
ITT XTRA™ Family of Business and Personal Computers

QUICK REFERENCE

ITT ITT INFORMATION SYSTEMS
A Division of ITT SYSTEMS, INC.

QUICK REFERENCE



ITT INFORMATION SYSTEMS
A Division of ITT SYSTEMS, INC.

COPYRIGHT NOTICE

Copyright © 1981, 1982, 1983, 1984, and 1985 by Microsoft Corporation. Copyright © 1984, 1985, 1986 by ITT Corporation. All Rights Reserved Worldwide. This publication has been provided pursuant to an agreement containing restrictions on its use. The publication also is protected by federal copyright law. No part of this publication may be copied or distributed, transmitted, transcribed, stored in a retrieval system, or translated into any human or computer language, in any form or by any means, electronic, mechanical, magnetic, manual, or otherwise, or disclosed to third parties without the express written permission of ITT Corporation.

TRADEMARKS

MicroSoft, GW-BASIC, and MS-DOS are registered trademarks of Microsoft Corporation. SideKick is a registered trademark of Borland International, Inc. ProKey is a trademark of Rossoft, Inc.

DISCLAIMER

The ITT Corporation and Microsoft Corporation make no representation or warranties with respect to the contents hereof and specifically disclaims any implied warranties or merchantability or fitness for any particular purpose. Further, ITT Corporation and Microsoft Corporation reserve the right to revise this publication and to make changes from time to time in the content hereof without obligation to notify any person or organization of such revision or changes.

This product or parts or components for this product were manufactured in the United States. Proper authorizations have been obtained for sales by authorized dealers in and outside the United States. Any purchaser or user who wishes to transfer this product across a national boundary is responsible for obtaining the required authorization from the United States Department of Commerce as well as any authorization that may be required by the country of export.

Address all correspondence to ITT Information Systems, 2350 Qume Drive, San Jose, CA 95131-1812.

Revised October 1986

401114-001

Preface

The contents of this *Quick Reference* comprise three areas:

"MS-DOS Quick Reference" contains a listing of all DOS commands and switch options that are available on MS-DOS® 3.10, Revision C. This chapter also provides a quick summary of how to load your MS-DOS operating system after you have followed the *User's Guide* to hook up your computer system.

"GW-BASIC Quick Reference" contains a listing of all BASIC commands that are available to GW-BASIC®, Version 3.11, as well as a quick rundown on how to load GW-BASIC and run a BASIC program from DOS.

"MS-DOS Error Messages" and "GW-BASIC Error Codes and Messages" each contain a listing of error messages that you may encounter using MS-DOS or GW-BASIC. Each message is followed by an explanation, with suggestions to enable you to continue your processing.

In the back of the booklet you will find a System Configuration Record. When filled in, this form contains all the information you might need for expanding your system or making service queries of any kind.

In addition, extra space is provided in the back of the booklet where you can write notes on your directory or partition structure, files, batch files, programs, variables or any other notes that you may have gleaned from your manual or other sources.

COMMAND SYNTAX

The following notation is used in DOS and GW-BASIC command descriptions to show how commands should be entered.

- Enter any words shown in **BOLD CAPITAL LETTERS**. These words are called **KEYWORDS** and must be entered exactly as shown. You can enter these keywords in any combination of upper/lowercase; DOS and BASIC will convert all keywords to uppercase.
- You must supply the text for any items shown in lowercase italic letters. For example, you should enter the name of your file when *filename* is shown in the format.

- Items in brackets [] are optional or variable. If you want to include this information, do not include the brackets, only the information within the brackets.
- An ellipsis ... indicates that you may repeat an item as many times as you want.
- Include all punctuation where shown (with the exception of brackets), such as commas, equals signs, question marks, colons, slashes, or backward slashes.
- A vertical bar | separates multiple arguments from which you must choose only one; for example, ON | OFF indicates that you may select either ON or OFF as a part of the command.

Examples:

When you see [**d:**], you must write in a drive name and a colon.

All of the external commands are prefaced by [**d:**][\path\]. This syntax provides for the possibility that you might be in a directory that is disconnected from the location of your external commands. If you set up a path to your commands at the beginning of your processing session, you do not have to use this syntax.

You may have to direct a command to a subdirectory several levels away from your location. An example of a COPY command is shown below:

```
copy e:\wordproc\sales\billing.est a:billing.est
```

The syntax for this command would be:

```
COPY [d:][\pathname][\filename].[ext] [d:][filename].[ext]
```

Other conventions given below are used in the MS-DOS and GW-BASIC manuals.

File Naming

filespec = **device: filename**

device: = A: | B: | C: | COM1: | COM2: | LPT1: | LPT2: | LPT3: | KYBD: | SCRN:

filename = **name** [**.ext**]

where **name** is 1 to 8 characters and **.ext** is 1 to 3 characters (in GW-BASIC, **.ext** defaults to **.BAS**).

Variable Naming

The first character must be a letter; additional characters may be any alphanumeric character. The last character identifies the variable type as follows:
% = integer, ! = single precision (default), # = double precision, \$ = string.

MS-DOS Quick Reference

LOADING MS-DOS

After you have followed the *User's Guide* to connect your computer, keyboard, monitor, and printer, your next step is to load MS-DOS into your computer, and run SELECT if you have a European keyboard. How you take this step depends on whether you have a dual-floppy-drive system, or a hard-disk-drive system. The following instructions summarize the steps described in the *User's Guide*.

If You Have a Dual-Floppy-Drive System

The following steps will help you to create a working copy of the system diskette. This diskette will have most of the MS-DOS utilities you may need.

- Insert your system diskette into floppy drive A, close the drive, and turn on the computer.
- Enter or verify the date and time.
- Run the SELECT utility if you have a European system. This utility will enable you to use your keyboard properly, and it will also produce a working copy. If you have a U.S. system, type the following:

diskcopy

Your system diskette is the "source diskette," so all you have to do is insert a "target diskette" in drive B, and press a key to continue. Make sure that you use a blank, unused, or redundant diskette. Follow the instructions displayed on the screen.

- Use the DISKCOPY command to make a copy of the supplemental diskette. This diskette contains other DOS utility programs.

You can use the working copy you just made of the system diskette to restart, or "boot up," your computer. This copy is also called a "bootable" diskette. If you want to use an external command, such as FORMAT, CHKDSK, or DEBUG, you will have to insert the diskette that contains that utility file before you can use the command. You can now create an AUTOEXEC.BAT file on the bootable diskette and copy an application program onto it, if enough space remains on the diskette. If not enough space remains, read on.

Creating Bootable Application Diskettes

You can customize a bootable diskette so that it contains the utilities and applications that you always use. Insert your copy of the system diskette and type:

```
format b:/s
```

Insert a fresh or unused diskette in drive B, close the drive, and press **Enter**. When the formatting is finished you can press **y** if you want to format other diskettes. This newly formatted diskette contains COMMAND.COM and the two hidden files required to start MS-DOS.

Now you can use the COPY command to transfer your AUTOEXEC.BAT file, your word processing files, or other application programs onto this diskette, and you can boot from this diskette just as you would from your system diskette.

If You Have a Hard-Disk-Drive System

Loading MS-DOS onto a hard-disk drive means transferring the system files (including hidden files) and all utilities onto drive C.

- Insert your system diskette into floppy drive A, close the drive, and turn on the computer.
- Enter or verify the date and time. If you have a European system, you must run the SELECT utility so that you can use your keyboard properly. SELECT will produce a working copy of the system diskette. Insert this working copy, restart your computer, and type the following commands. If you have a U.S. system, you are already properly started. Enter the following commands:

```
sys c: Enter
```

```
copy a:*. * c: Enter
```

If your DOS boot drive is a drive other than C, then enter that drive name instead of C.

- Remove the system diskette from drive A. Insert the supplemental diskette into floppy drive A, close the drive, and enter `copy a:*. * c: Enter`.

You can now remove the diskette and boot up with MS-DOS by turning the computer off and on again or by pressing **Ctrl** + **Alt** + **Del**. Use the same command to copy application diskettes to your system (some programs do not allow you to copy them; these systems will always have to be loaded from drive A).

It is recommended that you invoke the HDSETUP command to verify that your partitions are set up as you want them and that the boot partition is selected. If you have changed the setup, you must reboot from your system diskette to install the new settings; then use the FORMAT command to format new partitions. For example, if drive C is your boot drive (the drive that the operating system will search for startup files), type `format c:/s`. For other drives, use this command without the /s option.

SPECIAL-FUNCTION AND CONTROL KEYS

Alt + **Ctrl** + **Del**

Resets system.

Alt + **Ctrl** + **Esc**

Invokes ROM monitor.

Ctrl + **Scroll Lock/Break**

Breaks or interrupts current operation.

Ctrl + **C**

Breaks or interrupts current operation.

Shift + **PrtSc**

Prints contents of screen display.

Ctrl + **P**

Turns printer echo feature on/off.

Ctrl + **Enter**

Continues entering line on next display line.

Ctrl + **J**

Continues entering line on next display line.

Esc

Cancels the current line

Ctrl + **S**

Suspends output to the screen; press any key to continue.

Ctrl + **Num Lock**

Suspends output to the screen; press any key to continue.

Tab

Moves the cursor eight spaces to the right.

Back Space

Deletes characters and backspaces.

Ctrl + **H**

Deletes character and backspaces.

Ctrl + **Alt** + **↓**

or

Ctrl + **Alt** + **Tab** (non-U.S.)

Changes speed of system.

DOS EDITING KEYS**→** or **F1**

Copies one character from the template to the screen.

F2

Copies characters up to the character specified in the template and puts these characters on the screen.

F3

Copies all remaining characters in the template to the screen.

Del

Skips over (does not copy) a character.

F4

Skips over (does not copy) the characters in the template up to the character specified (the opposite of F2.)

Esc

Cancels the line currently displayed on the screen; leaves the template unchanged.

Ins

Enters/exits insert mode.

F5

Makes the new line the new template.

DOS COMMANDS

[d:] [\pathname\] ASSIGN [d=d [...]]

Redirects drive requests.

[d:] [\pathname\] ATTRIB [+/-R] [d:] [\pathname] [\filename] [.ext]

Converts a file to read only.

[d:] [\pathname\] BACKUP [d:] [\pathname] [\filename] [.ext] [d:] [/S] [/M] [/A] [/D:mm-dd-yy]

Backs up hard-disk files on DOS formatted diskettes.

BREAK [ON | OFF]

Sets Break (or Ctrl-C) check.

CHDIR or CD [[d:] \pathname]

Changes the directory of the current or designated drive to a different path; displays current (working) directory.

[d:] [\pathname\] CHKDSK [d:] [filename] [/F] [/V]

Scans the directory of the specified drive and checks for consistency; provides a disk and memory space report.

CLS

Clears the monitor screen.

[d:] [\pathname\] COMMAND [/P] [/C string]

Calls a second-level command processor; *string* represents a nested command or batch file that executes and returns to first level.

COPY [/A] [/B] [d:] [\pathname\] [filename] [.ext] [/A] [/B] [d:] [\pathname\] [filename] [.ext] [/A] [/B] [/V]

Copies one or more files to the same disk or to another disk or device (that is, copy.con [filename] or copy [filename] >PRN). You can give the copies different names if you so desire.

CTTY [device]

Allows you to change the device from which you issue commands (TTY represents the video monitor and keyboard).

DATE [mm - dd - yy]

Displays or sets the date. This date will be recorded in the directory for any files you create or alter.

DIR [d:] [\pathname] [\filename] [/P] [/W]

Displays a list or directory or all the files that are on the disk in the directory specified.

[d:] [\pathname\] DISKCOMP [d:] [d:] [/1] [/8]

Compares the contents of two diskettes.

[d:] [\pathname\] DISKCOPY [d:] [d:] [/1]

Copies the contents of the diskette in the source drive to the diskette in the destination drive.

ERASE or DEL [d:] [\pathname] [\filename]

Erases or deletes any disk file(s) that you designate.

[d:] [\pathname\] EXE2BIN [filename] [d:] [filename] [.ext]

Converts .EXE executable files to binary format. This results in a saving of disk space and faster program loading.

EXIT [d:] [\pathname]

Exits the program COMMAND.COM (the command processor) and returns to a previous level of COMMAND.COM, if a secondary processor exists.

[d:] [\pathname\] FC [/a] [/b] [/c] [/l] [/lb n] [/w] [/t] [/n] [/NNNN] [file1] [file2]

Matches two files and reports differences between them.

[d:] [\pathname\] FIND [/V] [/C] [/N] "string" [[d:] [\pathname] [\filename]

Searches for a specific string of text in a file or files.

[d:] [\pathname\] FORMAT [d:] [/S] [/8] [/1] [/V] [/I] [/4] [/C]

Formats the hard or floppy disk in the specified drive to accept DOS files. Verifies entire diskette for any defective sectors.

[d:] [\pathname\] GRAFTABL

Loads additional character data for color.

[d:] [\pathname] GRAPHICS

Allows you to dump graphics display images to dot-matrix printer via [Shift] + [PrtSc].

HDFMT

Replaced by HDSKDIAG in MS-DOS 3.10, Revision B.

[d:] [\pathname\] HDSETUP

Divides hard disk into distinct partitions to be used by separate operating systems.

[d:] [\pathname\] HDSKDIAG

Physical format utility for hard disk that allows surface analysis and recording of known defects.

[d:] [\pathname\] JOIN [d:] [d:] [\directory name]

OR

[d:] [\pathname\] JOIN

OR

[d:] [\pathname\] JOIN [d:] /D

Forms one directory structure from two drives united under a subdirectory that has no files.

[d:] [\pathname] xxKBD

Replaces U.S. keyboard program with alternate configuration. See *MS-DOS Command Reference* for complete listing.

[d:] [\pathname\] LABEL [d:] [\pathname\] [volume label]

Allows you to create, change, or delete a disk label.

[d:] [\pathname\] LINK

OR

[d:] [\pathname\] LINK [filenames] [/switches]

OR

[d:] [\pathname\] LINK @ [filename]

Combines several object modules into one load module or run file (.EXE or executable file).

MKDIR [d:] [\pathname]

Makes a new directory.

[d:] [\pathname\] MODE

OR

[d:] [\pathname\] MODE LPTn [: [c] [,I] [,P]]

OR

[d:] [\pathname\] MODE [mode] [, [s], [T]]

OR

[d:] [\pathname\] MODE COMn [: [baud], [parity], [databits], [stopbits], [P]]

OR

[d:] [\pathname\] MODE LPTn : = COMn

Sets the parameters for the printer, monitor, and asynchronous communication.

PATH [[d:] [\pathname] ; [d:] \pathname] . . .]

Sets a command search path.

[d:] [\pathname\] PRINT [/D] [/B] [/U] [/M] [/S] [/Q]

[/T] [/C] [/P] [[d:] [\pathname] [\filename] [.ext] . . .]

Prints a text file on a line printer while you are processing other DOS commands (this feature is typically called background printing).

PROMPT [*prompt-text*]

Changes the DOS command prompt.

[d:] [\pathname\] xxPRT LPTn

Loads printer driver for correct printing of DOS messages; for non-U.S. print wheels. See *MS-DOS Command Reference* for complete listing.

[d:] [\pathname\] RECOVER [d:] [\pathname] [\filename]

Recovers a file or an entire disk containing bad sectors.

REN [AME] [d:] [\pathname\] [filename] [filename]

Changes the name of the first filename that you designate to the second filename.

[d:] [\pathname\] RESTORE [d:] [d:] [\pathname] [filename] [.ext] [/S] [/P]

Restores to the hard disk the diskette files previously saved with the BACKUP command.

RMDIR or RD [d:] [\pathname]

Removes a directory from a hierarchical directory structure.

[d:] [\pathname\] SELECT [xxx] [yy]

Creates modified copy of DOS system diskette with default country and keyboard layout of your choice.

SET [string = string]

Sets one string value equivalent to another string for use in later programs.

[d:] [\pathname\] SHARE [/F:filesize] [/L:locks]

Loads support for file sharing and locking when networking is active.

[d:] [\pathname\] SPEED [F | N]

OR

[d:] [\pathname\] SPEED K = nn

OR

[d:] [\pathname\] SPEED

Sets the processor to fast or normal mode.

[d:] [\pathname\] SPINDOWN

Prepares the hard disk to be relocated.

[d:] [\pathname\] SUBST [d:] [d:] [\pathname] [/D]

Allows programs and the user to access a subdirectory as if it were a separate drive.

[d:] [\pathname\] SYS [d:]

Transfers DOS system files from the disk in the default drive to the disk in the drive specified by [d:].

TIME [hh [: mm:ss.cc]]

Displays or sets the time of day.

[d:] [\pathname\] TREE [d:] [/F]

Displays the disk volume label and each of the subdirectories on the designated drive.

TYPE [d:] [\pathname] [\filename]

Displays contents of a file on the video screen.

VER

Prints DOS version number.

VERIFY [ON | OFF]

Turns the verify switch on or off when writing to disk.

VOL [d:]

Prints volume identification name or number, if it exists.

BATCH COMMANDS**ECHO [ON | OFF message]**

Turns batch file echo feature on/off.

FOR % c IN (set) DO [command]

Batch processing command extension.

FOR % c IN (set) DO [command]

Interactive processing command extension.

GOTO [label]

Batch command extension.

IF [condition] [command]

Batch command extension.

PAUSE [comment]

Causes the system to pause and wait for a keystroke from the keyboard.

REM [comment]

Displays remarks that are on the same line as the REM command in a batch file during execution of that batch file.

SHIFT

Allows access to more than ten replaceable parameters in batch file processing.

FILTERS**[command] | MORE**

Displays output on the monitor one screen at a time.

SORT [/R] [/ + n] [oldfilename > newfilename]

Sorts ASCII data forward or backward and then writes the results to your monitor screen or files.

CONFIGURATION COMMANDS

These commands are included in the CONFIG.SYS file according to system needs. Refer to *Getting Started with MS-DOS*.

BREAK = [ON | OFF]

Instructs DOS to check for control break when running a program that requests DOS functions.

BUFFERS = [number]

Determines the number of disk buffers (1 to 99) that DOS will allocate in memory.

COUNTRY = [number]

Sets date and time format, and currency symbol and decimal separator to suit a particular country.

DEVICE = [filename]

Installs designated device driver file (ANSI.SYS, IGA.SYS, TIMER.SYS, and VDISK.SYS).

DEVICE = [d:][\pathname\] VDISK.SYS [bbb] [sss] [ddd][/E:m]

Partitions a portion of system memory into simulated disk drives.

FCBS = [m][n]

Determines the number of file control blocks that DOS can keep open at the same time.

FILES = [number]

Specifies the number of files (1 to 99) that can be open at the same time.

LASTDRIVE = [d:]

Sets maximum number of drives that you can access (a to z).

SHELL = [filename]

Loads designated command processor rather than COMMAND.COM.

EDLIN COMMANDS**[n] A**

Appends lines.

[line], [line], [line], [count] C

Copies lines.

[line] [,line] D

Deletes lines.

[line]

Edits line.

E

Ends edit.

[line] I

Inserts lines.

[line] [,line] L

Lists lines.

[line], [line], [line] M

Moves lines.

[line] [,line] P

Pages 23 lines.

Q

Quits edit.

[line] [,line] [?] R [string 1] [F6] string 2

Replaces text.

[line] [,line] [?] S [string] [Enter]

Searches text.

[line] T [filename]

Transfers lines.

[n] W

Writes lines.

DEBUG COMMANDS**A [address]**

ASSEMBLE—Assembles statements.

C [range] [address]

COMPARE—Compares memory.

D [range]

DUMP—Displays memory.

E [address] [list]

ENTER—Changes memory values at specified address.

F [range] [list]

FILL—Fills addresses in the range with values in the list.

G [= address [address . . .]]

GO—Executes program in memory with optional breakpoints.

H [value] [value]

HEX—Performs hexadecimal add/subtract.

I [value]

INPUT—Inputs/displays input byte from designated port.

L [address [drive [record] [record]]]

LOAD—Loads file or absolute diskette sectors into memory.

M [range] [address]

MOVE—Moves block of memory to specified address.

N [filename] [filename . . .]

NAME—Defines files and parameters.

O [value] [byte]

OUTPUT—Sends byte to designated output port.

Q

QUIT—Ends DEBUG program.

R [register-name]

REGISTER—Displays CPU registers/flags.

S [range] [list]

SEARCH—Searches for characters.

T [= address] [value]

TRACE—Executes and displays.

U [range]

UNASSEMBLE—Unassembles instructions.

W [address [drive [record] [record]]]

WRITE—Writes file or absolute disk sectors.

FXP COMMANDS**/xxx** Allocates xxxK bytes of memory.**/ULM** Uses lower memory as buffers.**/CPS** Checks the program status.**/HLP** Displays Help information.**/DPB** Disables the print buffer.**/EPB** Enables the print buffer.**/CPB** Clears the print buffer.**/SAP** Suspends all printing.**/RAP** Resumes all printing.**/LP1** Buffers line printer 1.**/LP2** Buffers line printer 2.**/LP3** Buffers line printer 3.**/DF1** Disables floppy 1 buffering.**/DF2** Disables floppy 2 buffering.**/DHD** Disables hard-disk buffering.**/EF1** Enables floppy 1 buffering.**/EF2** Enables floppy 2 buffering.**/EHD** Enables hard-disk buffering.**/CF1** Clears the floppy 1 buffer.**/CF2** Clears the floppy 2 buffer.

- /CHD** Clears the hard disk buffer.
- /EBW** Enables buffered writes.
- /DBW** Disables buffered writes.

MS-DOS Error Messages

If a disk or device error occurs at any time during a command or program, DOS returns an error message in the following format:

```
yyy ERROR WHILE I/O action ON  
DRIVE x  
Abort, Retry, Ignore?
```

In this message, yyy may be one of the following:

- WRITE PROTECT
- BAD UNIT
- NOT READY
- BAD COMMAND
- DATA
- BAD CALL FORMAT
- SEEK
- NON-DOS DISK
- SECTOR NOT FOUND
- NO PAPER
- WRITE FAULT
- READ FAULT
- DISK

I/O-action may be either of the following:

- READING
- WRITING

Drive x indicates the drive in which the error has occurred. DOS waits for you to enter one of the following responses:

- A (Abort)** Terminate the program requesting the disk read or write.
- R (Retry)** Repeat the operation. This response is to be used after you have corrected the error (such as with NOT READY or WRITE PROTECT errors).
- I (Ignore)** Ignore the bad sector and pretend the error did not occur.

Usually, you will want to attempt recovery by entering responses in this order:

- R (to try again)
- A (to terminate program and try a new disk)

Other error messages listed below may appear in response to your commands. Most of those that you will encounter indicate some kind of typing error, incorrect application of syntax, or incorrect use of drive, directory, and pathname designations. Others will indicate conditions that require attention, such as removing a write-protect tab or formatting a diskette. You should refer to the *MS-DOS Command Reference* to be sure that your command conforms to the syntax described there.

Some messages, such as CHKDSK or EXE2BIN messages, will be of interest only to programmers. However, if CHKDSK gives you messages you don't usually see, such as "lost clusters," it may be time to back up your files and reformat the disk.

Error Messages

Access denied

You tried to delete or write to a protected file. Use ATTRIB to release the file from its protected state.

All specified file(s) are contiguous

CHKDSK message: Refer to CHKDSK. May require file backup and disk format.

Allocation error, size adjusted

CHKDSK message: Refer to CHKDSK. May require file backup and disk format.

Bad command or filename

You have entered a command that DOS does not recognize. DOS was unable to load the executable file (.COM, .EXE, or .BAT) in the specified directory. Check to be sure that you have no misspellings in the command.

Cannot CHDIR to root

Processing cannot continue

CHKDSK message: The disk you are checking is bad. Try restarting DOS and RECOVER the disk.

Cannot CHKDSK a SUBSTed or ASSIGNed drive

Since a drive created by SUBST is not a physical drive, CHKDSK cannot obtain information from it. CHKDSK is disabled when you execute it for a drive that is redirected by ASSIGN.

Cannot edit .BAK file—rename file

You cannot use EDLIN to edit a file with an extension of .BAK. If you need to edit the .BAK file, you must either rename the file with a different extension (using REN) or copy it to another file with a different extension.

Cannot reassign network drive

This message will occur if you are using a network drive and have tried to use the ASSIGN or SUBST commands.

Cannot SUBST a network drive

This message will occur if you are using a network drive and have tried to use the SUBST command.

Country code missing

You have entered the SELECT command without specifying the country code. Refer to SELECT in *MS-DOS Command Reference*.

Directory not empty

JOIN message: This message is displayed if the subdirectory name under which you are uniting two drives contains files. You can simply use a new name in the JOIN command, and DOS will accept it as a new subdirectory.

Disk error reading FAT

CHKDSK message: Refer to CHKDSK. May require file backup and disk format.

Disk error writing FAT

CHKDSK message: Refer to CHKDSK. May require file backup and disk format.

Disk full. Edits lost. or Disk Full—file write not completed

This is an EDLIN message. Every time you end a session of EDLIN with the E command, a backup (.BAK) is created of your working file. This is displayed if you have allowed the size of your file to grow into the space required for a .BAK duplicate of that file. To prevent this, type dir [your filename]. The screen will display:

```

your filename          yyyy      date      time
                    1 File(s)    xxxxxx  bytes free

```

Double the number represented by yyyy. If that number is close to or greater than xxxxxx bytes free, then you need to delete some files to provide more space, or use a fresh disk.

Diskette is not a backup diskette

You have used the BACKUP switch /A to append files to those already backed up on your diskette and then inserted a nonbackup diskette. You can see if your diskette

has been used as a backup by inserting it in the floppy drive and typing `dir a:`. The first line will be `BACKUPID @@@`. If the last line shows some bytes free, then you can use it as a backup diskette.

Entry Error

EDLIN message: This means that the last command typed contained a syntax error. Verify that your command has correct syntax.

Errors found, F parameter not specified

Corrections will not be written to disk

CHKDSK message: You must specify the `/F` switch if you want the errors to be corrected by CHKDSK.

`fc:` cannot open *filename* - no such file or directory

You tried to compare two files, one of which does not exist. Verify the file names in your command.

File allocation table bad for drive x

The copy in memory of one of the allocation tables has pointers to nonexistent blocks. The disk may have been incorrectly formatted or was not formatted before use. If this message persists, you need to format the disk before it can be used.

File cannot be converted

EXE2BIN message: The input file is not in the correct format; that is, the `.COM` or `.SYS` file does not reside in one contiguous 64K segment.

File cannot be copied onto itself

You have executed a `COPY` command without specifying a target filename. The correct syntax is:

```
copy [oldfile] [newfile]
```

File creation error

DOS cannot create the output file. Run CHKDSK to determine what condition caused the error. If the directory is full, you might have to back up some files and delete them from the disk. Or you might have tried to create more files than you allowed in the drive created by `VDISK.SYS`. You might also have used `>` to redirect a screen output to a file that was protected by `ATTRIB`. This message may also occur if you have directed a command to a drive that does not exist or has not been formatted.

If this message resulted from an EDLIN command, first check to be sure that the directory has enough space to create a temporary file. Also, make sure that the file does not have the same name as a subdirectory in the directory where the file to be edited is located.

File error

This occurs if you continue to ignore an error message such as Write protect where a specified command could not complete successfully on a file. You may have to take some remedial action, such as removing the write-protect tab from a diskette.

File is READ-ONLY

You have attempted to use EDLIN to open a file that was converted by `ATTRIB` to read only. You will have to use the command `TYPE` or a word processor to read the file. If you want to write to the file, you will have to execute the `ATTRIB` command to release the file from its protected state.

Filename contains non-contiguous blocks

CHKDSK message: Refer to CHKDSK. May require file backup and disk format.

Filename is cross linked on cluster

CHKDSK message: Make a copy of the file you want to keep, and then delete both files that are cross-linked.

Filename must be specified

You must specify a filename when you start EDLIN.

File not found

The file is not on the disk specified, or you have misspelled the filename. Make sure that you are in the correct drive or subdirectory.

File not found (filename)

FIND message: The filename you have specified does not exist or FIND cannot locate it. Make sure that you are in the correct drive and that you have accurately typed in the file name.

Find: read error in (filename)

An error occurred when FIND tried to read the file specified in the command.

First cluster number is invalid entry truncated

CHKDSK message: Refer to CHKDSK. May require file backup and disk format.

Fixups needed - base segment (hex)

EXE2BIN message: Refer to EXE2BIN. The source (`.EXE`) file contained information indicating that a load segment is required for the file. Specify the absolute segment address at which the finished module is to be located.

Has invalid cluster, file truncated

CHKDSK message: Refer to CHKDSK. May require file backup and disk format.

Incompatible system size

This message occurred if you were performing a SYS to a diskette which contained a copy of DOS smaller than the one being copied. You should format a fresh diskette using `FORMAT /S` before copying other files to it.

Incorrect DOS version

You are trying to run a utility that requires a different version of DOS than that resident in your system. Make sure that you have installed your most recent version. If the message persists, you should obtain a more recent version of DOS.

Incorrect number of parameters

You may have left something out, such as a needed entry or a space, or you have inserted an unwanted space. For example:

```
JOIN F:C:NEW2
```

should be corrected to:

```
JOIN F: C:NEW2
```

Incorrect parameter

SHARE message: One of the options you specified is wrong.

Insufficient disk space

There is not enough disk space to copy a file or to create a new one. You may need to delete some files. If you are copying to a diskette, you can transfer more files by using `BACKUP`.

Insufficient memory**Processing cannot continue**

There is not enough memory resident on your system to run the utility or application program. You will have to obtain more system memory, either on the motherboard or on an expansion board. This message could be displayed if you have memory taken up by programs such as SideKick® or ProKey™. Once loaded, these programs are resident on your computer memory until the computer is turned off or rebooted. If you need the memory, make sure that you do not have these or similar programs installed in your `AUTOEXEC.BAT` file. Reboot your computer; then load application programs as you have need.

Insufficient memory space

Not enough memory is resident on your system to run the utility or application program. You will have to obtain more system memory, either on the motherboard or on an expansion board.

Insufficient room in root directory

No more directory entries are available in your root directory. You should create sub-directories and transfer your files into them from the root directory.

Insufficient room in root directory

Erase files in root and repeat `CHKDSK`

CHKDSK message: CHKDSK cannot continue process until you delete some files in the root directory.

Intermediate file error during pipe

This message may occur after you have used the `MORE` or `SORT` filter in a command. If the space remaining on the drive is insufficient, you should delete some files to make room. These filters create temporary files. If the command in which you used a filter deleted these files, then the command would fail.

Invalid current directory

Processing cannot continue

CHKDSK message: Restart the system and rerun `CHKDSK`.

Invalid date

The calendar date setting has been incorrectly entered. Be sure to use the correct delimiters, such as slashes or hyphens. If you are using a non-U.S. keyboard, you should use the same delimiters that are shown on the screen after you enter the `DATE` command. Also, verify that the day, month, and year are in the correct order.

Invalid directory

You may have entered incorrect pathname or syntax. You may also have specified a subdirectory that does not exist or misspelled the name. Verify that all the command parameters are correct.

Invalid drive in search path

This message may result from an incorrect `PATH` setup, if you specified a drive that does not currently exist. In this case, you should verify the path. This message will also be displayed if you have implemented a command—particularly `JOIN`—that might be in conflict with the search path specified in the `PATH` command. If you use `JOIN`, you may have to change the drive names in the `PATH` command to search the new locations created under the `JOIN` command.

Invalid drive or file name

You are trying to access a drive or a file that does not exist or that is misspelled. Use `HDSETUP` to verify that the drive is in place and `DIR` to see if the file exists.

Invalid drive specification

Your syntax is incorrect; refer to the command syntax. You may be trying to run CHKDSK or FORMAT on a drive that does not exist. Run HDSETUP to verify that you have cylinders and partition length specified for the drive and that a boot partition has been chosen.

Invalid number of parameters

You did not specify a string when issuing the FIND command. Verify that you have the correct number of quote marks.

Invalid parameter

Check the requirements for your command. You specified an option that does not exist. Or you may have used a slash instead of a back slash. Verify the accuracy of your command.

Invalid parameter: option

File Comparison message: One of the switches you specified is invalid. Compare your command with the syntax.

EDLIN message: You specified a switch other than /B when starting EDLIN. You should use the /B switch if you have end-of-file marks (Ctrl-Z characters) embedded in the file; otherwise EDLIN will stop loading the file at the first Ctrl-Z.

Invalid path, not directory, or directory not empty

You specified an incorrect pathname or one that does not exist. Verify the parameters in your command. Or you tried to remove a directory that had files still resident. If you want to remove a directory, you must first remove the files using DEL.

Invalid subdirectory entry

CHKDSK message: Refer to CHKDSK.

Invalid time

The time of day setting has been incorrectly entered. Be sure to use the correct delimiters, such as colons or periods. If you are using a non-U.S. keyboard, you should use the same delimiters that are shown on the screen after you enter the TIME command.

Keyboard code missing

You have entered the SELECT command without specifying the keyboard code. Refer to SELECT in the *MS-DOS Command Reference*.

Line too long

EDLIN message: This error occurs if a REPLACE command caused a line to expand beyond the limit of 253 characters. Divide the long line into two lines; then try the REPLACE command twice.

Must specify destination number

EDLIN message: You tried to execute a COPY or MOVE command without a destination line number. Reissue the command with a destination line number.

Must specify ON or OFF

This message is displayed if you have entered the BREAK command incorrectly.

No paper error writing device PRN

Your printer is turned off or out of paper.

No room in directory for file

EDLIN message: This error occurred either because the directory was full or because you specified an illegal disk drive or an illegal filename. Verify that your command contains no illegal entries; then run CHKDSK for the specified disk drive. If the status report shows that no bytes are free, you should either delete some files or insert a new disk.

Not enough room to merge the entire file

EDLIN message: Caused because there was not enough room in memory to hold the file during a TRANSFER command. You will have to free some memory by deleting files or moving them elsewhere before transferring the file.

Not ready error reading drive A

Abort, Retry, Ignore?

Insert a floppy disk in drive A and close the drive.

Out of environment space

DOS was unable to accept a command because it could not expand the area in which the environment information is kept. This usually happens when you try to add to the environment after loading a program that becomes resident in memory, such as GRAPHICS, PRINT, GRAFTABL, MODE, SideKick, or ProKey.

Path not found

You may have tried to redirect a screen output to a protected file. Use ATTRIB to release the file from its protected state. Refer also to File creation error.

Probable non-DOS disk

Continue (Y/N)?

You may have inserted a nonsystem disk or a system disk that does not have DOS command files. Press Y if you want the program to continue.

!! Selected drive is not hard disk !!

Format failure

You tried to format a drive partition that was created by VDISK.SYS. The FORMAT command only works on the disk drive partition that is resident on the disk drive. VDISK creates a partition that is resident only in memory.

Speed not changeable on this machine

Your computer does not have the capability to switch speeds.

Syntax error

You typed an illegal string when issuing the command. For example:

```
ATTRIB filename +r
```

is corrected to:

```
ATTRIB +r filename
```

Target diskette unusable

The diskette may need to be formatted. Diskettes used for BACKUP usually require reformatting before being put to other uses, as, for example, the COPY command.

Target diskette write protected

If you want to copy onto a diskette, change files, or use the MORE or SORT (!) filters, you must take off the write-protect tab.

Unable to create directory

This message occurs if you attempt to create a directory that already exists. Enter dir \directory to see if it is already resident on your drive.

Unrecoverable error in directory

Convert directory to file (Y/N)?

CHKDSK message: If you respond to this prompt by pressing Y, CHKDSK will convert the bad directory into a file. You can then fix the directory yourself or delete it.

Unrecoverable format error on target

DISKCOPY message: Use another diskette. Run FORMAT to correct the diskette.

Unrecoverable read error on source

DISKCOPY message: Copy files onto another location in your disk drive or onto another floppy disk.

Unrecoverable verify error on target

DISKCOPY message: Target disk may be corrupt. Repeat operation onto another disk for insurance.

Unrecoverable write error on target

DISKCOPY message: Target disk may be corrupt. Repeat operation onto another disk for insurance.

Unsupported source diskette format

DISKCOPY message: You may have to reformat a diskette before using it, or it may be a type incompatible with your floppy-disk drive. Refer to FORMAT in MS-DOS Command Reference.

WARNING! No files were found to back up

Your BACKUP command specified a drive in which no files exist. Verify the drive that contains the files you want to back up.

WARNING - Read error on EXE file.

Amount read less than size in header

This is a warning message only

EXE2BIN message: Refer to EXE2BIN.

Warning: System switch has NOT been selected!

WARNING, ALL DATA ON NON-REMOVABLE DISK

DRIVE x: WILL BE LOST!

Proceed with Format (Y/N)?

The FORMAT command you have entered will not install the system files on the drive you want to format. If you continue, you will not be able to boot from this drive until you perform a SYS. You can either enter SYSGEN [d:], or type SYS [d:] and then copy your system files onto the drive. All the contents of your current drive will be deleted if you press Y.

WARNING, ALL DATA ON NON-REMOVABLE DISK

DRIVE x: WILL BE LOST!

Proceed with Format (Y/N)?

This is a warning message to verify the drive that you are about to format. Once you press Y, the process of formatting will delete all the files that are on the disk.

Write protect error writing drive A

Abort, Retry, Ignore?

Insert disk without write-protect tab.

X lost clusters found in y chains

Convert lost chains to files (Y/N)?

CHKDSK message: If you respond to this prompt by pressing Y, CHKDSK will create a directory entry and a file for you to resolve this problem (files created by CHKDSK are named FILEnnnn.CHK). CHKDSK will then display:

X bytes disk space freed

If you respond to this prompt by pressing N and have not specified the /F switch, CHKDSK frees the clusters and displays:

X bytes disk space would be freed

GW-BASIC Quick Reference

LOADING GW-BASIC

To load GW-BASIC you must be able to access the files that are provided on the GW-BASIC diskette. The method for accomplishing this depends on whether you have a dual-floppy-drive system or a hard-disk-drive system. The following instructions summarize the steps described in the *User's Guide*.

If You Have a Dual-Floppy-Drive System

It is recommended that you create a working copy of your GW-BASIC diskette just as you did for the MS-DOS diskette. Use this copy to execute the following procedures after you have loaded MS-DOS.

Loading from the GW-BASIC Diskette

Insert the GW-BASIC diskette in floppy drive A. All you have to do to load GW-BASIC is to enter this command:

xbasic

Loading from a Bootable Application Diskette

Insert the GW-BASIC diskette in drive A and a bootable application diskette in drive B (refer to the "MS-DOS" chapter for creating an application diskette). Enter this command:

copy a:*. * b:

Now that the application diskette contains the GW-BASIC files, you can use it to boot from drive A; then enter the command xbasic.

If You Have a Hard-Disk-Drive System

Copy the files from the GW-BASIC diskette onto your drive. After you have booted up in drive C, enter the following command:

copy a:*. * c:

If you have transferred the files to the root directory of drive C, you can load GW-BASIC by entering the command `xbasic`. Be sure to modify the search path in your `AUTOEXEC.BAT` file if you are putting the files in a subdirectory or in a drive other than the root directory of drive C.

Loading a BASIC Program from DOS

You can load and execute BASIC programs from the DOS command level. For example, if you want to load and execute a BASIC program file called `GOFILE.BAS`, which is resident in the subdirectory `\BASIC` in drive D, type:

```
xbasic d:\BASIC\GOFILE Enter
```

To be sure that you leave GW-BASIC after the program has finished executing, you have to make the command `SYSTEM` the last command of your program. This command will cause you to exit from GW-BASIC back to the DOS command line.

ALTERNATE KEYS

A AUTO	G GOTO	N NEXT	U USING
B BSAVE	H HEX\$	O OPEN	V VAL
C COLOR	I INPUT	P PRINT	W WIDTH
D DELETE	K KEY	R RUN	X XOR
E ELSE	L LOCATE	S SCREEN	
F FOR	M MID\$		

CONTROL KEYS

Ctrl + **Break**

Stops program execution.

Ctrl + **Num Lock**

Pauses till any key is pressed.

Ctrl + **-**

Tabs to next word.

Ctrl + **-**

Reverses tab to last word

Ctrl + **Home**

Erases screen and moves cursor to upper-left corner.

Ctrl + **Alt** + **Del**

Resets system (soft boot).

GW-BASIC COMMANDS

AUTO [*start line*] [*increment*]

Generates line numbers automatically.

BEEP

Sounds speaker at 800Hz for 1/4 second.

BLOAD *filespec* [*offset*]

Loads machine language programs.

BSAVE *filespec*, *offset*, *length*

Saves machine language programs.

CALL *numeric variable* [(*variable* [, *variable*]. . .)

Calls a machine language subroutine.

CHAIN [**MERGE**] *filespec* [, [*line number*] [**ALL**] [**DELETE** *line number* - *line number*]

Transfers control to another program.

CHDIR *pathname*

Changes current directory designation to a directory designated pathname.

CIRCLE (*xcoord*, *ycoord*), *radius* [, *color* [, *start*, *end* [, *aspect ratio*]]]

Draws an ellipse on the screen.

CLEAR [, [*byte count*] [, *stack*]

Sets all numeric variables to zero and all string variables to null. Optionally sets the end of memory and the amount of stack space.

CLOSE [[#] *file number* [, [#] *file number*. . .]

Closes a file OPENed for I/O.

CLS

Clears the screen.

COLOR [*foreground*] [, [*background*] [,*border*]]

Used in text mode to set the foreground, background and border colors. Valid colors are:

0 - Black	8 - Gray
1 - Blue	9 - Light Blue
2 - Green	10 - Light Green
3 - Cyan	11 - Light Cyan
4 - Red	12 - Light Red
5 - Magenta	13 - Light Magenta
6 - Brown	14 - Yellow
7 - White	15 - High Intensity White

The colors 8-15 may be used only as foreground colors. Adding 16 to any foreground color causes that color to blink.

COLOR [*background*] [,*palette*]

Used in graphics mode to set the colors used by the graphics statements. Valid colors are:

Palette 0	Palette 1	Value
Green	Cyan	1
Red	Magenta	2
Brown	White	3

COM (1 | 2) **ON** | **OFF** | **STOP**

Enables or disables communications trapping.

COMMON *variable* [,*variable*] ...

Passes variables to CHAINED programs.

CONT

Resumes program execution after execution has been stopped.

DATA *constant* [,*constant*] ...

Stores constants to be accessed by READ statements.

DATE\$

Sets the system date.

DEF FN *name* [*argument* [,*argument*] ...] = *function definition*

Defines a function written by the user.

DEF SEG [= *address*]

Defines the current segment used for storage.

DEF USR [*number*] = *offset*

Specifies the starting address of a machine language subroutine which is to be called by the USR function.

DEFDBL *letter* [-*letter*] [,*letter* [-*letter*]] ...

Defines specified variables as double precision.

DEFINT *letter* [-*letter*] [,*letter* [-*letter*]] ...

Defines specified variables as integer precision.

DEFSNG *letter* [-*letter*] [,*letter* [-*letter*]] ...

Defines specified variables as single precision.

DEFSTR *letter* [-*letter*] [,*letter* [-*letter*]] ...

Defines specified variables as string variables.

DELETE [*line number*] [-*line number*]

Deletes program lines.

DIM *variable* (*elements*) [,*variable* (*elements*)] ...

Initializes and reserves space for arrays.

DRAW *command string*

Used in graphics mode only. Draws an object as specified by the command string.

Valid string characters are:

Un	Up
Dn	Down
Ln	Left
Rn	Right
En	Diagonal up and right
Fn	Diagonal down and right
B	Move do not plot
Gn	Diagonal down and left
Hn	Diagonal up and left
Mx,y	Move absolute or relative
An	Set angle
Cn	Set color
Sn	Set scale
N	Move and return
Xstring	Execute string

EDIT *line number*

Displays a line for editing.

END

Ends program execution.

ERASE *array* [*array*] ...

Erases an array from a program.

ERROR *number*

Duplicates a BASIC error or defines an error code.

FIELD [#] *file number*, *length AS string name* [*length AS string name*] ...

Reserves variable space in a random file buffer.

FILES *filespec*

Displays the names of files on the specified diskette.

FOR *counter=start TO finish* [STEP *increment*]

Performs a series of instructions occurring between the FOR and NEXT statements in an iterative fashion.

GET [#] *file number* [*number bytes*]

Reads a record from a communications file.

GET [#] *file number* [*record number*]

Reads a record from a random file into a random buffer.

GET (*x1*, *y1*) - (*x2*, *y2*), *array name*

Copies points from the screen into an array.

GOSUB *line number*

Branches to a specified line number (subroutine) and returns.

GOTO *line number*

Unconditionally branches to the specified line number.

IF *logical expression* [,] GOTO *line number* [,] ELSE *statements* | *line number*

Branches to specified line number or performs statements based upon the value of the logical expression.

IF *logical expression* [,] THEN *statements* | *line number* [,] ELSE *statements* | *line number*

Conditionally processes statements based on the value of the logical expression.

INPUT [:] ["*prompt string*";] *list of variables*

Permits input from the keyboard during program execution.

INPUT # *file number*, *variable list*

Reads data from a sequential file and assigns them to the specified variables.

IOCTL [#] <*file—number*>,"<*control—string*<"

Sends a control data string to an OPENed character device driver.

KEY ON | OFF | LIST

Controls display of function keys on line 25 of the screen.

KEY (*number*) ON | OFF | STOP

Enables/disables Key trapping.

KEY *number*, "*string*"

Assigns a string to a specified function key.

Kill *filespec*

Deletes a file from the specified diskette.

LCOPY

Copies the data on the screen to the printer.

[LET] *variable* = *expression*

Assigns a value to a variable.

LINE [(*x1*, *y1*)] - (*x2*, *y2*) [,] [*color*] [,B [F]]

Used only in graphics mode. Draws a line or box on the screen.

LINE INPUT [:] ["*prompt string*";] *list of variables*

Permits an entire line to be input from the keyboard.

LINE INPUT # *file number*, *variable list*

Permits an entire line to be read from a sequential file.

LIST [*line number* [- [*line number*]] [,] [*filespec*]

Lists the program currently in memory on the screen or specified device.

LLIST [*line number* [- [*line number*]]]

Lists the program currently in memory on the printer.

LOAD *filespec* [,R]

Loads a program in to memory and optionally runs it.

LOCATE [*row*] [,] [*column*] [,] [*cursor switch*] [,] [*start*] [,*stop*]]]]

Positions the cursor on the screen and optionally alters its shape.

LPRINT [*list of expressions*] [:]

Prints data at the printer.

LPRINT USING *format string; list of expressions* [;]
Prints formatted data at the printer. See PRINT USING.

LSET *field variable*
Moves data to a random file buffer, left justified.

MERGE *filespec*
Merges program lines from an ASCII program file into the program currently in memory.

MID\$ (*string, start pos* [, *number of chars*])
Replaces a portion of one string with another string.

MKDIR *pathname*
Creates a new directory.

NAME *filespec AS filename*
Changes the name of the specified file.

NEW
Erases the program in memory and clears all variables.

NEXT [*counter* [, *counter*] . . .]
Increments counter of FOR statement and ends loop.

ON COM (1 | 2) GOSUB *line number*
Establishes a subroutine to which control will transfer when data comes into the communications buffer.

ON ERROR GOTO *line number*
Establishes a subroutine to which control will transfer when an error occurs.

ON *expression* GOSUB *line number* [, *line number*] . . .
Branches to one of the specified subroutines based on the value of the expression.

ON *expression* GOTO *line number* [, *line number*] . . .
Branches to one of the specified lines based on the value of the expression.

ON KEY (*number*) GOSUB *line number*
Establishes a subroutine to which control is passed when the specified key is pressed.

ON PEN GOSUB *line number*
Establishes a subroutine to which control is passed when the light pen is activated.

ON PLAY (*n*) GOSUB *line number*
Plays music defined by a PLAY statement during execution of other program lines.

ON STRIG (0 | 2 | 4 | 6) GOSUB *line number*
Establishes a subroutine to which control is passed when the specified joystick button is pressed.

ON TIMER (*n*) GOSUB *line number*
Executes a subroutine every (*n*) seconds.

**OPEN *filespec* [FOR INPUT | OUTPUT | APPEND] AS [#] *file number*
[LEN = *record length*]**
Permits I/O to a specified file or device.

OPEN O | I | R, [#] *file number, filespec* [, *record length*]
Permits I/O to a specified file or device.

**OPEN "COM *number*: [*bit rate*] [, *parity*] [, *data*] [, *stop*]" AS [#] *file number*
[LEN = *record length*]**
Opens a communications file.

OPTION BASE *base*
Specifies the minimum value for array subscripts.

OUT *port number, data*
Sends a byte of data to a machine output port.

PAINT (*xcoord, ycoord*) [, *color* [, *boundary color*]]
Used in graphics mode only. Colors a bounded area around a specified point.

PEN ON | OFF | STOP
Enables/disables reading of the light pen.

PLAY *command string*
Plays music as specified by the command string. Valid string characters are:

L n	Length = n
MF	Music Foreground
MB	Music Background
MN	Music normal
ML	Music Legato
N n	Play note 0-84
O n	Set Octave 0-6
P n	Pause n 0-64
T n	Tempo 32-255
Xstring	Play string
.	Play as dotted note
A - G	Musical notes

+, # Play as a sharp
- Play as a flat

POKE *offset, data*

Writes a byte of data into a specified memory location.

PRINT [*list of expressions*] [:]

Displays data on the screen.

PRINT USING *format string; list of expressions* [:]

Displays formatted data on the screen. Valid formatting characters are:

Numeric Fields

Indicates digit position
 . Prints decimal point
 , Prints commas at thousands, and so on
 + Prints sign of number (before or after number)
 - Prints trailing minus sign
 ** Fills leading spaces with asterisks
 \$\$ Prints dollar sign next to number
 **\$ Fills leading spaces with asterisks and print dollar sign next to number
 Prints in exponential format

String Fields

! Prints first character
 \n spaces\ Prints first 2 + n characters
 & Prints entire string

PRINT # *file number, list of expressions*

Writes data to a sequential file.

PRINT # *file number, USING format string; list of expressions*

Writes formatted data to a sequential file. See PRINT USING.

PRESET (*xcoord, ycoord*) [*color*]

Used in graphics mode only. Draws a point at a specified position on the screen. Default color is the background color.

PSET (*xcoord, ycoord*) [*color*]

Used in graphics mode only. Draws a point at a specified position on the screen. Default color is the foreground color.

PUT [#] *file number, record number*

Writes a record from a random file buffer to a random file.

PUT (*xcoord, ycoord*) *numeric array* [, *logical operation*]

Used in graphics mode only. Places colors at a specified area of the screen.

RANDOMIZE [*seed*]

Reseeds random number generator.

READ *variable* [, *variable*] . . .

Assigns values from DATA statements to the specified variables.

REM *string*

Permits remarks to be inserted in a program.

RENUM [*new line number*] [, [*start line*] [, *increment*]

Renumbers the program line numbers.

RESET

Closes all diskette files and clears the system buffer.

RESTORE [*line number*]

Resets the DATA statement pointer to the specified line.

RESUME [0] | **NEXT** | *line number*

Resumes program execution following execution of a subroutine.

RETURN [*line number*]

Returns program control from a subroutine to the line immediately following the most recently executed GOSUB statement.

RMDIR *pathname*

Removes a directory name.

RSET *field variable*

Moves data to a random file buffer, right justified.

RUN [*line number*]

Begins program execution at specified line.

RUN *filespec* [,R]

Begins execution of a specified program.

SAVE *filespec* [,A] [,P]

Stores a BASIC program on diskette.

SCREEN [*mode*] [, [*color switch*] [, [*active page*] [, *visual page*]]]

Sets the screen attributes.

SOUND *frequency, duration*

Generates sound at a specified frequency for a specified duration.

STOP

Terminates program execution.

STRIG ON | OFF

Enables/disables STRIG function.

STRIG (0 | 2 | 4 | 6) ON | OFF | STOP

Enables/disables trapping with the ON STRIG statement.

SWAP *variable 1, variable 2*

Exchanges the values of two variables.

SYSTEM

Returns to DOS from BASIC.

TIMES\$

Sets current system time in hh:mm:ss format.

TROFF

Turns tracing off.

TRON

Turns tracing on.

VIEW [SCREEN] [(x1,y1)-(x2,y2)][,fill—color] [,border—line]]]

Defines an area on the monitor screen, bounded by the specified physical screen coordinates.

WAIT *port number, and expression* [,xor expression]

Suspends program execution until a specified bit pattern is matched by the contents of the specified port.

WEND

Marks the end of loop statements associated with a WHILE statement.

WHILE *logical expression*

Establishes a conditional loop which executes until the logical expression evaluates as false. See also WEND.

WIDTH [*file number, device*] *characters*

Sets width of output line.

WINDOW [[SCREEN] (x1,y1)-(x2,y2)]

Defines a coordinate system that is independent of the physical coordinates of the monitor screen, within which images may be defined and displayed.

WRITE [*list of expressions*]

Displays data on the screen.

WRITE # *file number, list of expression*

Writes data to a sequential file.

FUNCTIONS AND VARIABLES**ABS (*argument*)**

Returns the absolute value of the argument.

ASC ("*string*")

Returns the ASCII code for the first character of the string.

ATN (*argument*)

Returns the arctangent of the argument.

CDBL (*argument*)

Converts the argument to a double-precision number.

CHR\$ (*code*)

Converts an ASCII code to its character equivalent.

CINT (*argument*)

Converts the argument to an integer by rounding the fraction.

COS (*argument*)

Returns the cosine of the argument.

CSNG (*argument*)

Converts the argument to a single-precision number.

CSRLIN

Returns the vertical coordinate of the cursor.

CVD (*8 byte string*)

Converts an eight-byte string to a double-precision number.

CVI (*2 byte string*)

Converts a two-byte string to an integer.

CVS (*4 byte string*)

Converts a four-byte string to a single-precision number.

DATE\$

Returns the system date string.

EOF (file number)

Returns -1 if the end of the file has been reached for the specified file. Otherwise it returns 0.

ERDEV

Holds the actual value of the last device error.

ERDEV\$

Holds the name of the device causing the error.

ERL

Returns the line number associated with the most recent error.

ERR

Returns the error code of the most recent error.

EXP (argument)

Returns the value of e to the power of the argument (that is, e^x).

FIX (argument)

Converts the argument to an integer by truncating the fraction.

FRE (dummy | "string")

Returns the number of free bytes in memory.

HEX\$ (decimal argument)

Returns a string which is the hexadecimal equivalent of CINT (decimal argument).

INKEY\$

Returns a character string based upon the character in the keyboard buffer.

INP (port number)

Returns the byte read from a specified port.

INPUT\$ (number of characters [, [#] file number)

Returns a string of specified length, read from a specified file.

INSTR ([offset,] searched string, target string)

Returns the position of the first occurrence of the target string in the searched string.

INT (argument)

Returns the largest integer less than or equal to the argument.

IOCTL\$

Returns a control data string from an OPENed device driver.

LEFT\$ ("string", number of characters)

Returns the leftmost n characters from the specified string.

LEN ("string")

Returns the number of characters in the specified string.

LOC (file number)

Returns the current position in the file.

LOF (file number)

Returns the length of the file (in bytes).

LOG (argument)

Returns the natural logarithm of the argument (i.e. $\ln(x)$).

LPOS (printer number)

Returns the position of the print head within the buffer.

MID\$ ("string", start pos [, number of characters])

Returns specified number of characters from within a string.

MKD\$ (double precision expression)

Converts the value of a double-precision expression to an eight-byte string.

MKI\$ (integer expression)

Converts an integer expression to a two-byte string.

MKS\$ (single precision expression)

Converts the value of a single-precision expression to a four-byte string.

OCT\$ (decimal argument)

Returns a string which is the octal equivalent of the argument.

PEEK (address)

Returns the byte read from the specified address.

PEN (number)

Returns the coordinates of the light pen.

PLAY (n)

Returns the number of notes in the music background buffer.

PMAP

Translates coordinate values between WINDOW and SCREEN coordinate systems.

POINT (xcoord, ycoord)

Returns the number of the color at the specified coordinates.

POS (dummy)

Returns the current cursor column position.

RIGHT\$ ("string", number of characters)

Returns the rightmost n characters from a string.

RND [(seed)]

Returns a random number between zero and one.

SCREEN (row, column [, logical expression])

Returns the ASCII code for the character at the specified position.

SGN (argument)

Returns the value of 1 if the argument is positive, -1 if negative, and zero if it is 0.

SHELL

Exits temporarily from a program to the operating system. Returns to the program after entering EXIT.

SIN (argument)

Returns the sine of the argument.

SPACE\$ (number)

Returns a string consisting of the specified number of spaces.

SPC (number)

Used with PRINT, LPRINT, and PRINT# to print the specified number of spaces.

SQR (positive argument)

Returns the square root of the argument.

STICK (0 | 1 | 2 | 3)

Returns the coordinates of the joysticks.

STR\$ (expression)

Returns a string representation of the value of the expression.

STRIG (number)

Returns the status of the joystick buttons.

STRING\$ (length, code | "string")

Returns a string of specified length whose characters are all the first character of the specified string or all have the specified ASCII code.

TAB (position)

Used with PRINT, LPRINT, and PRINT# to tab to the specified position.

TAN (argument)

Returns the tangent of the argument.

TIME\$

Returns the current system time.

TIMER

Returns the number of seconds elapsed since midnight, or since a system reset.

USR [number] (argument)

Returns the result of a machine language subroutine which operates on the specified argument.

VAL ("string")

Returns the numerical value of the string argument.

VARPTR (variable | #file number)

Returns the address of a variable or file control block.

VARPTR\$ (variable)

Returns a character representation of the address of a variable in memory.

LOGICAL OPERATORS

Logical operators perform tests on multiple relations, bit manipulation, or Boolean operations.

NOT	A	NOT A	OR	A	B	A OR B
	T	F		T	T	T
	F	T		F	T	T
	T	F		T	F	T
	F	T		F	F	F

AND	A	B	A AND B	XOR	A	B	A XOR B
	T	T	T		T	T	F
	F	T	F		F	T	T
	T	F	F		T	F	T
	F	F	F		F	F	F

IMP	A	B	A IMP B	EQV	A	B	A EQV B
	T	T	T		T	T	T
	F	T	F		F	T	F
	T	F	T		T	F	F
	F	F	T		F	F	T

GW-BASIC Error Codes and Messages

GW-BASIC ERROR CODES AND ERROR MESSAGES

This section contains detailed descriptions of all GW-BASIC error codes and error messages. The first listing gives the error messages in numerical order by error code. The error number refers to the number that is returned by an ERR trap. The second listing is in alphabetical order by error message with a description of the error.

Numerical Listing of Error Messages

Message	Error Code Number	
	1	NEXT without FOR
	2	Syntax error
	3	Return without GOSUB
	4	Out of data
	5	Illegal function call
	6	Overflow
	7	Out of memory
	8	Undefined line number
	9	Subscript out of range
	10	Duplicate definition
	11	Division by zero
	12	Illegal direct
	13	Type mismatch
	14	Out of string space
	15	String too long
	16	String formula too complex
	17	Can't continue
	18	Undefined user function
	19	No RESUME
	20	RESUME without error
	21	Unprintable error
	22	Missing operand
	23	Line buffer overflow
	24	Device timeout
	25	Device fault
	26	FOR without NEXT

GW-BASIC

27	Out of paper
29	WHILE without WEND
30	WEND without WHILE
50	Field overflow
51	Internal error
52	Bad file number
53	File not found
54	Bad file mode
55	File already open
57	Device I/O error
58	File already exists
61	Disk full
62	Input past end
63	Bad record number
64	Bad file name
66	Direct statement in file
67	Too many files
68	Device unavailable
69	Communication buffer overflow
70	Disk write protected
71	Disk not ready
72	Disk media error
74	Rename across disks
75	Path/file access error
76	Path not found

Alphabetical Listing of Error Messages**Bad file mode**

An attempt is made to use PUT, GET, or LOF with a sequential file, to LOAD a random file, or to execute an OPEN statement with a file mode other than I, O, or R.

Bad file name

An illegal form is used for the filename with a LOAD, SAVE, KILL, or OPEN statement (for example, a filename with too many characters).

Bad file number

A statement or command references a file with a file number that is not OPEN or is out of the range of file numbers specified at initialization.

Bad record number

In a PUT or GET statement, the record number is either greater than the maximum allowed (32,767) or equal to zero.

Can't continue

An attempt is made to continue a program that

- Has halted due to an error
- Has been modified during a break in execution
- Does not exist

Communication buffer overflow

A communication input statement was executed but the input buffer was already full. You should use an ON ERROR statement to retry the input when this condition occurs. Subsequent inputs will attempt to clear this fault unless characters continue to be received faster than the program can process them. If this is the case you might do one of several things, such as:

- Implement a "hand-shaking" protocol with the other computer to tell it to stop sending long enough so you can catch up
- Use a lower baud rate to transmit and receive

Device Fault

Indicates a hardware error indication returned by an interface adapter.

Device Timeout

BASIC did not receive information from an input/output device within a predetermined amount of time.

Device Unavailable

You tried to OPEN a file to a device which doesn't exist. Either you do not have the hardware to support the device (such as printer adapters for a second or third printer) or you have disabled the device.

Direct statement in file

A direct statement is encountered while LOADING an ASCII format file. The LOAD is terminated.

Disk full

All disk storage space is in use.

Disk I/O error

An I/O error occurred on a disk I/O operation. It is a fatal error; that is, the operating system cannot recover from the error.

Disk media error

The controller attachment card has detected a hardware or media fault. Usually this means the diskette has gone bad. Copy any existing files to a new diskette and re-format the bad diskette. If formatting fails, the diskette should be discarded.

Disk not Ready

The diskette drive door is open or a diskette is not in the drive. Place the diskette in the drive and continue the program.

Disk write protected

You tried to write onto a diskette that is write-protected.

Division by zero

A division by zero is encountered in an expression or the operation of involution results in zero being raised to a negative power. Machine infinity with the sign of the numerator is supplied as the result of the division, or positive machine infinity is supplied as the result of the involution, and execution continues.

Duplicate Definition

Two DIM statements are given for the same array; or a DIM statement is given for an array after the default dimension of 10 has been established for that array; or an OPTION BASE statement has been encountered after an array has been dimensioned, either by a DIM statement or by default.

Field overflow

A FIELD statement is attempting to allocate more bytes than were specified for the record length of a random file.

File already exists

The filename specified in a NAME statement is identical to a filename already in use on the disk.

File already open

A sequential output mode OPEN statement is issued for a file that is already open; or a KILL statement is given for a file that is open.

File not found

A LOAD, KILL, or OPEN statement references a file that does not exist on the current disk.

FOR without NEXT

A FOR statement was encountered without a matching NEXT statement.

Illegal direct

A statement that is illegal in direct mode is entered as a direct mode command.

Illegal function call

A parameter that is out of range is passed to a math or string function. An FC error may also occur as the result of:

- A negative or unreasonably large subscript.
- A negative or zero argument with LOG.
- A negative argument to SQR.
- A negative mantissa with a noninteger exponent.
- A call to a USR function for which the starting address has not yet been given.
- An improper argument to MID\$, LEFT\$, RIGHT\$, INP, OUT, WAIT, PEEK, POKE, TAB, SPC, STRING\$, SPACE\$, INSTR, or ON...GOTO.

Input past end

An INPUT statement is executed after all the data in the file has been INPUT, or for a null (empty) file. To avoid this error, use the EOF function to detect the end-of-file.

Internal error

An internal malfunction has occurred in BASIC. Report to your computer dealer the conditions under which the message appeared.

Line buffer overflow

An attempt has been made to input a line that has too many characters.

Missing operand

An expression contains an operator with no operand following it.

NEXT without FOR

A variable in a NEXT statement does not correspond to any previously executed, unmatched FOR statement variable.

No RESUME

An error-handling routine is entered but contains no RESUME statement.

Out of data

A READ statement is executed when no DATA statements with unread data remain in the program.

Out of memory

A program is too large, or it has too many FOR loops or GOSUBs, too many variables, or expressions that are too complicated.

Out of Paper

The printer is out of paper, or the printer is not turned on. Insert paper (if necessary); verify that the printer is properly connected and that the power is on.

Out of string space

String variables have caused BASIC to exceed the amount of free memory remaining. BASIC will allocate string space dynamically, until it runs out of memory.

Overflow

The result of a calculation is too large to be represented in BASIC number format. If underflow occurs, the result is zero, and execution continues without an error.

Rename across disks

You tried to rename a file on one disk with a file specification for another disk.

RESUME without error

A RESUME statement is encountered before an error-handling routine is entered.

RETURN without GOSUB

A RETURN statement is encountered for which there is no previous, unmatched GOSUB statement.

String formula too complex

A string expression is too long or too complex. The expression should be broken into smaller expressions.

String too long

An attempt is made to create a string more than 255 characters long.

Subscript out of range

An array element is referenced either with a subscript that is outside the dimensions of the array or with the wrong number of subscripts.

Syntax error

A line is encountered that contains some incorrect sequence of characters (such as unmatched parenthesis, misspelled command or statement, incorrect punctuation, or the like). BASIC automatically enters edit mode at the line that caused the error.

Too many files

An attempt is made to create a new file (using SAVE or OPEN) when all 255 directory entries are full.

Type mismatch

A string variable name is assigned a numeric value or vice versa; a function that expects a numeric argument is given a string argument or vice versa.

Undefined line number

A nonexistent line is referenced in a GOTO, GOSUB, IF...THEN...ELSE, or DELETE statement.

Undefined user function

AUSR function is called before the function definition (DEFUSR statement) is given.

Unprintable error

An error message is not available for the error condition which exists.

WEND without WHILE

A WEND statement was encountered without a matching WHILE statement.

WHILE without WEND

A WHILE statement does not have a matching WEND statement.

System Configuration Record

Fill in this form to have a complete record of the components in your system. Be sure to record jumper and switch settings that have been made for hardware components.

HARDWARE

Model _____ ROM BIOS _____

Base Memory Size _____ Expansion Memory Size _____

Disk Drive 1 _____

Disk Drive 2 _____

Disk Drive 3 _____

Disk Drive 4 _____

Tape Drive _____

Option Board 1 _____

Option Board 2 _____

Option Board 3 _____

Option Board 4 _____

Display Monitor 1 _____

Display Monitor 2 _____

Printer 1 _____ Serial [] Parallel []

Printer Interface _____

Printer 2 _____ Serial [] Parallel []

Printer Interface _____

Other Components _____

SOFTWARE

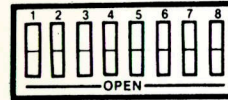
Operating System _____

Drivers Installed in CONFIG.SYS File _____

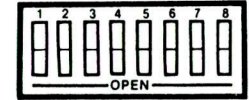
Notes _____

SWITCH SETTINGS

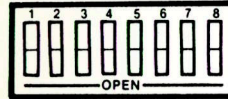
Record the switch settings for your system by putting an x or filling in the ON (Closed) or OFF (Open) positions of board switch settings. Note that some are 8-position switches and others are 6-position.



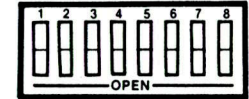
CLOSED
OPEN



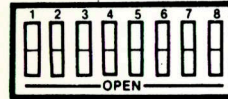
CLOSED
OPEN



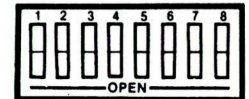
CLOSED
OPEN



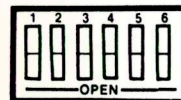
CLOSED
OPEN



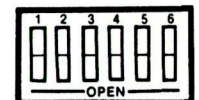
CLOSED
OPEN



CLOSED
OPEN



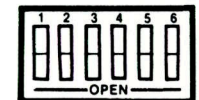
CLOSED
OPEN



CLOSED
OPEN



CLOSED
OPEN



CLOSED
OPEN

Notes

Notes

Notes



