a simple device driver that, once installed, does nothing but filter the Int21h, Function 5803h calls to link and unlink the UMB arena from the normal DOS MCB chain. UMBLINK must appear before any DEVICEHIGH lines in your CONFIG.SYS File.

Unlike EMM386.EXE, UMBLINK.EXE is not a protected mode control program. It finds certain kinds of memory that already exists in the adapter space and links that memory into the DOS upper memory chain. It then becomes resident and, like Datalight's EMM386, becomes the UMB LINK handler for ROM-DOS. The memory types that UMBLINK can recognize are S-ICE.EXE UMBs (eliminating the need for Numega's UMB.SYS), XMS UMBs, and pre-existing RAM that is already either physically or logically addressable within the adapter space at the time the program is run.

**Options**

The x=mmmm-nnnn argument specified a hex segment range of existing RAM to be excluded from the UMB conversion process. Both mmmm and nnnn must be in the range C000:0h to F000:0h.

**Examples**

```
DEVICE=UMBLINK.EXE X=C000 - C800
```

Load UMBLINK.EXE and exclude the range from C000:0h to C800:0h from the UMB creation process.

---

## VDISK

---

VDISK is a device driver that allows you to use RAM as a disk.

### Syntax

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

### Remarks

VDISK partitions some of your computer's memory as a disk. This disk is called a RAM disk or virtual disk and is much faster than either a floppy or hard disk. The RAM disk can use either standard DOS program memory or extended memory (above 1MB) for the disk. Any data on the VDISK is lost when the system power is turned off.

The *size* argument specifies the size of the VDISK in kilobytes. The default is 64KB. 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.

The *secs* argument 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 is used.

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

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

The VDISK driver increases the resident size of DOS.

---

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

---

### Examples

```
device = VDISK.SYS
```

Builds a 64KB RAM disk in DOS memory.

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

Builds a 220KB 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 45KB RAM disk in DOS memory. There are 128 byte sectors and 18 root directory entries.

---

## XCOPY

---

The XCOPY command copies multiple files and, optionally, subdirectories from one disk to another.

**Syntax**

`XCOPY source [destination] [/A | /M] [/D[:date]] [/P] [/S [/E]] [/W] [/Y | /-Y]`

**Remarks**

Use the XCOPY command to copy multiple files and subdirectories, if they exist.

The *source* and *destination* parameters are complete drive-path and file-specification descriptions. If you do not specify a path, XCOPY assumes the default path. If either filename is not specified, then \*.\* is assumed.

The ATTRIB command 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 on page 70 description for instructions.

**Options**

The /A option copies only source files that have the archive bit set in them. The archive is not reset.

The /D: *mm-dd-yy* option copies only those files with a date later than that specified.

The /E option creates subdirectories on the target even if they are empty.

The /M option copies only those source files that have the archive bit set. Once the source file is copied, the archive bit is reset.

The /P option prompts before each file is copied. The prompt appears as follows; enter Y to copy the file:

`C:\COMMAND.COM (Y/N)?`

The /S option copies files in subdirectories of the source directory.

The /V option verifies each write to the disk.

The /W option waits before starting to copy files and prompts with the following message.

`Press any key to begin copying file(s)`

**Example**

`XCOPY \bin\*.exe a: /a`

Copies all files in the BIN subdirectory to the A: drive that have an .EXE extension and that have the archive bit set.

## Mini-Command Interpreter

Some ROM-DOS applications run on standard PCs and require full command interpreter functionality. Some ROM-DOS applications do not require a command interpreter; for instance, when the application is loaded at boot time and runs until the power is turned off.

The mini-command interpreter (MINICMD.COM) suits applications that require a limited and possibly tailored command interpreter. MINICMD.COM contains only the basic functions of COMMAND.COM and is only 4KB as opposed to the 34KB of COMMAND.COM. MINICMD.COM supports program launching, limited batch file execution, echo control, and environment management. MINICMD.COM can also be used to load applications specified in the AUTOEXEC.BAT file at boot time.

/P (permanent) is the only option to MINICMD.COM. The permanent option causes MINICMD.COM to remain loaded at all times and is used by ROM-DOS when initially loading MINICMD.COM. The permanent option causes MINICMD.COM to ignore the EXIT command.

The environment size (/E) option is not implemented. The initial environment size can be specified when building ROM-DOS using the `_env_para` variable in the SYSGEN.ASM file.

### **External Commands**

MINICMD.COM executes applications and batch files in the same manner as COMMAND.COM. MINICMD.COM searches in the local directory for the program or batch file and then uses the directories specified in the PATH environment variable. The search precedence is .COM file, .EXE file, then .BAT file.

### **Internal Commands**

MINICMD.COM supports several internal commands. Each internal command can be enabled or disabled by means of a #define statement in the file MINICMD.H. A description of each internal command is given in the following table.

Command	Description	Example
CD	Displays or changes the current working directory.	CD \database\new Changes the current working directory to DATABASE\NEW.
COPY	Copies one file to another. This is only a single file copy. Wild cards are not supported. You must provide the full path (if not the current directory) and filename for both the source and the destination file.	COPY FILE1.DAT \BIN\BACKUP.DAT Copies one file to the \BIN directory with the new name of BACKUP.DAT.
DEL	Deletes a single file. Wild cards are not supported.	DEL file1.dat Deletes the file named FILE1.DAT.
DIR	Displays a list of the files in a specified or the current directory.	DIR \ROMDOS\DEVSRG Displays the contents of the \DEVSRG directory.
ECHO [ ON   OFF   Text ]	<p>The ECHO command followed by text displays the text to the console.</p> <p>The ECHO command followed by OFF disables echoing commands and prompts. Commands and prompts will not resume until ECHO ON is entered.</p> <p>The ECHO command followed by ON enables echoing of commands and prompts after they were disabled using an ECHO OFF command.</p>	<p>ECHO ON Echoes each subsequent command, causing the prompt and command to be displayed.</p> <p>ECHO OFF Disables echoing for each subsequent command, preventing display of prompt and command.</p> <p>ECHO This is an ECHO Example Displays "This is a ECHO Example" regardless of whether ECHO is on or off.</p>
EXIT	Terminates the mini-command	EXIT

Command	Description	Example
	interpreter. If the /P option was specified to mini-command interpreter during spawning, then the EXIT command has no effect.	
GOTO	Transfers control to another line in the batch file.	GOTO MESSAGE  Moves the control of execution within the batch file to the line labeled :MESSAGE.
IF	Allows conditional execution of commands.	Refer to the ROM -DOS User's Guide.
PAUSE	Causes the batch file processing to stop and wait for the user to press a key.	PAUSE
REM	Causes the subsequent command-line characters to be ignored; usually used to place comments in a batch file.	REM This is a comment.
SET [ARG=[Text]]	Sets, displays, or removes environment variables. SET without arguments displays the current environment variable settings.	SET Path=f:\bin Initializes the PATH environment variable to F:\BIN.
TYPE	Displays a text file on the console.	TYPE File1.dat Displays the file FILE1.DAT on the console.
VER	Displays the version number of MINICMD.COM	VER
[DRIVE]	Specifies a new current drive with a single letter followed by a colon.	A : Sets the current drive to the A drive.

### **Configuring the Mini-Command Interpreter**

MINICMD.COM is configurable and commands can be added to or removed from it. After modifications, MINICMD.COM must be rebuilt using the Borland C compiler and assembler provided in the Datalight SDK.

The *command-table* structure contained in CMDMAIN.C lists the internal commands and the associated C-function called when that command is entered on the command line. Removing an entry from the *command-table* structure makes that command unavailable.

A new internal command can be added by adding the command name, along with the name of the C-function into the *command-table* structure. Each new C-function is placed into a separate file so it can be added or removed easily.

The C-prototype for functions implementing a mini-COMMAND is:

```
void do_cmd(char *args);
```

When an internal command is entered, the function in the *command-table* is called with the text options following the command on the command line. For example:

```
C:\>DIR \bin\abc
```

The function to implement the DIR command is called with the *args* option equal to \bin\abc.





## *Chapter 7, ROM-DOS Server and Client Applications*

---

### **Serial File Server**

With these utilities, the client machine can access some or all of the drives of the server machine through a null modem cable and any serial port. Each of the selected drives on the Server will be mapped to available drive letters on the client machine, in a similar fashion to network drives.

The Serial File Server shares its drives and services the request of the client. The client, or Serial File Link, is able to access and use the remote drives. The serial ports on both systems must be connected via a null modem cable. The Serial File Server utilities work across a standard 3-pin serial cable, similar to other Datalight serial I/O utilities (REMDISK/REMSERV, COMM and TRANSFER). The cable does not require the CTS/RTS DTS/DTR pins.

### **Server Program**

The host or server machine runs the program SERSERV.EXE, which can make any or all of its drives available to the client. The syntax of SERSERV.EXE is

```
SERSERV.EXE [/R] [/N] [/O<list>] [/D<list>] [/C#] [/I#] [/B#] [/W#]
```

Option	Description
/R	Tells the Server not to map removable drives.
/N	Tells the Server not to map network drives.
/O<list>	The program will only map drive letters from this <list>.
/D<list>	The program will not map any drive letters from this <list>.
/C#	Selects the communication port. Available ports are 1 through 4. COM1 is the default port.
/I#	Set the IRQ for the communications port. Valid settings are 3 – 15. Default is IRQ3 for COM2 and COM4, and IRQ4 for COM1 and COM3.
/B#	Selects the baud rate for transmission. Available baud rates are 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, and 115000. The default baud rate is 115000.
/W#	Sets the time-out in the range of 2 to 60 seconds. The default is 10 seconds.

To display a help screen for SERSERV from the DOS prompt, enter

```
SERSERV /?
```

Example: To connect to a normal server with COM1 at 115200 baud, and map all drives except for logical drives (which are never mapped), enter

```
SERSERV
```

The server program can be terminated at any time by pressing the Q key. SERLINK can then no longer access any of the servers drives until the SERSERV program is run again.

Note: This option should not be used when the client has files open, since they will be closed when quitting.

### **Client Program**

The program SERLINK runs on the client system and creates new drive letters for the client. SERLINK uses the next available system drive letters, in order. For example, if the last assigned drive was D:, the first drive SERLINK creates is drive E:. This drive acts like any other network drive, except that it requires the serial port. The syntax for loading SERLINK is:

```
SERLINK.EXE [/R] [/N] [/O<list>] [/D<list>] [/C#] [/I#] [/B#] [/W#]
```

The syntax for unloading SERLINK is:

```
SERLINK.EXE /U
```

Option	Description
/U	Unloads SERLINK from memory, thereby disabling the drive letters and freeing the memory occupied by SERLINK
/R	Tells the Server not to map removable drives.
/N	Tells the Server not to map network drives.
/O<list>	The program will only map drive letters from this <list>.
/D<list>	The program will not map any drive letters from this <list>.
/C#	Selects the communication port. Available ports are 1 through 4. COM1 is the default port.
/I#	Set the IRQ for the communications port. Valid settings are 3 – 15. Default is IRQ3 for COM2 and COM4, and IRQ4 for COM1 and COM3.
/B#	Selects the baud rate for transmission. Available baud rates are 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, and 115000. The default baud rate is 115000.
/W#	Sets the time-out in the range of 2 to 60 seconds. The default is 10 seconds.

To display a help screen for SERLINK from the DOS prompt, enter

```
SERLINK /?
```

Example: To connect to a normal server with COM1 at 115200 baud, and map all drives except for logical drives (which are never mapped), enter

```
SERLINK
```

Example: To install SERLINK from the DOS prompt or from a batch file (such as AUTOEXEC.BAT) at 9600 baud, on COM2, and not map any removable drives, enter

```
SERLINK /B9600 /C2 /R
```

## Remote Disk Program

The remote disk program allows you to access a disk drive on a remote system via a serial cable and standard PC-style (8250UART) serial port. Remote disk allows added flexibility for diskless systems and systems tight on available space.

In a remote disk setup one system, called the server, shares its drives.. The other system, called the client, accesses and uses the remote drives. The serial ports on both systems must be connected via a null modem cable. Remdisk/Remserv works across a standard 3-pin serial cable, similar to other Datalight serial I/O utilities (COMM and TRANSFER). The cable does not require the CTS/RTS DTS/DTR pins.

### Server Program

The server system runs the program REMSERV.EXE that can make a single drive on the server system available to the client. The available drive can be changed at any time by quitting the REMSERV program and then running the program again with a new drive letter. The syntax of REMSERV.EXE is

```
REMSERV.EXE d: [ /Bnnnn] [+|-] [ /COMn] [ /Tnnn] [ /S] [ /H]
```

where d: represents the letter of the drive the server makes available to the client.

Option	Description
/Bnnnn	Selects the baud rate for transmission. Available baud rates are 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, and 115000. The default baud rate is 115000.
+/-	The plus sign (+) specifies packet-style transmission and is recommended for any baud rate over 19200. The default setting is to include + for packet transmission. Use the minus sign (-) to specify polling operation under Windows95.
COMn	Selects the communication port. Available ports are 1 and 2. COM1 is the default port.
/IRQn	Set the IRQ for the communications port. Valid settings are 3 – 15. Default is IRQ3 for COM 2 and COM4, and IRQ4 for COM1 and COM3.
/Tnnn	Sets the time-out in the range of 2 to 3,640 seconds.
/S	Instructs REMSERV to run without any display output.
/H	Selects hardware handshaking for flow control. To select drive B: as the available server drive at 115000 baud, packet transmission, using COM1, enter REMSERV B:  To set drive C: as the server disk at 9600 baud, without packet-style transmission, on COM2, enter: REMSERV C: /B96 00 /COM2

The server program can be terminated at any time by pressing the Esc key. The client can then no longer access the server's drive until the REMSERV program is run again.

### Client Program

The program REMDISK runs on the client system and creates a new drive letter for the client. REMDISK uses the next available system drive letter. For example, if the last assigned drive was D:, REMDISK creates drive E:. This drive acts like any other drive, except that it requires the

serial port. REMDISK.EXE can be loaded by a DEVICE= command in the CONFIG.SYS file can be entered at the DOS prompt. The syntax for REMDISK is:

```
REMDISK [/U] [/H] [/Bnnnn] [+|-] [/Tnnn] [/COMn]
```

Option	Description
/U	Unloads REMDISK from memory, thereby disabling the drive letter and freeing the memory occupied by REMDISK. This option can only be used when REMDISK is installed from the DOS command line. A remote disk installed via CONFIG.SYS cannot be unloaded.
/H	Selects hardware handshaking for flow control.
/Bnnnn	Selects the baud rate for transmission. Available baud rates are 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, and 115000. The default baud rate is 115000.
+/-	The plus sign (+) specifies packet-style transmission and is recommended for any baud rate over 19200. The default setting is to include + for packet transmission. Use the minus sign (-) to specify polling operation under Windows95.
/Tnnn	Sets the time-out in the range of 2 to 3,640 seconds.
/IRQn	Set the IRQ for the communications port. Valid settings are 3 – 15. Default is IRQ3 for COM 2 and COM4, and IRQ4 for COM1 and COM3.
COMn	Selects the communication port. Available ports are 1 and 2. COM1 is the default port.

To install the REMDISK program from CONFIG.SYS at 19200 baud, on COM1, using packet-style transmission, insert the following line in CONFIG.SYS and then reboot the system (remember to include the full path to find REMDISK.EXE if not located in the root directory).

```
DEVICE=REMDISK.EXE /B19200 +
```

To display a help screen for REMDISK from the DOS prompt, enter

```
REMDISK /?
```

To install REMDISK from the DOS prompt or from a batch file (such as AUTOEXEC.BAT) at 9600 baud, without packet-style transmission, on COM2, enter

```
REMDISK /B9600 /COM2
```

To unload the REMDISK installed from the batch file or the DOS prompt, enter

```
REMDISK /U
```

## **Using the Remote Disk**

To use the remote disk, both REMDISK and REMSERV must be running on their respective systems and must use the same baud rate and packet or nonpacket style transmission. After starting both programs, you can access the new drive on the client system. You can change the default directory to this new drive, copy files to and from the remote drive, and also run utilities such as CHKDSK on the drive. The remote drive on the server system can be used as any other drive on the client system.

## **Unloading the Server Remotely**

To unload the remote server, REMSERV, use the utility REMQUIT on the client, REMDISK. REMQUIT takes no command line parameters.

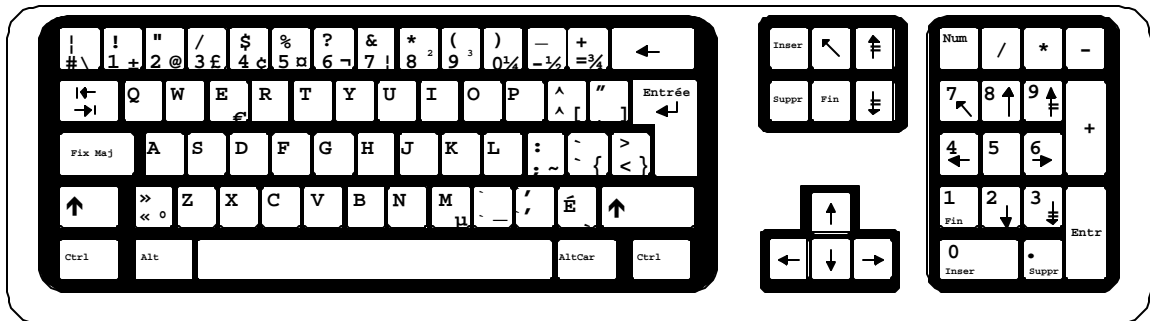
## Chapter 8, ROM-DOS Keyboard Layouts

---

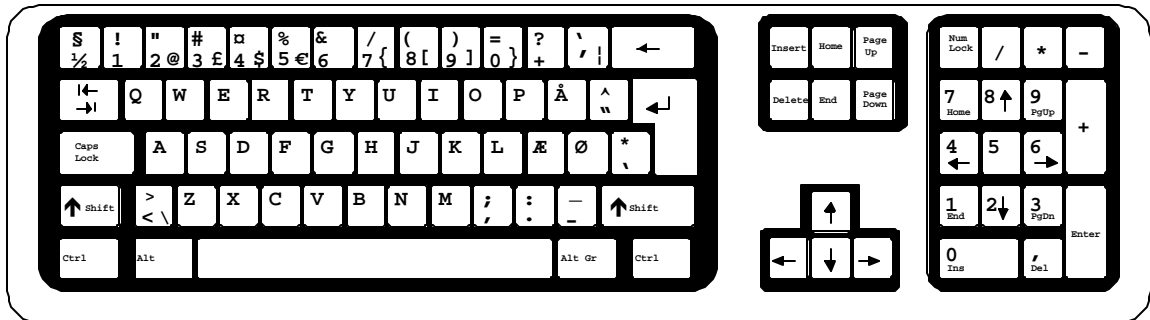
### Keyboard Layouts

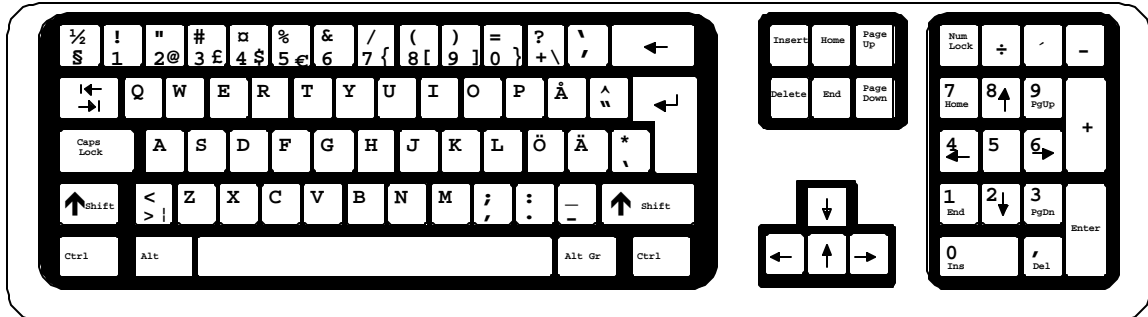
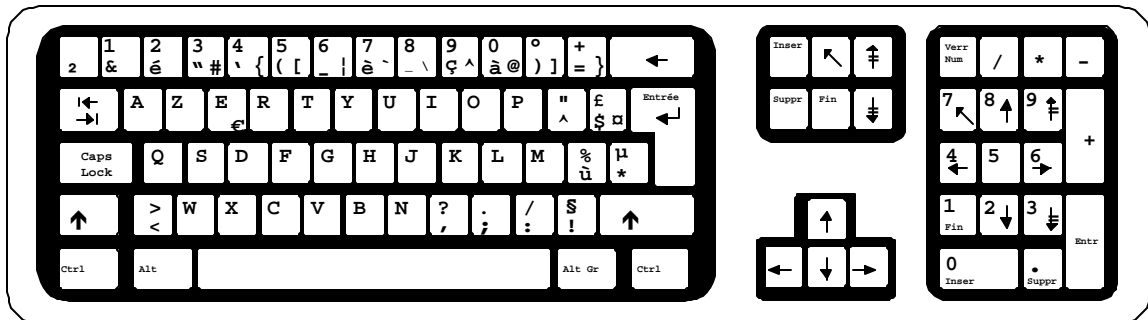
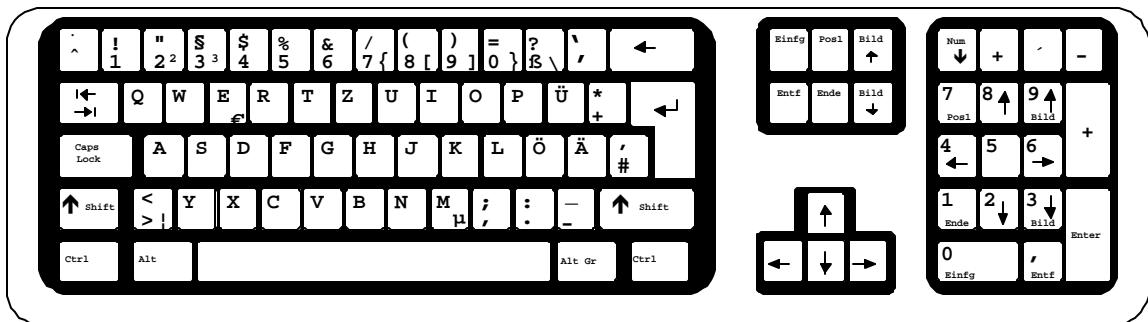
The following keyboard charts represent several countries supported by ROM-DOS. If you need a keyboard layout not displayed in this Appendix, please contact Datalight.

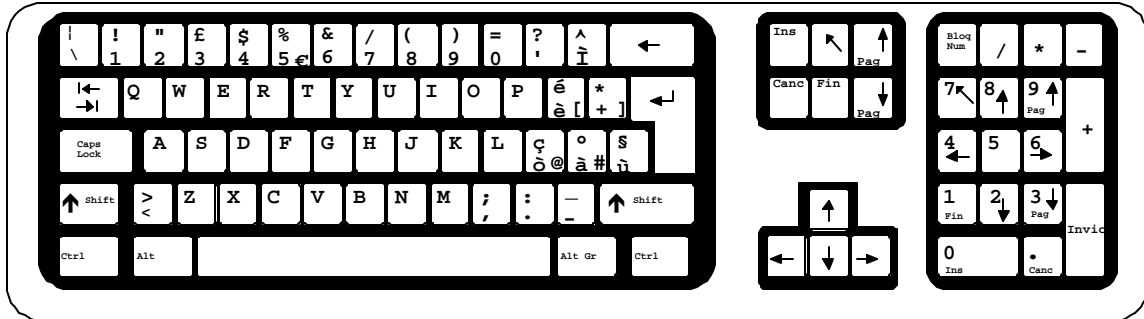
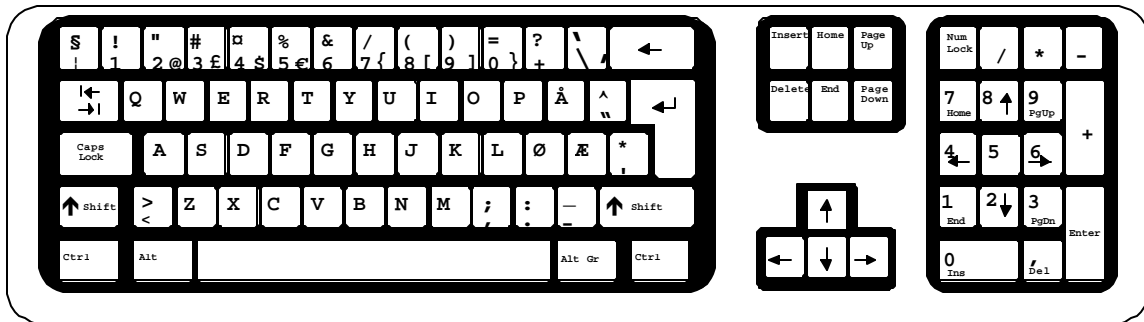
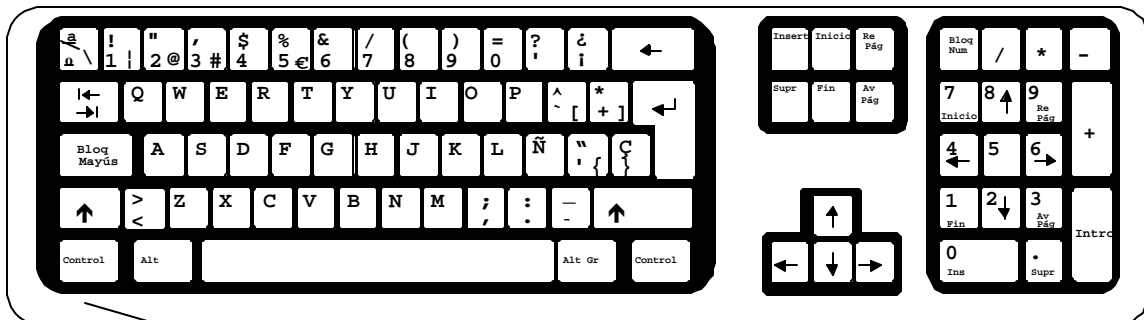
#### Canada

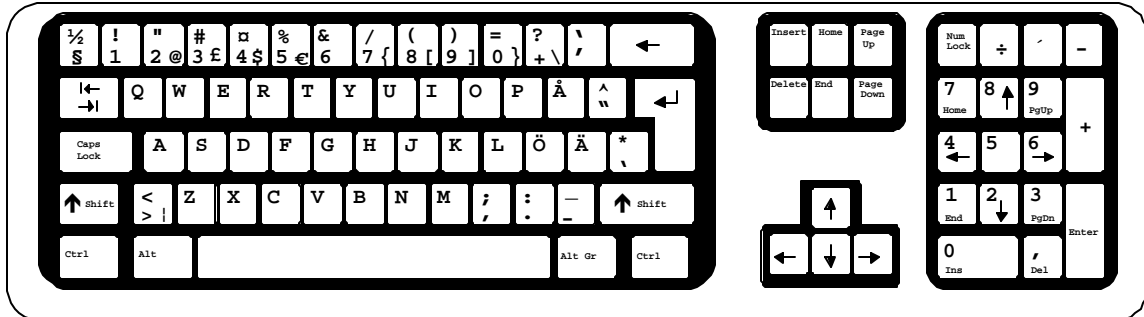
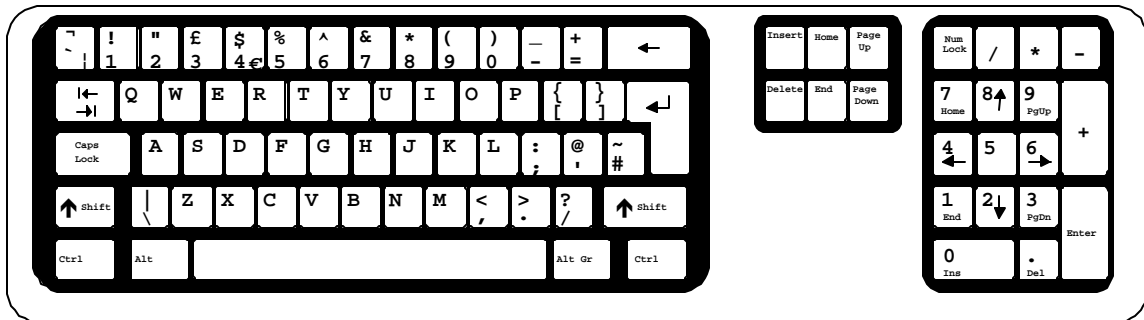
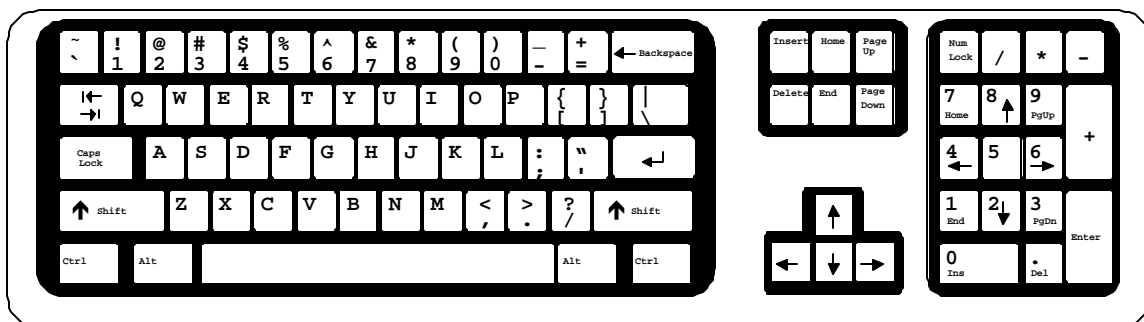


#### Denmark



FinlandFranceGermany

ItalyNorwaySpain

SwedenUnited KingdomUnited States



## ***ROM-DOS Glossary***

---

### **ANSI (American National Standards Institute)**

A group that defines U.S. standards for the information processing industry. ANSI participates in defining network protocol standards.

### **API (Application Program Interface)**

An API is a specification of the methods an application programmer can use to access services provided by a software module. In the case of a network, the API specifies the interface to the network software. A common way of interfacing a terminal emulator to networking software in a PC is to use Interrupt 14h. This is the PC BIOS entry point for serial port support, but when used for networking purposes, the original entry point is reused to provide a similar, but much expanded function. In addition to the native character at a time transfer, block transfers are also offered to increase throughput.

### **Baud**

Literally, the number of times per second the signal can change on a transmission line. Commonly, the transmission line uses only two signal states making the baud rate equal to the number of bits per second that can be transferred. The underlying transmission technique may use some of the bandwidth, so it may not be the case that users experience data transfers at the line's specified bit rate.

### **BIOS**

Basic Input Output System – software that interfaces directly with the hardware.

### **BIOS extension**

A short program that the BIOS recognizes and executes as the BIOS initializes the system.

### **Boot**

Booting is restarting and reloading DOS. A PC can be booted by turning it off and then turning it on or by pressing the Ctrl, Alt, and Del keys simultaneously.

### **Bootable disk**

A system disk that contains the files necessary to start and run the computer.

### **BOOTP (Bootstrap Protocol)**

A protocol a host uses to obtain startup information, including its IP address, from a server.

### **Built-in device**

Built-in device is an input/output device which is part of the DOS kernel.

**DOS**

Disk Operating System – an operating system that relies on disks for file storage.

**DOS kernel**

The DOS kernel is the part of DOS that handles a standard DOS call (Int 21h). It handles opening, reading/writing of files, loads programs, and manages memory.

**FAT**

File Allocation Table – a data table which allows DOS to keep track of file location on the disk so that they can be accessed by programs running on DOS.

**Flow control**

Control of the rate at which hosts or routers inject packets into a network or Internet, usually to avoid congestion.

**LAN (Local Area Network)**

Any physical network technology designed to span short distances (up to a few thousand meters). Usually, LANs operate at tens of megabits per second through several gigabits per second.

**Memory disk**

A disk that uses either ROM or RAM for the disk media. The memory disk has a FAT, directories, and file data.

**MTU (maximum transmission unit)**

The largest amount of data that can be transferred across a given physical network.

**PCMCIA**

Personal Computer Memory Card Interface Association. PCMCIA is a group that defined the standard for a credit-card size card that may act as a memory RAMDISK, ROMDISK, or FLASHDISK. These cards are commonly referred to as PC cards.

**POST**

Power On Self Test – a test performed by the BIOS that checks the computer hardware for problems before fully initializing the computer.

**RAM disk**

A disk drive that uses RAM for the media in place of the usual rotating disk drive.

**ROM-DOS**

Datalight operating system that can be placed in and execute from within a ROM.

**ROM disk**

A disk drive that uses ROM for the media in place of the usual rotating disk drive.

**ROM scan**

The scanning of the ROM area for BIOS extensions performed by the BIOS at initialization time.

**ROM**

Read Only Memory. This is memory that is not changeable once placed in a computer.

**Shell**

The Shell is the command interpreter, usually COMMAND.COM. The shell takes text commands and calls the DOS kernel to implement them.

**System disk**

See Bootable disk.



# *Index*

---

- % symbol
  - how to use in a batch file, 15
- ; sign
  - using in a batch file, 32
  - using in CONFIG.SYS, 32
- ? symbol
  - using to pause CONFIG.SYS commands, 31
- @ sign
  - using to suppress command echo, 32
- A20 line control
  - handling by HIMEM.SYS, 87
- ANSI.SYS, 69
- APM BIOS
  - how it fits in the power management scheme, 97
- Archive files
  - changing with ATTRIB, 71
- ATA.SYS, 70
- ATTRIB command
  - using to manage file attributes, 71
- AUTOEXEC.BAT
  - bypassing commands in, 24
  - CONFIG environment variable, 23
  - troubleshooting commands in, 24
  - using to extend menu items, 23
- BACKUP command
  - using to using to save files, 71
- Batch file commands
  - ., 32
  - @, 32
  - disabling with a, 32
  - conditional execution with IF, 48
  - disabling with REM, 58
  - displaying batch file commands/messages, 44
  - displaying while executing, 44
  - executing a secondary batch file, 34
  - pausing processing of with PAUSE, 56
  - repeating the execution of with FOR, 46
  - SHIFT, 61
  - shifting parameters, 61
  - transferring control with GOTO, 47
- Batch file execution
  - from within another batch file, 34
- Batch file messages
  - creating with REM, 58
- Batch files
  - bypassing commands in, 24
  - clearing the display, 35
  - commands that can be used in, 16
  - disabling the user prompt, 73
  - how to create and use, 14
  - parameters, 15
  - preventing echo with @, 32
  - selecting available user options, 73
  - specifying allowable input keys, 73
  - comment text using, 32, 61
- BIOS
  - needed for advanced power management, 97
- Boot time problems
  - stepping through AUTOEXEC.BAT, 24
  - stepping through CONFIG.SYS, 24
- Bootable disk
  - how to created using SYS, 107
- BREAK command
  - using to expand Ctrl+C operations, 33
- BUFFERS command
  - using to set buffer memory, 33
- Bypassing AUTOEXEC.BAT commands
  - how to, 24
- Bypassing CONFIG.SYS commands
  - how to, 24
- Cache
  - using SMARTDRV to enhance disk speed, 104
- CALL command
  - using to start a batch file, 34
- CD command
  - using to switch directories, 34
- CD-ROM drives
  - using MSCDEX.EXE to support, 93
- Changing the command line
  - using specific keys, 13
- CHDIR command
  - using to switch directories, 34
- CHKDSK command
  - using to fix disk errors, 72
- CHOICE command
  - using to set time delay, 73
  - using to set users prompt/keys, 73
- Clock settings
  - changing with the TIME command, 63
- CLS command
  - using to clear the display, 35

- Code page
  - loading different with DISPLAY.SYS, 26, 80, 81
- Code page numbers
  - for different countries, 25
- Code pages
  - using for different keyboards, 25
- Colors
  - setting text and background with MENUCOLOR, 51
- COM ports
  - changing configuration with MODE, 91
- COMMAND command
  - using to start the command processor, 77
- Command interpreter
  - adding/removing commands, 114
  - loading other than COMMAND.COM, 60
  - using small version with ROM-DOS, 112
- Command line
  - batch file parameters, 15
  - how to edit the contents of, 13
- Command line functions
  - adding/removing from the command interpreter, 114
- Command line prompt
  - using PROMPT to configure, 57
- Command processing
  - bypassing commands with ?, 31
- Command processor
  - description of, 77
  - how to start, 77
- COMMAND.COM
  - loading a different command interpreter than, 60
  - transferring to floppy disk, 107
- Commands
  - summary of, 17
- CONFIG.SYS
  - bypassing commands in, 24
  - CONFIG environment variable, 23
  - setting the processing level, 21
  - settings for multiple users, 21
  - troubleshooting commands in, 24
  - using the COUNTRY command, 38
- CONFIG.SYS commands, 32
  - changing how CONFIG.SYS executes, 63
  - disabling with a, 32
  - configuring ROM-DOS for international use, 38
  - dynamic use of data stacks, 62
  - executing from a secondary file, 54
  - install programs to high memory, 50
  - loading ROM-DOS in high memory, 43
  - loading TSR programs, 50
  - pausing during command processing with ?, 31
  - setting number of open files, 46
  - setting the display colors, 51
  - setting the highest driver letter, 50
  - setting the number of data buffers, 33
  - specifying the number of FCBs, 46
  - the processing of, 21
  - use of the SUBMENU command, 62
  - using INCLUDE for multiple-user setups, 49
  - using to install device drivers, 40, 41, 50
  - using to set for multiple users, 52, 53
  - using to set the command interpreter, 60
  - using to set the NumLock key, 55
- CONFIG.SYS comment text
  - using, 32
- Configuration block
  - using for multiple configurations, 21
- Configuration blocks
  - example of using in CONFIG.SYS, 22
  - including for multiple configurations, 49
  - using for multiple configurations, 49
- Configuring ROM-DOS
  - for multiple users, 21, 52, 53
  - with CONFIG.SYS, 21
- COPY command
  - using to create new disk files, 36
- COUNTRY command
  - using in CONFIG.SYS, 38
  - using to select character set, 25, 26, 38
- COUNTRY identifier
  - for Belgium, 27
  - for Canadian-French, 27
  - for Czech Republic, 27
  - for Denmark, 27
  - for France, 27
  - for Germany, 27
  - for Italy, 27
  - for Latin America, 27
  - for Netherlands, 27
  - for Norway, 27
  - for Poland, 27
  - for Portugal, 27
  - for Russia, 27
  - for Spain, 27
  - for Sweden, 27
  - for Switzerland, 28
  - for United Kingdom, 28
  - for United States, 28
  - for Yugoslavia, 28
- CTTY command
  - using to redirect input/output, 38
- Current directory

- definition of, 11
  - using CD or CHDIR to establish, 34
- Danish keyboard layout, 121
- Data buffers
- setting the number of, 33
- DATE command
- using to display/set the date, 39
- DEL command
- using to erase disk files, 40
- DELTREE command
- using to erase disk files, 78
- DEVICE command
- using to display/set the date, 40
- Device drivers
- about DISPLAY.SYS, 80, 81
  - installable, 40, 41, 50, 80, 81, 82, 87, 93, 97, 111
  - installing in high memory, 41
  - using to support CD-ROM drives, 93
- DEVICEHIGH command
- using to install device drivers, 41
- DIR command
- setting list size, 42
  - setting options with DIRCMD, 42
  - using to list disk files, 41
- DIRCMD environment variable
- using to set DIR command preferences, 42
- Directories
- about the directory system, 10
  - recovering with RESTORE, 100
  - saving with BACKUP, 71
  - using CD or CHDIR to switch to, 34
  - using DELTREE to delete, 78
- Directories and subdirectories
- changing from one to another, 11
  - creating with MKDIR, 53
  - definition of the current directory, 11
  - deleting with MKDIR, 59
  - naming conventions, 11
- Directories/subdirectories
- using TREE to list, 64, 109
  - using XCOPY to create, 112
- Directories/subdirectory
- how to delete, 59
- DIRSIZE environment variable
- using to set DIR command preferences, 42
- Disk, 42
- Disk drives
- checking with CHKDSK, 72
  - creating a volume label, 89
  - creating multiple with FDISK, 84
  - displaying a volume label, 66
  - formatting floppy with FORMAT, 86
  - formatting/initializing with FDISK, 84
  - setting the maximum number of, 50
- Disk drives used in computers
- description of, 2
  - naming conventions, 11
- Disk drives, using SUBST to map drive letters, 106
- Disk files
- checking free disk space, 72
  - definition of, 9
  - displaying directory contents, 41, 64, 109
  - displaying text using MORE, 92
  - displaying the contents of, 65
  - locating text within using FIND, 85
  - locking by loading SHARE.EXE, 103
  - making hidden with ATTRIB, 71
  - making read-only with ATTRIB, 71
  - managing with directories, 10
  - recovering with RESTORE, 100
  - saving with BACKUP, 71
  - sharing by loading SHARE.EXE, 103
  - sorting text within using SORT, 105
  - specifying the number of open with FILES, 46
  - using COPY to create, 36
  - using DEL to delete, 40
  - using DELTREE to delete directories, 78
  - using DISKCOPY to create, 79, 80
  - using ERASE to delete, 44
  - using in place of keyboard input, 14
  - using MOVE to relocate or rename, 92
  - using PRINT to print copies, 100
  - using REN to rename, 58
  - using to receive system output, 14
  - using XCOPY to create, 112
- Disk partitions
- creating with FDISK, 84, 86
- Disk speed
- using SMARTDRV to enhance disk speed, 104
- DISKCOMP
- comparing diskettes with, 79
- DISKCOPY command
- using to create new disk files, 79, 80
- Display colors
- setting text and background, 51
- DISPLAY.SYS
- using to display a different code page, 26
  - using to display code pages, 80, 81
  - using to display international characters, 80, 81
- Displaying file lists
- using DIR to list, 41
- Displaying subdirectories
- using TREE to list, 64, 109

- DOS command
  - using to load ROM-DOS, 43
- DOS references, 3
- Duplicating directories/files
  - using XCOPY to create new, 112
- Duplicating files
  - using COPY to create new, 36
  - using DISKCOPY to create new, 79, 80
- Echo batch file commands
  - preventing with @, 32
- ECHO command
  - using to display batch file messages/commands, 44
- EGA.CPI, 81
- EMM386.EXE
  - using to support expanded memory, 82
- Environment, 57
- Environment variables
  - CONFIG.SYS, 23
  - DIRCMD, 42
  - DIRSIZE, 42
  - PATH, 55
  - setting PATH and PROMPT, 25
  - setting with SET, 59
- ERASE command
  - using to erase disk files, 44
- Erasing files
  - using DEL to delete, 40
  - using DELTREE to delete directories, 78
  - using ERASE to delete, 44
- Euro, 25
- EXE2BIN
  - using to creat .COM files, 83
- EXIT command
  - using to terminate a nested ROM-DOS session, 45
- Expanded memory
  - using EMM386.EXE to support, 82
- Extended memory
  - using HIMEM.SYS to support, 87
- F5 key
  - using to bypass CONFIG/AUTOEXEC commands, 24
- F8 key
  - using to step AUTOEXEC.BAT commands, 24
  - using to step CONFIG.SYS commands, 24
- FCBS command
  - specifying number of file control blocks, 46
- FDISK command
  - using to format a hard disk, 84
- File control blocks
  - specifying the number of with FCBS, 46
- File names
  - how to create using wildcard characters, 11
- Files
  - displaying the contents of, 65
  - fixing errors with CHKDSK, 72
  - locating text within using FIND, 85
  - locking by loading SHARE.EXE, 103
  - making hidden with ATTRIB, 71
  - making read-only with ATTRIB, 71
  - naming conventions explained, 9
  - recovering with RESTORE, 100
  - saving with BACKUP, 71
  - sharing by loading SHARE.EXE, 103
  - sorting text within using SORT, 105
  - using COPY to create, 36
  - using DEL to delete, 40
  - using DELTREE to delete, 78
  - using DIR to list directory contents, 41
  - using DISKCOPY to create entire disk, 79, 80
  - using ERASE to delete, 44
  - using in place of keyboard input to the system, 14
  - using MOVE to relocate or rename, 92
  - using PRINT to print copies, 100
  - using REN to change the name of, 58
  - using to receive display/printer data, 14
  - using XCOPY to create, 112
- FILES command
  - specifying number of file control blocks, 46
- FIND command
  - using to search within disk files, 85
- Finnish keyboard layout, 122
- Flash memory
  - using as a disk drive, 3
- FOR command
  - using to repeat batch file commands, 46
- FORMAT command
  - using to format a floppy disk, 86
- French keyboard layout, 122
- French-Canadian keyboard layout, 121
- German keyboard layout, 122
- GOTO command
  - using to transfer control in batch files, 47
- HELP command
  - using to obtain ROM-DOS help, 47
- Help information on ROM -DOS
  - displaying with the HELP command, 47
- Hidden files
  - making so with ATTRIB, 71
- High memory
  - using EMM386.EXE to support, 82
  - using HIMEM.SYS to support, 87
- High memory area (HMA)



- using the DOS command to load ROM-DOS, 43
- HIMEM.SYS
  - using to support extended memory, 87
- IF command
  - conditional execution of batch file commands, 48
- INCLUDE command
  - using to include configuration blocks, 49
- INSTALL command
  - using to load TSR programs, 50
- International use
  - configuring ROM-DOS for, 25, 38, 88
- Italian keyboard layout, 123
- KEYB
  - using to select a different keyboard/country, 27
- KEYB command
  - using to alter keyboard layouts, 88
- Keyboard input
  - how to redirect from a file, 14
- Keyboard layout
  - Canada, 121
  - Denmark, 121
  - Finland, 122
  - France, 122
  - Germany, 122
  - Italy, 123
  - loading different countries with KEYB, 27
  - Norway, 123
  - setting for different countries, 27
  - Spain, 123
  - Sweden, 124
  - United Kingdom, 124
  - United States, 124
- Keyboard layouts
  - altering with KEYB, 88
- KEYBOARD.SYS, 89
- Keyboards code pages
  - using for different keyboards, 25
- KEYBRD2.SYS, 89
- LABEL command
  - using to create/delete a disk label, 89
- LASTDRIVE command
  - using to set the highest drive letter, 50
- LFN, 9
- LOADHIGH command
  - using to install programs in high memory, 50
- Loading programs
  - installing in high memory, 50
- Long Filenames, 9
  - Wildcards, 12
- MD command
  - using to create a new directory/subdirectory, 53
- MEM command
  - using to display the amount of memory, 90
- Memory configuration
  - using expanded memory, 82
  - using extended memory, 87
- Memory disk
  - creating a RAM disk with VDISK, 111
- Menu blocks
  - using for multiple configurations, 22, 49
- Menu configuration block
  - using to define menu items, 62
- Menucolor
  - example of using in CONFIG.SYS, 22
- MENUCOLOR command
  - using to set text/background colors, 51
- Menudefault
  - example of using in CONFIG.SYS, 22
- MENUDEFAULT command
  - using to set default menu, 52
- MenuItem
  - example of using in CONFIG.SYS, 22
- MENUTITEM command
  - using to specify startup menu, 53
- Mini-command interpreter
  - how to configure, 114
  - using with ROM-DOS, 112
- MKDIR command
  - using to create a new directory/subdirectory, 53
- MODE command
  - using to change a COM port/printer/display, 91
- MORE command
  - using when displaying text files, 92
- MOVE command
  - using to move/rename new files/directories, 92
- Moving files
  - using MOVE to relocate/rename, 92
- MSCDEX.EXE
  - using to support CD-ROM drives, 93
- MTU, 126
- Naming conventions
  - for batch files, 14
  - for disk drives, 11
  - for disk files, 9
  - using wildcard characters, 11
- Naming files
  - using REN to rename existing, 58
- NEDREMOT.EXE editor program
  - using to remotely create/alter text files, 95
- NEWFILE command

- using to switch from CONFIG.SYS, 54
- Norwegian keyboard layout, 123
- NUMLOCK command
  - using to set the NumLock key, 55
- Parameters
  - for batch files, 15
- Partitions for disk drives
  - creating with FDISK, 84
- PATH Command
  - using to set the environment variable, 55
- PAUSE command
  - halting execution of batch file commands, 56
- POWER.EXE
  - using to implement power management, 97
- PRINT command
  - using to print files, 100
- Printer ports
  - changing configuration with MODE, 91
- Printing files
  - using PRINT to output copies, 100
- Processing AUTOEXEC.BAT commands
  - how to, 24
- Processing CONFIG.SYS commands
  - how to, 24
- Programming
  - through the use of batch files, 14
- PROMPT command
  - using to alter the command line prompt, 57
- PROTO
  - creating C language function prototypes with, 99
- RAM disk
  - creating a memory-based disk, 111
- Random Access Memory (RAM)
  - definition of, 2
- RD command
  - using to delete a directory/subdirectory, 59
- Read Only Memory (ROM)
  - definition of, 2
  - using as a disk drive, 3
- Read-only files
  - making so with ATTRIB, 71
- Redirecting input/output
  - using the CTTY command to, 38
- REM command
  - inserting comments in batch files, 58
- REN command
  - using to rename disk files, 58
- RESTORE command
  - using to recover saved files, 100
- RMDIR command
  - using to delete a directory/subdirectory, 59
- RMDIR/RD command
  - using to delete an empty directory, 59
- ROM disk
  - creating with ROMDISK.EXE, 6
- ROM-DOS
  - changing the displayed version number, 66
  - configuring for international use, 25, 38, 88
  - configuring for multiple users, 21, 52, 53
  - configuring with CONFIG.SYS, 21
  - displaying the version number of, 66
  - transferring system files, 107
  - using a command interpreter with, 112
- ROM-DOS memory contents
  - displaying with MEM, 90
- Searching files
  - using FIND to examine contents, 85
- SET command
  - using to set, remove, display environment variable, 59
- SHARE command
  - using to allow file sharing, 103
- SHELL command
  - using to start a command interpreter, 60
- SHIFT command
  - using to shift batch file parameters, 61
- SMARTDRV command
  - using to enhance disk speed, 104
- SORT command
  - using to sort contents of text files, 105
- Spanish keyboard layout, 123
- STACKDEV.SYS, 106
- STACKS command
  - using to set dynamic data stacks, 62
- SUBMENU commands
  - using to define menu items, 62
- SUBST command
  - using to substitute disk drive letters, 106
- SUPERBOOT
  - creating partitions with FDISK, 84
- Swedish keyboard layout, 124
- SWITCHES command
  - using to control CONFIG.SYS execution, 63
- SYS (System) command
  - using to copy system files to a disk, 107
- System files
  - changing with ATTRIB, 71
- System output
  - how to redirect to a file, 14
- System prompt
  - definition and contents of, 13
- Terminate and stay resident programs
  - loading with INSTALL, 50
- TIME command

- using to set the internal clock, 63
- Time settings
  - changing with the TIME command, 63
- TREE command
  - using to list directories/subdirectories, 64, 109
- TSR, 50
- TSR programs
  - loading with INSTALL, 50
- TYPE command
  - using to display the contents of text files, 65
- U.K. keyboard layout, 124
- U.S. keyboard layout, 124
- UMBLINK
  - gaining upper memory blocks with, 110
- UMBLINK.EXE, 110
- VDISK.SYS
  - using to create a RAM disk, 111
- VER command
  - using to display ROM-DOS version number, 66
- VERIFY
  - changing the verify state, 66
- VOL command
  - using to display a disk label, 66
- Volume labels
  - how to create and delete, 89
  - how to display, 66
- Wildcard characters
  - using to create file names, 11
- XCOPY command
  - using to create new directories/files, 112