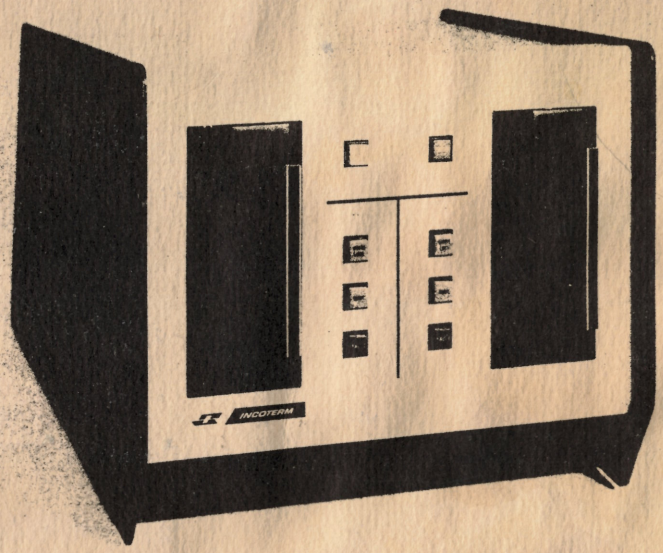
V6 F&S SYSTEMS

INCOTERM®

# SPD/DOS
# DISKETTE OPERATING SYSTEM
# OPERATORS REFERENCE MANUAL




# SPD/DOS

# DISKETTE OPERATING SYSTEM

# OPERATORS REFERENCE MANUAL

ORDER NUMBER: MS-7177.1

DATE: August, 1975

# PREFACE

This manual is intended as an Operating Guide for the SPD/DOS Diskette Operating System running on an INCOTERM SPD 10/20, 10/24, 10/25, or 20/20. It should be noted that the SPD 10/24 was especially designed for overseas customers. The SPD 10/24 is built and sold solely by a licensee, TRANSAC, and is not available within the United States. Programming information for the SPD/DOS System may be obtained from SPD/DOS Diskette Operating System Programmers Reference Manual, Order Number MS-7178.0.

The information in this manual is presented for informational purposes and is not intended or licensed to be used for the construction of equipment. The information is believed to be accurate, but no responsibility is assumed for inaccuracies or for consequences of using the information.

Further, INCOTERM Corporation makes no representation that use of the information in this manual will not infringe on existing or future patent rights of INCOTERM or of others.

# TABLE OF CONTENTS

TABLE OF CONTENTS (cont'd)

TABLE OF CONTENTS (cont'd)

## LIST OF ILLUSTRATIONS



## LIST OF TABLES

# SECTION I

# INTRODUCTION

## GENERAL DESCRIPTION

SPD/DOS is a diskette operating system for use on SPD 10/20, 10/24, 10/25 and SPD 20/20 configurations equipped with an SPD D-250 diskette unit. It provides facilities for program development, maintenance, and storage and implements a file system and attendant utilities for use by diskette applications programs.

Track Ø of every SPD/DOS diskette contains a bootstrap loadable program called the nucleus. The nucleus provides a set of basic commands for maintenance of a file system, the remainder of the diskette being used for file directory and data file area. The files are of four types:

Source Files (S) A source file is a sequential file containing characters from the standard graphic set and is normally used for assembly language source programs.

Object Files (O) These are executable binary programs, typically produced as output from the SPD/DOS assembler.

Relocatable Files (R) These are libraries of relocatable modules produced by the SPD/DOS relocatable assembler. A given file can contain one or more modules.

Data Files (D) A data file is a contiguous set of tracks. DOS imposes no particular structure on a data file. The structure of a data file is determined by the applications program which makes use of the file. Within DOS itself, data files are used as direct access work files.

In addition to the nucleus, SPD/DOS comprises a set of general purpose utility programs which can be loaded from the nucleus like any other binary object program. This operators manual describes the set of nucleus commands and also the standard DOS utilities. It does not discuss the details of preparation of source programs. These details are contained in a separate manual (SPD/DOS Programmers Reference Manual, Order Number (MS-7178.0).

## HARDWARE REQUIREMENTS

### SPD 10/20 Required Hardware

The following hardware is required for operation of SPD/DOS on an SPD 10/20.

Single or dual diskette attached to channel 7.

Upper case only or upper/lower case keyboard attached to channel 2.

15 or 30 line RMTU with 74 or 04 end refresh code attached to channel 8.

SPD 10/24 Required Hardware

The following hardware is required for operation of SPD/DOS on an SPD 10/24.

Single or dual diskette attached to channel 7.

Upper case or upper/lower case keyboard attached to channel 2.

Standard 10/24 RMTU.

SPD 10/25 Required Hardware

The following hardware is required for operation of SPD/DOS on an SPD 10/25.

Single or dual diskette attached to channel 7.

Upper case only or upper/lower case keyboard attached to channel 2.

Single or dual display (display zero is the one used).

SPD 20/20 Required Hardware

The following hardware is required for operation of SPD/DOS on

an SPD 20/20.

Single or dual diskette attached to channel 7.

64 or 80 characters per line display on channel 9, screen 0.

Upper case or upper/lower case keyboard attached to channel 10, keyboard 0.

Optional Hardware

The following peripheral devices may be used by DOS if they are available.

LP125, LP200, LP250, LP300 or LP400 line printer on any channel using the parallel controller.

P-100, P-120C, P-165/P-165A, or P-165B character printer on any channel using the parallel controller.

P-100-2, P-120C-2, P-165/P-165A-2, or P-165B-2 character printer attached to any unit of a SPD 20/20 multiple printer controller on channel 13.

P-15B printer attached to an asynchronous controller on any channel or to any unit of an SPD 20/20 multiple printer controller.

Termiprinter (30 cps, 60 cps or 120 cps) attached to an asynchronous controller on any channel.

Card reader attached to any channel.

Paper tape reader attached to any channel.

Card reader punch attached to any channel.

Magnetic tape transport attached to any channel.

Cassette tape transport attached to any channel.

Built-in cassette loader on channel 11 of an SPD 20/20.

# SECTION II

# OPERATION OF NUCLEUS

## MANUAL LOAD PROCEDURE

The nucleus is loaded from the diskette by following the diskette standard boot procedure. Load any SPD/DOS diskette into unit Ø of a single or dual diskette and depress the boot button. The nucleus will be loaded automatically.

Note: If a diskette fails to boot, its bootstrap record may have been destroyed. If it will boot only with WRITE PROTECT pressed then Track 2 has become deformatted. In either case, reformatting with the N option (see FORMAT utility) may be used to "rescue" such a diskette.

The format of the nucleus display is as shown by the example in Figure 2-1. From this it is seen that the file directory is displayed on the screen using one line of display for each active file.

## DIRECTORY DISPLAY FORMAT

Every SPD/DOS diskette is assigned a Disk Serial Number (DSN) of up to eight characters in length. The nucleus displays this DSN automatically. Following a boot procedure, it is always unit Ø for which the directory is displayed. This may be modified by an appropriate command which will be described later. The unit whose directory is displayed is referred to as the currently selected unit.

The meaning of the various fields in the directory display is as follows:

S File Status. This character is blank for a normal active file. It is set to an asterisk (*) for a file which has been erased. When a file is erased, the space is not immediately reclaimed, rather the asterisk in the file entry indicates the erased status. A third possibility for this field is "?" (for error). This status occurs as a result of certain operations in which errors were detected. A file marked with an ? status may be used as though it had a normal blank status, although there is a high probability that the information in the file is not entirely recorded correctly.

T Type. The type field contains one of the following four letters. O for object file, D for data file, R for relocatable file, or S for source file.

SPD/DOS SPD DISKETTE OPERATING SYSTEM VERSION 5.05-J

UNIT 0 DSN=DISK059

| ST | ..NAME.. | FT | IF | NT | .....LABEL..... | | |
|---|---|---|---|---|---|---|---|
| 0 | CNFG | Ø3 | Ø5 | Ø1 | SPD/DOS CNFG | V5.Ø3 | 75-Ø3-Ø9-Ø11Ø |
| 0 | CREATE | Ø8 | Ø5 | Ø1 | SPD/DOS CREATE | V5.Ø1 | 75-Ø3-Ø2-Ø13Ø |
| 0 | DCOPY | Ø9 | Ø5 | Ø1 | SPD/DOS DCOPY | V6.Ø1 | 75-Ø6-Ø1-17ØØ |
| 0 | EDIT | 1Ø | Ø5 | Ø2 | SPD/DOS EDIT | V5.Ø6 | 75-Ø3-11-Ø74Ø-H |
| 0 | ERASE | 12 | Ø5 | Ø1 | SPD/DOS ERASE | V6.Ø1 | 75-Ø6-Ø1-2ØØØ |
| 0 | FORMAT | 13 | Ø5 | Ø1 | SPD/DOS FORMAT | V5.Ø5 | 75-Ø3-Ø8-214Ø-J |
| 0 | LIST | 14 | Ø5 | Ø1 | SPD/DOS LIST | V6.Ø2 | 75-Ø6-Ø1-1ØØØ |
| 0 | PACK | 15 | Ø5 | Ø1 | SPD/DOS PACK | V6.Ø1 | 75-Ø6-Ø1-12ØØ |
| 0 | RENAME | 16 | Ø5 | Ø1 | SPD/DOS RENAME | V6.Ø1 | 75-Ø6-Ø1-Ø13Ø |
| 0 | XDISK | 17 | Ø5 | Ø1 | SPD/DOS XDISK | V5.Ø1 | 75-Ø3-Ø2-Ø22Ø |
| 0 | ZAP | 18 | Ø5 | Ø2 | SPD/DOS ZAP | V5.Ø3 | 75-Ø3-Ø8-1Ø3Ø-F |
| 0 | ASSEMBLE | 20 | Ø5 | Ø6 | SPD/DOS ASSEMBLE | V6.Ø6 | 75-Ø6-15-1Ø15-C |
| D | WORK | 26 | ØØ | 3Ø | ***ASSEMBLE's WORK FILE*** | | |
| R | DOSLIB | 56 | Ø9 | Ø2 | SPD/DOS LIBRARY | V6.Ø4 | 75-Ø6-15-2Ø4Ø |

Figure 2-1. Example of NUCLEUS Display

NAME This field contains the file name which is up to eight alpha-numeric characters in length starting with an alphabetic character.

FT First Track. This field gives, in decimal, the number of the initial track of the file, ranging from three to sixty-three.

IF Interlace Factor. The interlace factor is the physical separation between logically sequential sectors. It is possible to read more than one sector of data on a single revolution of the disk providing that the sectors are sufficiently spaced to allow processing of the data from one sector before the following sector is read. Thus if sequential data is written to sectors separated by a sufficiently large factor, the processing speed is improved as a result of being able to read more than one sector in each revolution. Within SPD/DOS, all object files have an interlace factor of 5. Relocatable files have an interlace factor of 9. Source files created by the standard source file utility routines have an interlace factor of 11. In general, the interlace factor may be any odd value from 1 through 31. For random access files where data is not processed in a sequential manner, a value of Ø is used.

NT Number of Tracks. This field shows, in decimal, the number of consecutive tracks assigned to the file. It is a value between 01 and 61.

LABEL The label on a file is up to 40 characters of information from the graphic character set. The label is used for identification purposes only, and the choice of the label is entirely up to the user.

In the case of a diskette with a large number of files, the directory may overflow a single screen display. In this case, pressing the space bar causes successive sections of the directory to be displayed. This occurs in a circular manner so that once the entire directory has been displayed, pressing the space bar results in re-display of the initial section.

## UNIQUENESS OF FILE NAMES

Two files on the same diskette of the same type may not have the same name. However, it is permissible to have files of different types with the same name. For example, O (object) files have the same name as their corresponding S (source) files.

A further restriction applies to object file names in that the first

two characters of names should be kept unique if the two letter abbreviation method is to be used to load the programs.

## KEYBOARD ENTRY OF NUCLEUS COMMANDS

When the nucleus is booted in, the cursor is visible and the keyboard unlocked for command entry. The keyboard layout for entry of commands is shown in Figure 2-2. It will be noticed that two special function keys are employed. The left cursor key is used to provide a backspace and erase function for error correction. The LINE RETURN key is used at the end of a command to terminate the entry.

## FILE NAME FORMAT

In various places the commands employ references to file names. A uniform format is used for such references as follows:

U.NAME — U is a single digit either 0 or 1 representing the unit on which the file resides. NAME is the file name, up to eight alpha-numeric characters. The unit number and its following period may be omitted in which case the currently selected unit is assumed.

## NUCLEUS COMMANDS

Nucleus commands are those which are processed by the nucleus

itself. They always start with a period as the first character. This period acts to distinguish nucleus commands from program load requests. The following sections list the format and use of various nucleus commands in alphabetical order.

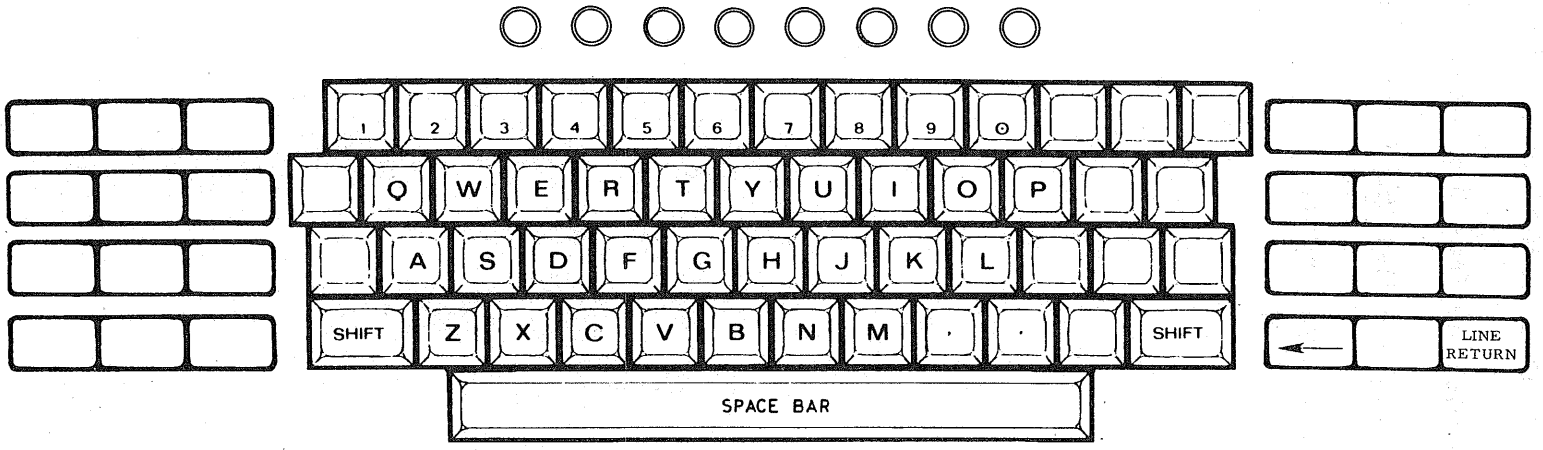



Figure 2-2. Keyboard Layout for Command Entry, All Keyboards

## .C Set Card Input Mode

This command is entered as follows:

.C

It causes subsequent commands to be read from the system card reader. Commands are punched left justified on successive cards, one command to a card. The format of commands entered from cards is exactly the same as that used for keyboard entry except that no backspace or line return functions are needed or available. Input of commands from cards continues until a .E (end of file) record is read. At this point, command input automatically switches back to the keyboard. In order to read commands from cards using the .C command, a system card reader channel must have been configured. See CNFG utility for details.

## .E End of File

The .E command, which is entered as:

.E

is used to return input to the keyboard. It is thus only meaningful when entered from card reader, tape reader, or a command file. The last tape or card record of a command file should always be .E to return to the keyboard for command entry.

.F Set File Input Mode

The format of the.F command is as follows:

.F FILENAME

FILENAME is the name of a source file on one of the currently loaded diskettes. Subsequent to the entry of the .F command, commands will be taken from successive records of the specified file. The format of commands on the file is one command to a record using the same character set as would be used from the keyboard except that no backspace or line return functions are required. Commands are automatically read from the file until the end of file is detected, at which time command input automatically returns to the keyboard.

Note that once a .F command has been entered, the unit containing the command file may not be dismounted until the entire file has been read. The .F command may occur in the file being read but the effect is simply to transfer control to another file. There is no nesting of .F commands. Thus the only useful application of a .F command within a diskette command file is as the last command in such a file. One particular application of this technique is to name the file itself in that last .F command. This causes an indefinite repeated execution of the commands in the file.

.L Set Log Mode

Following a manual boot, commands appear on the screen as they are entered but are not printed. Use of this command in the form;

.L

causes log mode to be entered. In log mode all commands and error messages are logged to the system printer. Thus in order to use the log command, a printer must have been configured.

.M Operator Message

When commands are read from a batch source (cards, paper tape, or diskette file) it may be necessary to pause to allow operator action such as loading or unloading of media from peripheral devices. The message command may be used to effect a delay for this purpose as follows:

.M TEXT

This command will cause TEXT to be displayed on the operator's screen. TEXT may consist of any message composed from the normal graphic character set. All processing is suspended until the space bar is pressed at which time processing resumes by reading the next command. Note that it is possible to enter the .M command in keyboard mode, but its only meaningful use occurs in the context of batch entry of commands.

.N Set No Log Mode

The .N command entered as;

.N

cancels the log mode as set by the .L command. Subsequent commands and error messages are not printed regardless of whether a printer is configured.

.T Set Tape Input Mode

The .T command is used as follows;

.T

to initiate reading of commands from paper tape. A paper tape channel must be configured via CNFG for this command to be legal. The format of command input from paper tape is standard 7 channel ASCII coding. The parity bit is ignored and may be even or odd or missing. Each record is terminated by a carriage return (CR) character. Following the final record there should be an end of tape (EOT) character. All other control characters such as line feed, null, and rubout are ignored. Thus the actual end of line sequence could be carriage return, line feed, or line feed, carriage return, or any other convenient sequence. The character underline or left arrow (ASCII X'5F') has a special function if it occurs immediately prior to the carriage return character. In this case

the entire line is omitted from the input and ignored. This effect occurs only if the underline character occurs immediately prior to the CR character. In any other position it is treated as a normal data character. Once the .T command has been entered, commands are read from paper tape until either a .E command is read or the EOT punch is detected, at which time command input returns to the keyboard mode.

.0 Select Unit 0

This command, entered in the form:

.0

causes unit zero to be set as the currently selected unit. The directory display will be for unit zero, and unit zero will be the default unit for file specifications omitting an explicit unit number.

.1 Select Unit 1

This command, in the form:

.1

sets unit one as the currently selected unit. Directory display and the default unit for file specification references unit one.

## PROGRAM LOADING

Binary object programs are loaded by using a command in the form:

FILENAME, OPTIONS PARAMETERS

FILENAME is the name of an object file on a currently loaded diskette. The name may be abbreviated to its first two characters if desired.

OPTIONS are a series of letters. The order is unimportant but the loaded program can determine which of the 26 possible options were set. The use of option letters will be specified separately for each application program or DOS utility using this feature. If options are not used or not needed in a particular case, they may be omitted together with the preceding comma.

PARAMETERS is a string of characters separated from the option field by a single blank. The meaning of the parameters field depends on the particular application or utility program in use. This field may be omitted if parameters are not required. The maximum command length including the parameter field is 80 characters.

Following execution of a program load command, the specified program is loaded and executed. Operating system utilities automatically return to the nucleus on completion. Application programs may or may not return. Following execution of a program which

does not return automatically, control may be returned to the nucleus by manually rebooting the system.

## ERROR MESSAGES

If an error condition is detected during operation of the nucleus or one of the operating system utilities, the alarm is sounded (if available) and the following message is displayed (and printed if log mode is set):

****E R R O R****  CODE=XXNN

XX is the first two characters of the utility program name or NU for a nucleus error.

NN is a numeric code from 01 - 99.

The error condition is cleared by pressing the letter C key (for continue), whereupon command input resumes from the current command source. Alternately, pressing the space bar will force a return to keyboard mode.

Appendix A contains a complete list of error codes and their interpretation.

## CORE IMAGE SAVE ON MANUAL BOOT

When the nucleus is booted in manually by pressing the boot button or when a program makes an abnormal return to the nucleus, that part of the current contents of memory which is used by SPD/DOS (excluding locations X'ØØØØ' - X'ØØFF') is saved on track 2 of unit zero if unit zero is not write protected. Track 2 of every diskette is reserved for this function. The ZAP utility may be used to inspect, modify, restart or save this core image. The saved core image is unaffected by operation of other utility programs (except FORMAT, ASSEMBLE and RASSEMBL) or the nucleus. A manual boot has no effect on the image if unit zero is write protected. Execution of other application programs, especially on the 20/20, may destroy the saved image.

## USE OF INDICATOR LIGHTS

Following loading of the nucleus the rightmost keyboard indicator light is lit following a manual boot. The next to rightmost indicator is lit if a program returns to the nucleus in an abnormal manner. All indicators are off following a normal return to the nucleus. The core image is saved, as described above, in either situation which causes an indicator to be lit. Entering any command causes all indicators to be turned off.

## AUTO FILE

The source file name AUTO (for automatic) is reserved for a special use. If a diskette on unit zero is booted manually or an abnormal nucleus return occurs and there is a source file named AUTO on unit zero, then commands are automatically read from this file as though .F AUTO had been specified. If there is an object file named AUTO, then it is loaded automatically as though the command AUTO has been given.

# SECTION III

# UTILITY PROGRAMS

This section contains operator instructions for the SPD/DOS utilities. Each utility program is a standard object file which is loaded from the nucleus by an appropriate command, possibly including options and parameters. On completion of its task, each utility returns control to the DOS nucleus, signalling successful completion or returning an error code as appropriate.

## NOTATION

A word surrounded by angle brackets, e.g. < label > represents a generic type of operand. The ensuing text describes the allowed possibilities for the operand.

A section of syntax enclosed in square brackets, e.g. [code], indicates an optional part of the construction which may be omitted. The text explains the effect of including or excluding such components.

## USE OF PRINTER

If a utility program uses the printer and a printer is configured, the printer must be ready. If the printer is not ready, the program will "hang" until it is made ready. This is not considered an error condition. In fact, the operator may unready the printer at any time for forms adjustment, etc., and execution will resume as soon as the printer is readied again.

If no printer is configured, then utilities using the printer will execute normally except that no printout will be generated.

## VERIFY OPTION

Nearly all the utilities permit the use of the V letter option to

specify that writing to the diskette is performed with the verify (reread check) option. Use of this option slows down processing considerably and experience has shown the need for such verification to be minimal. Thus the use of the V option should be avoided if possible. In particular, using the V option all the time "to be sure" severely degrades the performance of the DOS utilities.

## ASSEMBLE -- Assemble Source File

The ASSEMBLE utility is used to assemble a source file to give an executable object file. The form of the call is as follows:

```
ASSEMBLE,options <sfile>,<b>,<ofile>,<olabel>
AS,options <sfile>,<b>,<ofile>.<label>
```

<sfile> specifies the source program. If the program is in a single file <sfile> is a normal format DOS filename. If the source program is contained in more than one file, <sfile> is given in the format:

<fname> .<dsn1> .<dsn2> [.<dsn3>]

<fname> specifies the name of all the files (which must be the same) and the unit of the initial file. <dsn1>,<dsn2> (and <dsn3> if three files are used) specify in sequence the serial numbers of the disks containing the files. As shown, the parts of this parameter are separated by periods.

<b>  is 4 hexadecimal digits (0-9, A-F) specifying the value of the assembly parameter $B.

<ofile>  is the name of the output object file to be created.

<olabel>  is the label of the object file to be created.

If the fourth parameter is omitted, the label on the object file is copied from the label of the source input file.

If the third and fourth parameters are omitted, the name and label of the object file are the same as those of the source input file.

If all but the first parameter are omitted, then $B is set to X'000' and the name and label of the object file are the same as those of the source input file.

If, during the assembly, a disk must be mounted, a pause will occur for mounting. The disk containing the work file, object file and ASSEMBLE load file must never be reloaded.

ASSEMBLE requires a D (data) type work file to be established prior to initialization of the assembly. The file is called WORK and may be conveniently created using the CREATE utility. The minimum size is 6 tracks. The number of symbols and cross references which can be handled is a function of the size of this work file as shown in Table 3-1.

Table 3-1

ASSEMBLE Work File Capacity

| Tracks | Maximum Symbols | Maximum Cross References |
|---|---|---|
| 6 | 100 | 2048 |
| 9 | 350 | 3072 |
| 12 | 600 | 4096 |
| 15 | 850 | 5120 |
| 18 | 1100 | 6144 |
| 21 | 1350 | 7168 |
| 24 | 1600 | 8192 |
| 27 | 1850 | 9216 |
| 30 | 2100 | 10240 |
| 33 | 2350 | 11264 |
| 36 | 2600 | 12288 |
| 39 | 2850 | 13312 |
| 42 | 3100 | 14336 |
| 45 | 3350 | 15360 |
| 48 | 3600 | 16384 |

Option Letters

The <option> field consists of one or more of the following

letters in any order.

A Alternate unit. The object file is generated on the opposite unit from that implied by the call.

B Both pass list. A listing is given during pass 1 as well as pass 2. Usually used only for diagnosing system errors.

C Clean list. LIF 0 mode is enforced throughout the assembly regardless of the occurrence of LIF operations in the source program.

E Erase. Any previous object file of the same name is erased. In the absence of this option, it is an error to have such a duplicate file name.

F Full list. LIF 2 mode is enforced throughout the assembly regardless of the occurrence of LIF operations in the source.

G Generate included code. LIN 1 mode is enforced throughout the assembly regardless of the occurrence of LIN operations in the source program.

H Hold included code. LIN 0 mode is enforced throughout the assembly regardless of the occurrence of LIN operations in the source program.

I Inhibit object. The assembly proceeds normally, but output of the object file is inhibited.

K Kill hash comments. LIST 2 mode is enforced throughout the assembly regardless of the occurrence of LIST operations in the source program.

L List mode. LIST 3 mode is enforced throughout the assembly regardless of LIST operations in the source program.

N No printer. "Listing" output is written to diskette using a pre-existing source file called LIST. The CREATE utility with the S option may be used to create this file. EDIT may be used to examine the file following assembly.

P Paper save. LIST 1 mode is enforced throughout the assembly regardless of the occurrence of LIST operations in the source program.

Q Quick assembly. XREF 0 mode is enforced throughout the assembly and no cross reference output is generated.

R Reference unassembled. XREF 2 mode is enforced throughout the assembly regardless of the occurrence of XREF operations in the source program.

S Short List. LIST 0 mode is enforced throughout the assembly regardless of the occurrence of LIST operations in the source program.

T Table of contents. A table of contents showing the initial page number for each listed sub-title line is printed immediately before the listing of the first sub-title line in the program. This allows header comments to precede the table of contents.

U Unlist deleted code. LIF 1 mode is enforced throughout the assembly regardless of tab occurrence of LIF operations in the source program.

V Verify. Object file output is verified using a reread check. Other disk write operations (to the WORK and LIST files) are never verified.

X Xref. XREF 1 mode is enforced throughout the assembly regardless of the occurrence of XREF operations in the source program.

## File Allocation

The assembler will work no matter how the files are positioned, but the following rules should be followed for maximum efficiency.

The WORK file and ASSEMBLE program load file should be on one unit and the source on the other unit in a two disk system. ASSEMBLE will preferentially select the WORK file

on the unit opposite to the source if WORK files exist on both units.

The WORK file should immediately follow the ASSEMBLE program load file. This is of particular importance from an efficiency point of view on machines with more than 4K memory (SPD 10/24, 20/20). The output object file may be on either unit with little impact on efficiency.

The LIST file (N option set) may be on either unit. ASSEMBLE preferentially selects the LIST file on the unit opposite to the source if LIST files exist on both units. In the case where a full listing is obtained on disk, an approximate guide is to make the LIST file one and a half times as large as the source file itself.

If disks must be reloaded during assembly (more than one source file), then the WORK file, LIST file, ASSEMBLE program load file and generated object file must all be on one unit, the unit which is not reloaded.

## Definition of Standard Symbols

Standard DOS symbols, as defined in the DOS Programmers Reference Manual, Order Number MS-7178, may be referenced in an

assembly without being defined.

These definitions are accessed only for otherwise undefined symbols, thus the program is free to use names of standard symbols for its own purposes.

Standard symbols do not count towards the limits shown in Table 3-1.

## Display Messages

A one line display is active throughout the assembly. The first forty characters contain the version number identification for ASSEMBLE. The remaining twenty-four characters are used to display various messages as follows:

| | |
|---|---|
| INITIALIZATION | Displayed during the assembly initialization process. |
| PASS: 1 REC: nnnnn ERR: mmm | Displayed throughout pass one. nnnnn is the number of the current record. mmm is the number of lines with errors detected. Note that not all errors are detected in pass one. |
| PASS: 2 REC: nnnnn ERR: mmm | Displayed throughout pass two. nnnnn is the record number, mmm is the number of lines with errors. |
| CROSS REFERENCE ERR: mmm | Displayed during cross reference table output. mmm is the total number of lines with errors. |

| | | |
|---|---|---|
| TERMINATION | ERR:mmm | Displayed during assembler termination processing. mmm is the number of lines with errors. |
| MOUNT UNIT=X DSN=nnnnnnnn | | Displayed during pause for disk remount in the case where the source program is split over more than one diskette. The required diskette should be mounted on the indicated unit and then the space bar pressed. |

Examples

AS,EX 1.FRED

This command causes the source file FRED on unit 1 to be assembled generating an object file on unit 1 also called FRED with the same label as the source file. The value of $B is set to X'000'. XREF 1 mode will be enforced and any previously existing object file called FRED on unit 1 will be erased before generating the new object file output.

ASSEMBLE,L XY,1000,1.YZ,VERSION 1

This command assembles the source file XY on the currently selected unit generating an object file with name YZ on unit 1 with label VERSION 1. The value of $B is set to X'1000' and LIST 3 is enforced throughout the assembly regardless of the occurrence of LIST statements in the assembly source.

AS,A XX.DISK1.DISK2.DISK3,177A

The source program is made up of three files all called XX on disks DISK1, DISK2 and DISK3. $B will be set to X'177A'. The A option ensures that the object file is generated on the opposite unit to the remount unit as required.

AS,INQS PROG1

The source file PROG1 on the currently selected unit is assembled with no object file output. An errors-only listing will be written to the LIST file which must exist before starting the assembly. A cross reference listing will not be written to the LIST file.

CNFG --- Configure System

CNFG is used to provide configuration information to the system. It can be used only in keyboard mode and is called as follows:

```
CNFG
CN
```

The configuration parameters are stored in the label record of every SPD/DOS diskette. When a disekette is booted, these parameters are read and stored in the diskette buffer. They remain in effect until the next manual or abnormal boot. The CNFG utility may be used to change the parameters in the diskette buffer only (for temporary changes, the old parameters being restored on the next manual boot) or both in the diskette buffer and on the diskette itself so that the

changes are made permanent.

The FORMAT utility always copies the current parameters to the new diskette. Thus it is normally sufficient to use CNFG once on receipt of the system disk and then let the identical parameters propagate to other disks in the system.

CNFG operates by a question/answer dialog. The questions and possible responses are self-explanatory.

CNFG accepts no options. Rewrites of label records are always performed with a verify check.

## COPY -- Copy File

The COPY utility is used to copy a file from one diskette to another. It may also be used to copy files to and from a wide range of external peripheral devices. The form of the COPY command is as follows:

```
COPY,options <from>,<to>,<label>
CO,options <from>,<to>.<label>
```

The < from > parameter designates the file to be copied. The to parameter designates the destination to which it is to be copied. The third parameter, < label >, may be specified to provide a label for the output file. The third parameter may be omitted in which case the label on the output file is either the same as the input

file label, or if no input file label exists, then the output label is set to all blanks.

The from and to parameters can be normal diskette file designations in which case the appropriate file is referenced depending on the setting of the options. To designate external peripheral devices the <from> and/or <to> parameters are given in the form:

.TTC

TT is the device type specified by two letters as follows:

PP - paper tape punch

PR - paper tape reader

CT - cassette tape

MT - half inch magnetic tape

CR - card reader

CP - card reader/punch

The C is the channel number to which the devices are attached. This is specified as a single hexadecimal digit from Ø to 7 in case of the 10/20, or 1 to C in the case of the 20/20.

The following sections describe operating procedures for the various peripherals:

Paper Tape Input (PR)

The paper tape must be positioned at the start of the file to be read. Source files are basically in the same format as command files with the addition that \ (reverse slash) may be used as a tab character. Tab stops are set to 1-10-16-30 for use in assembly programs. Object format is identical to that output by the H716 assembler. If a bootstrap loader is present, it is skipped. A CO2Ø (error reading external medium) is posted if a checksum error is encountered.

Paper Tape Output (PP)

The output generated is compatible with the paper tape format described above and includes a 100 character null leader at the beginning and end of the tape. Output to a paper tape punch may be verified via a special use of the V option.

Cassette Tape Input (CT)

Files are read in a format compatible with that generated by the cassette tape output below. Error reading external medium is signalled if the tape is in manual mode, or if it stalls during read, or if a dropout/parity error is detected. Note: This information applies only to the sykes cassette unit. To copy files to the 20/20 (built-in) cassette, do not use COPY. Instead ZAP with the W option must be used.

Cassette Tape Output (CT)

One file is written on one cassette. The cassette is rewound before starting if necessary. The format for source files is the same as that used by the SPD cassette tape assembler and editor. Object files, which are not compatible with the cassette tape assembler format, are written in 512 byte blocks. For unsegmented programs, the output includes a bootstrap loader which uses device address 12 on a 20/20 and device address 1 or 3 (whichever is ready) on a 10/20. A CO21 (error writing external medium) is signalled if the tape is in manual mode, or if it stalls during write, or if it is write protected.

Magnetic Tape Input (MT)

The file is read without positioning the tape, thus the tape must be positioned to the beginning of the desired file using TMOVE. After reading the last file, the tape is left positioned ready to read the next file. Source file format is compatible with that used by the H716 assembler system. Error reading external medium is signalled if the tape is off line, or if the formatter is switched off, or a read error is encountered, or if the file format is incorrect.

Magnetic Tape Output (MT)

The file is written without positioning the tape, thus the tape must be positioned to the beginning of the desired file using TMOVE. After writing the last file, two end of file marks are written and the tape

is left positioned following the first end of file (ready to write the next file). Error writing external medium is signalled if the tape is offline or the formatter is switched off or a write error is encountered.

Punched Card Input (CR or CP)

On the card reader, the deck is loaded into the hopper and the RESET button pressed. On the reader-punch the deck is loaded into the primary (rear) hopper and the reader reset. Following reading the last card (.E end of file), the card must be runout manually on the reader punch. Error reading external medium is signalled if a checksum error is detected in an object deck. Other card reader errors cause COPY to pause for appropriate operator intervention.

Punched Card Output (CP)

The printing-reader-punch must be cleared and reset before starting. Blank cards are fed from the secondary (front) hopper. In the case of object files for unsegmented programs, the output includes a bootstrap loader. This loader is compatible with the reader-punch or the card reader. It uses the same device address as the punch unless a fourth digit is given on the device specification. For example, .CP36 specifies a file to be written to the punch on channel 3 with bootstrap loader for device address 6. Error writing external medium is signalled if an

attempt is made to punch on non-blank cards. In other error situations COPY waits for appropriate operator intervention.

It should be noted that the ZAP utility has the capability of writing core images as object files on diskette and also building core images from object files. This may be used as an additional mechanism for moving object files to and from external media in formats other than those supported by the COPY program.

Another form of the command, applicable only if the S option is specified, is as follows:

COPY, options , <to >, <label>
CO, options , < to> , < label >

This copies a cource file from the current command input source (keyboard, cards, tape, disk) into a specified file. Note that < label > is required in this case. The end of such an "imbedded" input file is signalled by a line with .E in columns 1, 2. This .E line is not included in the output file.

In the case of a diskette to diskette copy, an abbreviated form is available if the name and label of the output file is the same as the input file, as often happens when copying a file from one unit to another:

COPY, options < file >, < unit>
CO, options <file> , < unit >

In this case < file> is the file to be copied and <unit> is a digit specifying the unit on which the output is to be written.

In general, COPY attempts to complete the copy operation even if errors are detected, the appropriate error code being posted on completion. If the output is a diskette file and such an error occurs, the file is closed and marked with ? (error) status.

The following letter options are recognized by COPY:

D Data file. A data file is to be copied.

E Erase. This option is meaningful only if <to> specifies a diskette file. In this case, if there is a file with identical type and name to the output file, it will be erased on successful completion of the copy. If the E option is not specified, then it is an error to have a duplicate file name.

L Logical copy. This option is relevant only in the disk to disk case. It forces the copy to occur by logical records instead of the normal half sector by half sector mode. Its main use occurs in conjunction with the O (object) option. The effect in this case is to reorganize the object file into the more efficient segment-zero-first arrangement, thus optimizing initial load and segment load operations. The L option may not be used in conjunction with the M option.

M Mount. This option is meaningful only if the <to> and <from> fields specify diskette files. If the files are on different units, then COPY pauses once before starting, thus allowing COPY to be loaded from a separate disk. If the files are on the same unit, then COPY will pause before every read and every write thus allowing a diskette-to-diskette copy on a one unit system. Following each pause the appropriate disk or disks are loaded and the space bar pressed to continue the COPY.

O Object file. An object file is to be copied.

R Relocatable file. A relocatable file is to be copied.

S Source file. A source file is to be copied.

V Verify. This option is meaningful only if <to> specifies a diskette file. All write operations will be performed with the verify option (reread check).

Exactly one of the options D, O, R, or S must be used to specify the type of file to be copied.

Examples:

```
COPY,RE AFILE,1
```

The relocatable file AFILE on the currently selected unit (presumably unit zero) is copied to unit one with the name and label unchanged. Any previous relocatable file AFILE on unit one is erased.

COPY, O. MT3, . CT4, LQ

An object file is copied from a magnetic tape on channel 3 to a cassette tape on channel 4 with output label LQ.

CO, D Ø. Q, 1. XY. RS

The data file named Q on unit zero is copied to unit one where it is named XY with label RS.

CO, S . TRØ, XF

A paper tape source file is copied from a tape reader on channel zero to a new file XF on the currently selected unit. The label will be taken from the first image of the tape.

Following a copy to a paper tape punch (for example on channel 6)

CO, O 1. X, . PP6

the verify option may be used as follows. Manually remount the tape on a paper tape reader on channel 1, and verify the copy by:

CO, OV 1. X, . PR1

## CREATE -- Create Data File

The CREATE utility is used to establish a new data or source file. The form of the call is as follows:

CREATE, options n , m , file , label
CR, options n , m , file , label

n is the number of contiguous tracks to be allocated to the data file. All sections of the file are initialized to hexadecimal zeros.

m is the sector interlace factor. It should be set to zero or some odd decimal value in the range of 1 to 31.

An alternate form of the command omits all but the first parameter:

CREATE, options <n>
CR, options <n>

In this case the following default options for the last three parameters are used:

| | D option specified | S option specified |
|---|---|---|
| <m> | Ø | 11 |
| <file> | WORK | LIST |
| <label> | blanks | blanks |

The letter options accepted by CREATE are as follows:

D Data. This option causes CREATE to establish an empty data file.

E Erase file. If this option is specified and if a data file of the same name as the specified file exists on a specified unit, then the old file is erased on successful completion. If this option is not specified and a duplicate file exists, this is considered an error condition.

S Source. This option causes CREATE to establish an empty source file. The only DOS utilities requiring the use of this option are ASSEMBLE and RASSEMBL in conjunction with the N option. Utilities like EDIT automatically create required output files.

V Verify. All diskette write operations are carried out using the verify (reread check) option.

Note: If neither D nor S is specified, D is assumed.

```
CREATE,E 3,7,1.XY,L
```

A data file called XY with label L is created on unit one, any previous data file of this name being erased. The size is three tracks with a sector interlace factor of seven.

```
CR,D 18
```

An 18 track data file called WORK with a sector interlace factor of zero and an all blank label is created on the currently selected unit. This file would be suitable for use as the work file for ASSEMBLE.

```
CR,S 6
```

A six track empty source file called LIST is created with a sector interlace factor of 11 and an all blank label. This file would be suitable for the listing output of an ASSEMBLE,N operation.

## DCOPY -- Copy Diskette File

The DCOPY utility is used to copy a file from one diskette to another. The form of the DCOPY command is as follows:

```
DCOPY, options <from >, <to >, <label >
DC, options <from> , < to> , < label>
```

The < from > parameter designates the file to be copied. The < to> parameter designates the destination to which it is to be copied. The third parameter, <label> , may be specified to provide a label for the output file. The third parameter may be omitted in which case the label on the output file is either the same as the input file label, or if no input file label exists, then the output label is set to all blanks.

An abbreviated form is available if the name and label of the output file is the same as the input file, as often happens when copying a file from one unit to another:

```
DCOPY, options < file> , < unit>
DC, options < file> , < unit>
```

In this case <file> is the file to be copied and < unit> is a digit specifying the unit on which the output is to be written.

In general, DCOPY attempts to complete the copy operation even if errors are detected, the appropriate error code being posted on completion. If such an error occurs, the file is closed and marked with ? (error) status.

The following letter options are recognized by DCOPY:

D Data file. A data file is to be copied.

E Erase. If there is a file with identical unit, type and name to the output file, it will be erased on successful completion of the copy. If the E option is not specified, then it is an error to have a duplicate file name.

M Mount. If the files are on different units, then DCOPY pauses once before starting, thus allowing DCOPY to be loaded from a separate disk. If the files are on the same unit, then DCOPY will pause before every read and every write, thus allowing a diskette-to-diskette copy on a one unit system. Following each pause the appropriate disk or disks are loaded and the space bar pressed to continue the DCOPY.

O Object File. An object file is to be copied.

R Relocatable file. A relocatable file is to be copied.

S Source file. A source file is to be copied.

V Verify. All write operations will be performed with the verify option (reread check).

Exactly one of the options D, O, R or S must be used to specify the type of file to be copied.

Examples:

```
DCOPY,RE AFILE,1
```

The relocatable file AFILE on the currently selected unit (presumably unit zero) is copied to unit one with the name and label unchanged. Any previous relocatable file AFILE on unit one is erased.

```
DCOPY,O, ABC,DEF
```

The object file ABC on the currently selected unit is copied onto the same unit with the new name DEF and its original label.

```
DC,DV O.Q,1.XY,RS
```

The data file named Q on unit zero is copied to unit one where it is named XY with label RS. All write operations are performed with the verify option (reread check).

EDIT -- Edit Source File

The EDIT utility is used to update existing source files and create new source files. The editing process is interactive and screen oriented. This EDIT is used online in keyboard command mode. For updating source files in batch mode, the UPDATE utility should be used. The form of the command to edit an existing file is:

EDIT, options < ifile> , < ofile >, <olabel>
ED, options < ifile> , < ofile >,< olabel>

<ifile> is the name of the existing input file.

< ofile> is the name of the new output file to be created.

< olabel> is the label of ofile. Note that the < olabel > parameter is required.

To create a new file, the command is:

EDIT, options < , ofile > ,< olabel>
ED, options <, ofile > , < olabel>

If this form is used, then any editor operation which would read a record from the input file reads a blank record instead.

A third form of the command specifies only one parameter:

EDIT <ifile >
ED <ifile >

In this case EDIT can be used to inspect or page through the input file < ifile >. EDIT controls such as SEARCH operate normally but no output is generated and the use of graphic data keys is inhibited.

The editor basically works on a sequential copy basis. The keyboard may be used to alter the displayed records before they are written to the output file. The layout of the various possible keyboards, including special function keys are shown in Figures 3-1, 3-2, 3-3 and 3-4.

The meaning of the special function keys is as follows:

DONE* If in normal mode (see MERGE key), then the rest of the input file is copied to the output file (including the current data on the screen), and control returns to the system. If in merge mode, the remainder of the merged file is copied to the output. After completing this copy, input from the original source file is resumed. If the diskette must be reloaded, it should be changed after the copy is completed.

MERGE* Input from the normal input file is suspended and a new unit and file name will be requested to be typed from the keyboard, terminated by LINE RETURN. The editor then enters merge mode with input from the merge file. If a diskette must be reloaded, this should be done

before pressing MERGE.

QUIT* — Exactly like DONE except that the copy of the rest of the input file is omitted.

SEARCH — A search key of up to nine characters is entered followed by DELETE PAGE, COPY PAGE, COPY LINE or DELETE LINE. The specified function will be repeated until the record searched for is read onto the screen (PAGE function) or is under the cursor (LINE function) or the end of the input file is reached. The search key is either the desired line number (1-5 decimal digits) or the initial non-blank characters of the line.

STOP — The current search or copy (from DONE function) is terminated immediately.

COPY PAGE — The current screen lines are written to the output file and a new screen of information read from the input file.

SET TAB — A tab stop is set at the current cursor position on the line. Initially tabs are set at column positions 1, 10, 16, 30. If this key is used in shifted mode, all tab stops are cleared.

| | |
|---|---|
| DELETE* PAGE | The current screen information is discarded and data read from the input file to refill the screen. |
| COPY LINE | The top screen line is written to the output file, the remaining lines are rolled up one line and a new line is read onto the bottom of the screen from the input file. |
| DUP FIELD | Characters are duplicated from the previous line until the next tab stop or end of line is encountered. |
| DUP | A single character is duplicated from the previous line. This key repeats automatically if held down. |
| TAB | The cursor is moved to the next tab stop. Intervening characters are not affected. |
| INSERT LINE | The top line is written to the output file. Lines above the current cursor are rolled up one line and a new blank line appears on the cursor line. |
| DELETE LINE | The cursor line is deleted, lines below are rolled up one line and a new input line is read onto the bottom of the screen. |
| ERASE* SCREEN | The screen is erased (blanked) from the current cursor position to the end of the screen. |

INSERT CHAR — Characters to the right of the cursor on the same line are moved one position to the right (the last being lost) and a blank is inserted at the cursor position.

DELETE CHAR — The cursor character is deleted and characters to the right of the cursor are moved one position to the left and a blank is inserted in the last column.

ERASE FIELD — Blanks are written to the next tab position.

↑ ↓ ← → — Move the cursor in the indicated direction. These functions repeat automatically if the key is held down.

REPEAT — If this key is pressed before any other key on the keyboard, the latter key repeats as long as it is held down.

LINE RETURN — The cursor is moved to the start of the next line.

Notes:

1) After the end of the input or merge file is reached, input operations generate blank lines. This is also the case if there is no input file (first EDIT parameter omitted).

2) Keys marked * are upper case keys and require the use of the shift key.

3) The editor continually displays the current input and output record counts and the cursor column number.

4) It is possible to MERGE a file into itself. This may be used to rearrange the order of records in a given file.

5) The line length is 64 characters. When an attempt is made to type in column 65:
   - the alarm is sounded, and
   - all characters beyond the 64th are truncated.

6) The EDIT program itself uses segment overlays. Thus the unit containing the EDIT program must remain loaded. In cases where multiple disks are used with the MERGE function, it is necessary to load the editor from the unit containing the output file which must itself remain loaded. Alternately, a convention may be adopted that EDIT utility is on all disks in the same track position. This obviates the need for this requirement.

The editor makes every attempt possible to close the output file if an unrecoverable output error occurs, thus allowing recovery of the output of the edit to that point. Unrecoverable input errors do not terminate the edit. Instead, a message **BAD BLOCK** is read onto the screen (corresponding to one or more unreadable records) and the

operator may use the keyboard to repair the error.

The following is a step by step procedure for merging a file from a separate disk:

1) Load editor from same disk as the output file.

2) (Edit commands for original file)

3) Load diskette with file to be merged.

4) Press MERGE.

5) Enter unit and name of file to be merged.

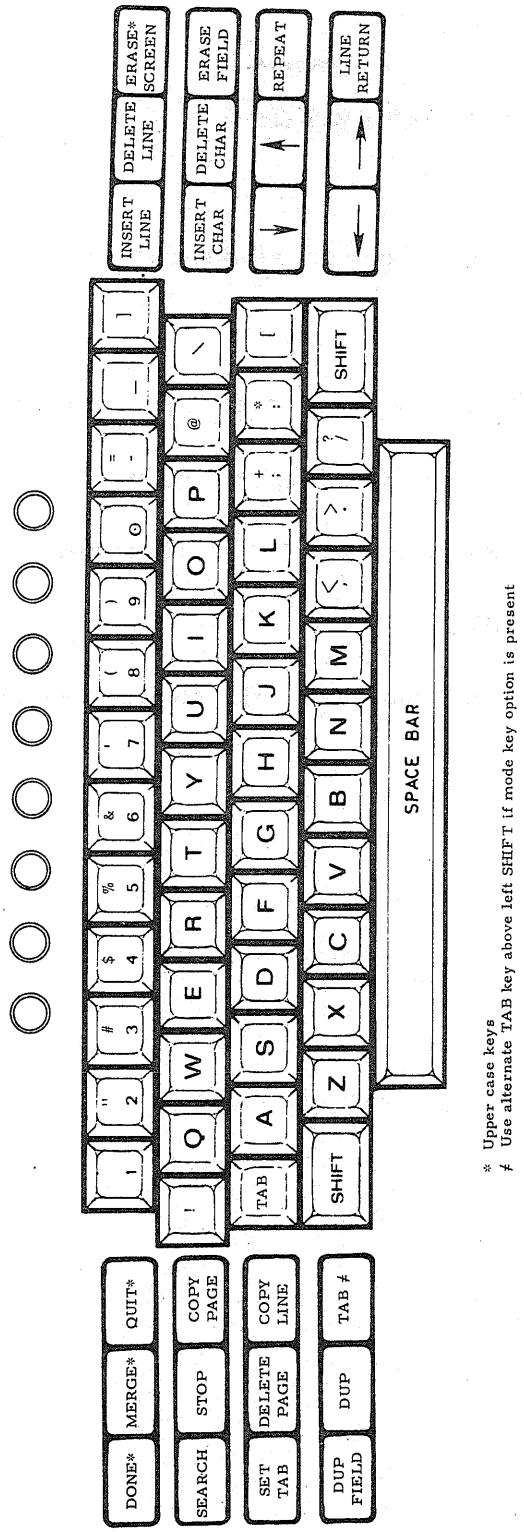



Figure 3-1. Keyboard Layout for EDIT (10/20 and 10/24 Upper Case Only Keyboard)

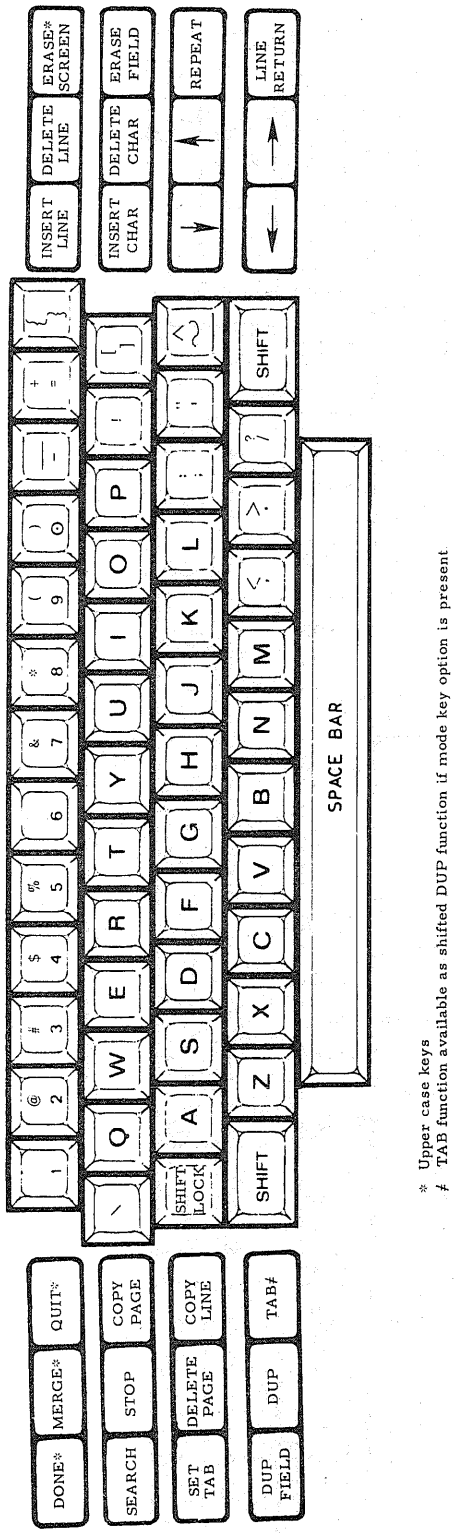



Figure 3-2. Keyboard Layout for EDIT (10/20 Upper/Lower Case Keyboard)

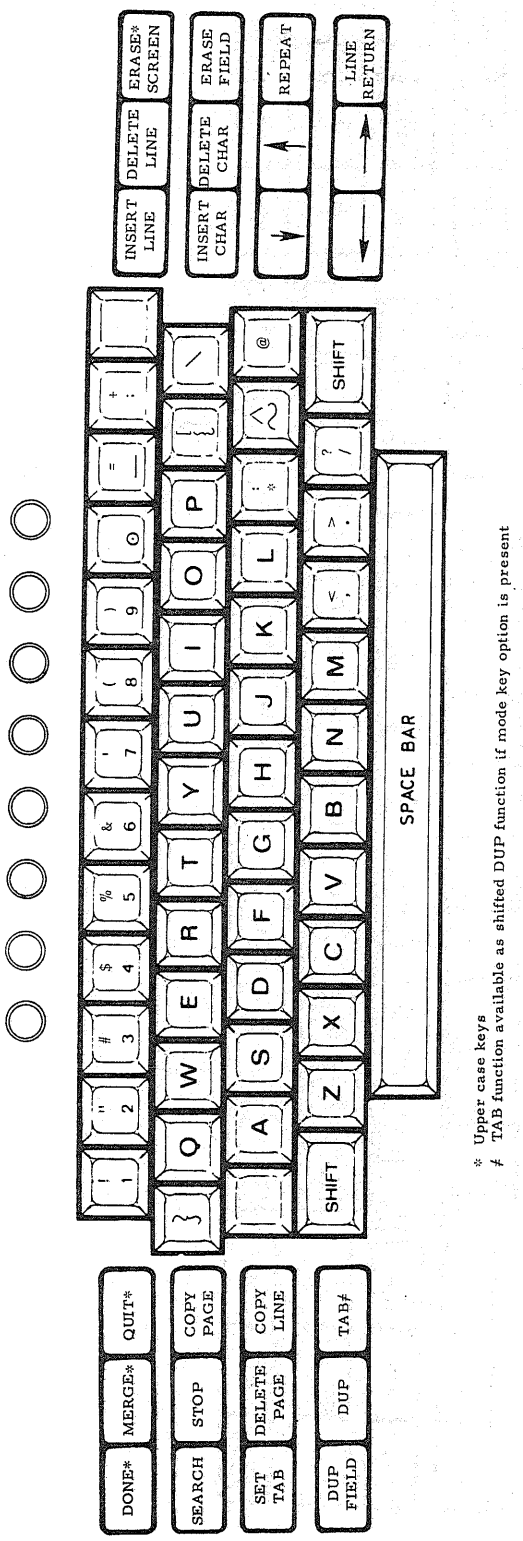



Figure 3-3. Keyboard Layout for EDIT (10/20 and 20/20 Upper Case Only Keyboard).

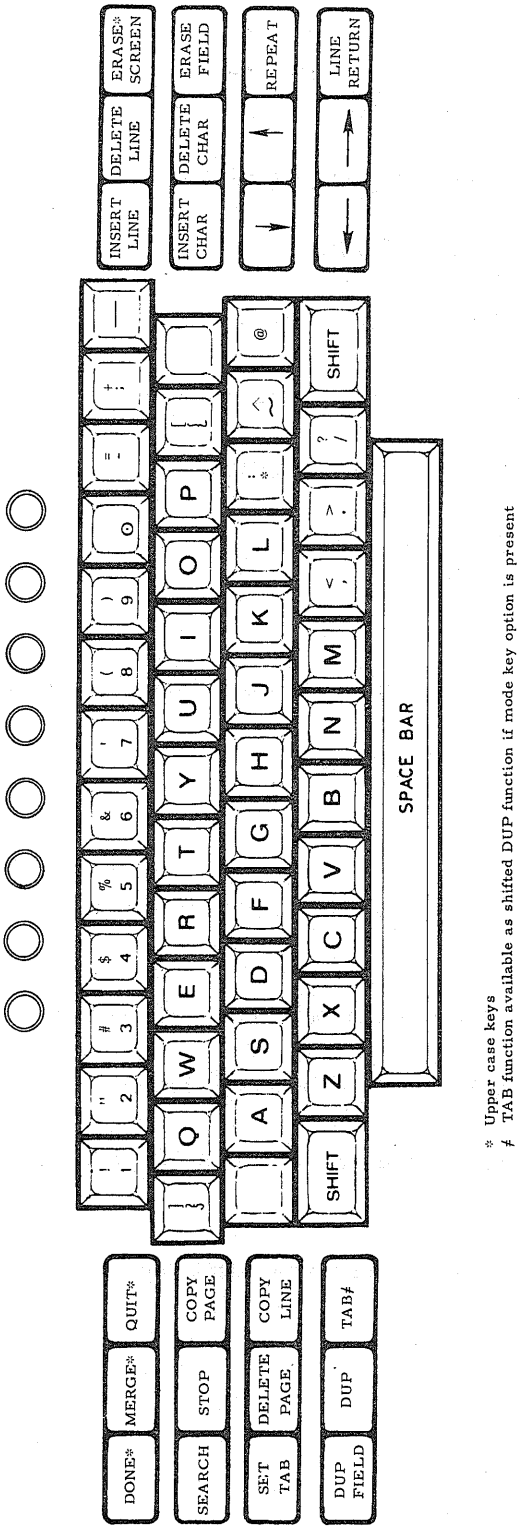



Figure 3-4. Keyboard Layout for EDIT (10/25 and 20/20 Upper/Lower Case Keyboard).

6) Press LINE RETURN.

7) (Edit commands for merged file).

8) Press DONE or QUIT.

9) Reload diskette with original input file.

10) (Edit commands continuing with original file).

11) Repeat steps 3-10 as required.

12) Press DONE or QUIT.

Note: If the specified merge file is not found, a new unit number/ file name input is solicited.

The following letter options are recognized by EDIT:

E Erase. Any existing source file with the same name as the output on the same unit is erased before starting the edit. Otherwise if is an error to have a duplicate file name.

V Verify. The output is written with the verify (reread check) option.

Examples:

```
ED,E ,1.OUT,
```

A new source file OUT is to be created on unit one with a blank label. Any previously existing source file OUT on this unit will be erased.

ED SF, 1. SF, VERSION 2.2

The source file SF on the currently selected unit (presumably zero) is to be edited with the new file being created on unit one, also called SF with label VERSION 2.2.

## ERASE -- Erase Files

The ERASE utility is used to mark one or more files as erased. The space used by these files is not immediately reclaimed (see PACK utility) but the files become inaccessible to all utilities except RENAME. The form of the command is:

ERASE, options < file > , <file> , . . . . . , < file >
ER, options <file> , < file > , . . . . . , < file >

As shown, one or more < file > parameters separated by commas may be specified. The effect of specifying more than one parameter is exactly the same as that of a series of ERASE commands with the same options and one parameter to a command.

The following letter options are accepted by ERASE:

D Data. For each specified file name, a data file is to be erased.

I Ignore. Normally it is an error to attempt to erase a non-existant file. If the I option is specified, then this condition is ignored.

O Object. For each specified file name, an object file is to be erased.

R Relocatable. For each specified file name, a relocatable file is to be erased.

S Source. For each specified file name, a source file is to be erased.

V Verify. All directory write operations will be performed with the verify (reread check) option.

Examples:

ER,O XYZ

The object file XYZ on the currently selected unit is erased.

ER,OS Ø.A,Ø.B

The object and source files named A and B on unit zero are erased.

All four files must exist or an error will be signalled.

ER, DORSI A, B, C, D, E, F, C, H, I, J, K, L

All files of any type with one letter names (A-L) are erased from the currently selected unit. Not all of these files may exist, but the I option causes this condition to be ignored.

FORMAT -- Format Diskette

FORMAT is used to format diskettes and write the nucleus and bootstrap code. The form of the command is:

FORMAT, options < unit > , < dsn >
FO, options < unit > , < dsn>

< unit> is a single digit representing the unit to be used.

< dsn > is the diskette serial number consisting of up to eight alphanumeric characters.

If the DSN is correctly written already on the disk to be formatted, then the following form of the command may be used:

FORMAT, options < unit >

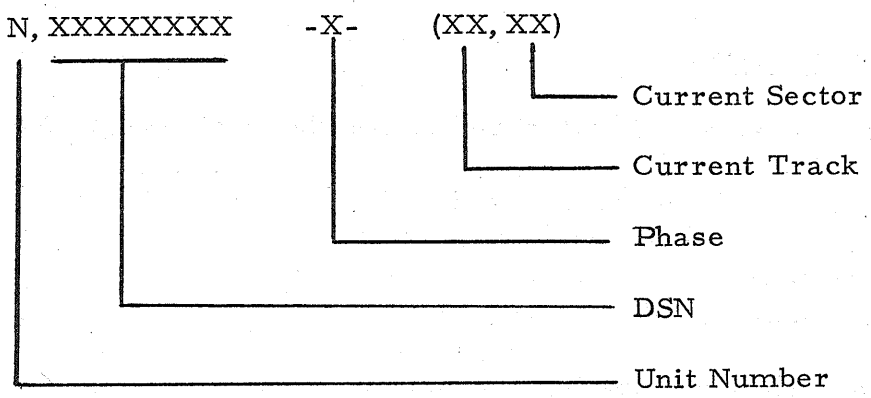
In this case, the existing DSN is retained.

The configuration parameters written to the new disk are copied from those currently active. This means that once the initial system disk is configured correctly, this correct configuration is propagated to other disks in the system. Thus the CNFG utility need in general only be used if the configuration changes.

FORMAT label... the format label is displayed on one line:

FORMAT label N, XXXXXXXX -X- (XX, XX)

Where the label elements are defined as follows:




FORMAT operates in five phases as indicated by the -X- field of the display.

-M- Pause for mounting of disk to be initialized. The disk should be mounted on the unit specified by the UNIT-X field of the display. Then the space bar is pressed to proceed. This phase only occurs if the program load unit is the same as the unit to be formatted.

-F- All sectors are formatted with initial contents of hexadecimal zeros. The (TT, SS) field of the display keeps track of the progress of this phase by showing the current track and sector in decimal.

-V- The initialization of the sectors in phase F is verified. This is done only if the V option is specified.

-W- The bootstrap and nucleus code is written to the newly initialized disk.

-E- Entered if a write error occurs so that the number of the erroneous sector is displayed. Pressing the space bar causes return to the nucleus with an appropriate error code.

The following letter options are recognized by FORMAT:

D Directory format. This option causes the formatting of the file area (tracks 3-63) to be skipped. The directory is cleared as usual. FORMAT can then be used as an easy means of reinitializing a directory. The D option speeds up this process in the case where the disk is known to be formatted correctly already.

N Nucleus. The formatting operation is restricted to tracks Ø and 2 (the nucleus and bootstrap code and core image save area). The files and file directory are unchanged. This option may be used to write a new version of DOS to an old diskette or to "mend" a diskette whose system track has gone bad as indicated by its failure to boot properly.

V Verify. The V option initiates a verify phase following formatting which checks that all sectors were correctly formatted. This verify phase is fast and thus the V option may be used freely to verify that the disk surface is undamaged.

Examples:

```
FO,V 1,DISKØ1
```

Format and verify a new diskette on unit one. The DSN will be set to DISKØ1.

FORMAT,NV Ø,99999999

Write a new copy of the nucleus and bootstrap code to the (previously formatted) disk on unit zero with verification. The files and file directory will be unchanged. The DSN will be set to all nines.

FO,D 1

Reformat the nucleus code and directory of the disk on unit 1 retaining the existing DSN.

## LIST -- List File

The LIST utility is used to list files. It also has options to list directories and generate printer alignment data. To list a file, the form of the command is:

LIST, options < file >
LI, options < file>

This causes < file > to be listed in a format appropriate to its type as follows:

D Data files are listed by sector in a logical order determined by the SIF value or, if the SIF value is zero, then by consecutive physical sectors. Each sector is listed in character form and also in hexadecimal if the H option is given.

O Object files are listed in hexadecimal by sectors.

R Relocatable files list the module directory and also the hexadecimal sector data if the H option is given.

S Source files are listed by logical records in character form with line numbers if the N option is specified.

To list the directory of a diskette in a format similar to that of the nucleus display; all options are omitted:

```
LIST <unit >
LI <unit>
```

<unit> is a single digit giving the unit number of the diskette whose directory is to be listed. If <unit> is omitted, the currently selected unit is assumed.

The use of the A option:

```
LIST, A
LI, A
```

generates alignment data for the printer and is used only in online (keyboard command entry) mode. For a character printer, the operator is invited to type an F or V for form eject or vertical tab alignment respectively. For a line printer, a digit from 1-4 is typed to align the corresponding forms channel. The S key prints test data with no form feed. The alignment data may be generated repeatedly until the X key is pressed to return to the nucleus.

The following letter options are recognized by LIST:

A Align. A printer alignment sequence is initiated under keyboard control. This must be the only option if it is specified.

D Data. File is a data file.

H Hexadecimal. Listing for D type file given in characters and hexadecimal form (normally only in characters, with unprintable characters replaced by *). Listing for R type file includes hexadecimal sector data.

I Ignore errors. Bad data records will cause appropriate messages to be printed instead of causing error termination.

N Number. Lines of S files will be numbered.

O Object. File is an object file.

R Relocatable. File is a relocatable file.

S Source. File is a source file.

If the file parameter is given, then one of the options D, O, R, S must be present. If more than one option is present, then more than one file may be listed with one LIST command where the files have the same name and different types.

Examples:

LIST,HDI 1.WORK

The data file named WORK on unit one will be listed in hexadecimal and character form. Disk read errors will print a message and continue.

LI,SON DS

Both the source and object files named DS on the currently selected unit will be printed, the source file with line numbers.

LIST 1

The file directory of the diskette on unit one will be listed.

LI,A

A printer alignment sequence is initiated.

## PACK -- Pack Diskette

The PACK utility is used to reclaim space on a diskette by removing erased files and packing the active files. This may be performed in place (i.e., rewriting a single diskette) by a command of the form:

PACK,options <unit>
PA,options < unit >

< unit> is the number of the unit on which the diskette to be packed is mounted. If < unit > is omitted, the currently selected unit is assumed.

PACK may also be used to copy the files to be retained onto a new diskette by using the command:

PACK, options < unit > , < unit >
PA, options <unit> ,< unit>

The first < unit > parameter is the diskette to be copied. The second < unit > parameter is the diskette to receive the copy. If an error occurs while copying a file, the file status on the output unit is set to "?" (error), the PACK is completed, and then the error is signalled.

The following letter options are recognized by PACK:

D Retain all data files.

O Retain all object files.

R Retain all relocatable files.

S Retain all source files.

V Verify. Perform all rewrite and directory update operations with the verify (reread check) option.

If none of D, O, R, or S are specified, then all active files are retained.

Examples:

PA, DO Ø

The currently selected unit is packed in place. Only active data and object files are retained. Any source and relocatable files will be eliminated.

PACK Ø, 1

All active files are copied from unit zero to unit one. Any files previously written on unit one are lost.

## RASSEMBL -- Relocatable Assembler for Source Files

The RASSEMBL utility is used to assemble a source file to give a relocatable file. The relocatable file is prepared for later processing by the ASSEMBLE utility. The form of the call is as follows:

RASSEMBL, options <sfile> , < b >,< rfile> , < rlabel >
RA, options < sfile >,< b > , < rfile >,< rlabel >

<sfile> specifies the source program. If the program is in a single file <sfile> is a normal format DOS filename. If the source program is contained in more than one file, <sfile> is given in format:

<fname> <dsnl> <dsn2>[< dsn3>]

< fname>specifies the name of all the files (which must be the same) and the unit of the initial file. < dsnl > , <dsn2> (and <dsn3> if three files are used) specify in sequence the serial numbers of the disks containing the files. As shown, the parts of this parameter are separated by periods.

< b > is 4 hexadecimal digits (0-9, A-F) specifying the value of the assembly parameter $B.

<rfile > is the name of the output relocatable file to be created.

<rlabel> is the label of the relocatable file to be created.

If the fourth parameter is omitted, the label on the relocatable file is copied from the label of the source input file.

If the third and fourth parameters are omitted, the name and label of the relocatable file are the same as those of the source input file.

If all but the first parameter are omitted, the $B is set to X'0000' and the name and label of the relocatable file are the same as those of the source input file.

If, during the relocatable assembly, a disk must be mounted, a pause will occur for mounting. The disk containing the work file relocatable file and RASSEMBL load file must never be reloaded.

RASSEMBL requires a D (data) type work file to be established prior to initialization of the relocatable assembly. The file is called WORK and may be conveniently created using the CREATE utility. The minimum size is 6 tracks. The number of symbols and cross references which can be handled is a function of the size of this work file as shown in Table 3-2.

Table 3-2

RASSEMBL Work File Capacity

| Tracks | Maximum Symbols | Maximum Cross Reference |
|---|---|---|
| 6 | 80 | 2048 |
| 9 | 240 | 3072 |
| 12 | 400 | 4096 |
| 15 | 560 | 5120 |
| 18 | 720 | 6144 |
| 21 | 880 | 7168 |
| 24 | 1040 | 8192 |
| 27 | 1200 | 9216 |
| 30 | 1360 | 10240 |
| 33 | 1520 | 11264 |
| 36 | 1680 | 12288 |
| 39 | 1840 | 13312 |
| 42 | 2000 | 14336 |
| 45 | 2160 | 15360 |
| 48 | 2320 | 16384 |

Option Letters

The <options> field consists of one or more of the following letters in any order.

A Alternate unit. The relocatable file is generated on the opposite unit from that implied by the call.

B Both pass list. A listing is given during pass 1 as well as pass 2. Usually used only for diagnosing system errors.

C Clean list. LIF 0 mode is enforced throughout the relocatable assembly regardless of the occurrence of LIF operations in the source program.

E Erase. Any previous relocatable file of the same name is erased. In the absence of this option, it is an error to have such a duplicate file name.

F Full list. LIF 2 mode is enforced throughout the relocatable assembly regardless of the occurrence of LIF operations in the source.

I Inhibit relocatable output. The relocatable assembly proceeds normally, but output of the relocatable file is inhibited.

K Kill hash comments. LIST 2 mode is enforced throughout the relocatable assembly regardless of the occurrence of LIST operations in the source program.

L List mode. LIST 3 mode is enforced throughout the relocatable assembly regardless of LIST operations in the source program.

N No printer. "Listing" output is written to diskette using a pre-existing source file called LIST. The CREATE utility with the S option may be used to create this file. EDIT may be used to examine the file following relocatable assembly.

P Paper save. LIST 1 mode is enforced throughout the relocatable assembly regardless of the occurrence of LIST operations in the source program.

Q Quick relocatable assembly. XREF 0 mode is enforced throughout the relocatable assembly and no cross reference output is generated.

R Reference unassembled. XREF 2 mode is enforced throughout the relocatable assembly regardless of the occurrence of XREF operations in the source program.

S Short List. List 0 mode is enforced throughout the relocatable assembly regardless of the occurrence of LIST operations in the source program.

T Table of contents. A table of contents showing the initial page number for each listed sub-title line is printed immediately before the listing of the first sub-title line in the program. This allows header comments to precede the table of contents.

U Unlist deleted code. LIF 1 mode is enforced throughout the assembly regardless of the occurrence of LIF operations in the source program.

V Verify. Relocatable file output is verified using a reread check. Other disk write operations (to the WORK and LIST files) are never verified.

X Xref. XREF 1 mode is enforced throughout the relocatable assembly regardless of the occurrence of XREF operations in the source program.

## File Allocation

The relocatable assembler will work no matter how the files are positioned, but the following rules should be followed for maximum efficiency.

The WORK file and RASSEMBL program load file should be on one unit and the source on the other unit in a two disk system. RASSEMBL will preferentially select the WORK file on the unit

opposite to the source if WORK files exist on both units.

The WORK file should immediately follow the RASSEMBL program load file. This is of particular importance from an efficiency point of view on machines with more than 4K memory (SPD 10/24, 20/20).

The output relocatable file may be on either unit with little impact on efficiency.

The LIST file (N option set) may be on either unit. RASSEMBL preferentially selects the LIST file on the unit opposite to the source if LIST files exist on both units. In the case where a full listing is obtained on disk, an approximate guide is to make the LIST file one and half times as large as the source file itself.

## Definition of Standard Symbols

Standard DOS symbols, as defined in the DOS Programmers Reference Manual, Order Number MS-7178, may be referenced in a relocatable assembly without being defined.

These definitions are accessed only for otherwise undefined symbols, thus the program is free to use names of standard symbols for its own purposes.

Standard symbols do not count towards the limits shown in Table 3-1.

## Display Messages

A one line display is active throughout the relocatable assembly, The first forty characters contain the version number identification for RASSEMBL. The remaining twenty-four characters are used to display various messages as follows:

| | |
|---|---|
| INITIALIZATION | Displayed during the relocatable assembly initialization process. |
| PASS: 1 REC: nnnnn ERR: mmm | Displayed throughout pass one. nnnnn is the number of the current record. mmm is the number of lines with errors detected. Note that not all errors are detected in pass one. |
| PASS: 2 REC: nnnnn ERR: mmm | Displayed throughout pass two. nnnnn is the record number, mmm is the number of lines with errors. |
| CROSS REFERENCE ERR: mmm | Displayed during cross reference table output. mmm is the total number of lines with errors. |
| TERMINATION ERR: mmm | Displayed during relocatable assembler termination processing. mmm is the number of lines with errors. |
| MOUNT UNIT=X DSN=nnnnnnnn | Displayed during pause for disk remount in the case where the source program is split over more than one diskette. The required diskette should be mounted on the indicated unit and then the space bar pressed. |

Examples

RA,EX 1.FRED

This command causes the source file FRED on unit 1 to be assembled generating a relocatable file on unit 1 also called FRED with the same label as the source file. The value of $B is set to X'0000'. XREF 1 mode will be enforced and any previously existing relocatable file called FRED on unit 1 will be erased before generating the new relocatable file output.

RASSEMBL,L XY,1000,1,YZ,VERSION 1

This command assembles the source file XY on the currently selected unit generating a relocatable file with name YZ on unit 1 with label VERSION 1. The value of $B is set to X'1000' and LIST 3 is enforced throughout the relocatable assembly regardless of the occurrence of LIST statements in the relocatable assembler source.

RA,A XX.DISK1.DISK2.DISK3,177A

The source program is made up of three files all called YX on disks DISK1, DISK2 and DISK3. $B will be set to X'177A'. The A option ensures that the relocatable file is generated on the opposite unit to the remount unit as required.

RA,INQS PROG1

The source file PROG1 on the currently selected unit is

assembled with no relocatable file output. An errors-only listing will be written to the LIST file which must exist before starting the relocatable assembly. A cross reference listing will not be written to the LIST file.

RENAME -- Rename File

The RENAME utility is used to rename an existing file. The form of the call is as follows:

RENAME, options <iname> , <oname> , <olabel>
RE, options < iname > , < oname > , < olabel>

iname is the name of the file to be renamed. This is either an active file or an erased file if the E option is set. In the latter case, the file will be "unerased" as well a renamed.

<oname> , <olabel> are the new name and new label to be written to the directory. The file itself is unchanged. <oname> may not specify a unit number.

If the label is to be unchanged, then the third parameter may be omitted:

RENAME, options < iname > , < oname >
RE, options <iname> , <oname>

If the file name is to be unchanged, then the second parameter may be omitted:

RENAME, options <iname> , , < olabel >
RE, options < iname > , , <olabel>

If the E option is present then all but the first parameter may be omitted:

RENAME, options < iname >
RE, options < iname >

In this case the file is "unerased" and retains its original name and label.

In no case is it possible to use RENAME to generate two active files of the same name and type on one diskette unit. Thus the new name must be unique. Any attempt to violate this rule is considered an error.

The following letter options are accepted by RENAME:

D The file to be renamed is a data file.

E The file to be renamed is an erased file. If several erased files with the same name exist, then the first one is used.

O The file to be renamed is an object file.

R The file to be renamed is a relocatable file.

S The file to be renamed is a source file.

V All directory write operations are performed using the verify (reread check) option.

Exactly one of the options D, O, R or S must be present.

Examples:

RE,D Ø.DATA1,DATA2

The data file called DATA1 on unit zero is renamed DATA2. The label is unchanged.

RENAME,SE SRC

The erased source file SRC on the currently selected unit is unerased and marked as active. No other active source file of this name is present.

RENAME,OE OLD,NEW,RESCUED

The erased object file OLD on the currently selected unit is unerased and given the name NEW with label RESCUED.

## TMOVE -- Tape Move

The TMOVE utility is used to position a magnetic tape unit to a particular file. It is used on conjunction with the COPY utility to manage multiple files on a single tape. TMOVE also contains a

directory listing option. The form of the command to position to the start of a specified file is:

```
TMOVE <c >, < n >
TM < c >, <n>
```

< c >is the channel number of the tape controller which is a digit from 0-8.

<n> is the number of the file. 1 means the first file on the tape.

To backspace files from the current position, the following command is used:

```
TMOVE, B <c> , < n >
TM, B < c >,< n>
```

<n> in this case is the number of file backspace operations. Backspacing one file involves moving backwards past one file mark and continuing to move backwards until the start of tape or next file mark is reached. If a file mark was reached, the tape is spaced forward past this file mark.

To forward space files from the current position, the following command is used:

```
TMOVE, F <c> , < n >
TM, F <c> , <n>
```

<n> is the number of files to skip forward. Skipping one file involves moving forward till a file mark is found, and positioning just past

this file mark.

To list the initial record of every file on a tape, the following command is used:

```
TMOVE, L <c>
TM, L <c>
```

The initial record is typically a label record. This command acts to give directory listings. The tape is rewound following completion of the listing. It is assumed that the last file is terminated by a double end of file mark.

As specified above, the following letter options are recognized by TMOVE:

B Backspace files.

F Forward space files.

L List directory.

At most one of these options can be present

Examples:

```
TM 3, 4
```

The tape on channel 3 is spaced to file 4.

```
TMOVE, B Ø, 2
```

The tape on channel zero is backspaced two files.

TM, L 3

The directory (first record of each file) of the tape on channel 3 is listed.

UPDATE -- Update Source File

UPDATE is used to update a source file. The update data is taken from the current input device (cards, tape, file or keyboard) depending on the current mode. Unlike EDIT, UPDATE is thus suitable for batch mode file amendment. The form of the command is:

UPDATE, options < ifile > , < ofile >, < olabel>
UP, options < ifile > , < ofile> , <olabel>

<ifile > is the input source file to be updated.

<ofile > , < olabel > specify the name and label of the output file.

The following is a list of possible update commands:

$=C comment

C is a single character which will be treated as $ (the update command signal character) from now on. In this description of commands, $ stands for $ sign or any character substituted by a $=C command.

$*any desired comment text

This command may be used to insert comments into an update. The line will be ignored (except for printing if the L option is set).

$DONE comments

This command causes the remainder of the input file to be copied to the output file until an end-file is detected. If in merge mode (see $MERGE), the merge input file is copied to the output file and then the update continues from the original input file. Otherwise, control returns to the system.

$MERGE filename comments

The $MERGE command switches the input to a named file. The input file line number is reset to zero and the update continues with input from the merge file until $DONE or $QUIT is read.

$PAUSE comments

This command causes UPDATE to halt without changing the state of the input or output files. The command can be used to give instructions to the operator. Pressing the space bar releases the pause condition and continues the update.

$QUIT comments

This command is exactly like $DONE except that the copy to end of the input file is omitted.

$12,15 comments

This command copies lines (card images) numbered 12 through 15 from

the input file to the output file. The input file must be positioned before line 12 (at any position from the beginning of the file to "having read" line 11) or an error will be signalled. The comments must, as always, be preceded by at least one blank and will be ignored. The input file will be left positioned between records 15 and 16. This command is really the combination of an "omit" and a "copy" in that all lines from the current position of the input file through line 11 will be omitted, and lines 12 through 15 will be copied. If an end-file is detected during the omit, an error will be signalled.

$17 comments

This command is a special case of the copy command mentioned above. It will omit any lines from the current position of the input file through line 16 (no omit will occur if the input file is positioned between lines 16 and 17). It will then copy line 17 alone and return to reading the next update command. An alternate form for this command is "$17,17".

$,21 comments

This command is also a special case of the copy command mentioned above. It will copy from the current position of the input file (with no omit) through line 21. Note that an alternate form of the $DONE command mentioned above is "$,9999" where "9999" is really intended to mean some line number greater than or equal to the last line number of the current input file.

$&K comments

Subsequent occurrences of the character & will be changed to the character K. The command $&& resets the normal mode.

_label opcode variablefield comments

The underlining of the "1" above merely emphasizes that all source lines must not begin with a dollar sign ("$"). This line, if it appears in the update command stream, is copied onto the output file.

Commands where $ is followed by a key work may be abbreviated to the first letter (e.g., $P for $PAUSE). Source lines read from cards or tape are in standard ASCII code. The \ (reverse slash) may be used as a tab character as described for Paper Tape under the COPY utility.

The following letter options are recognized by UPDATE:

E Erase. Any existing source file of the same name as the output is erased before starting. Otherwise it is an error to have a duplicate name.

L Log update input lines to printer.

V Verify. All diskette output is written with the verify (reread check) option.

Examples:

```
UPDATE,L IN,OUT,NEW
NEW LINE
$,23
$26,28
$30
$DONE
```

The source file IN on the currently selected unit is updated and the output written to source file OUT with label NEW on the same unit. The new file consists of NEW LINE followed by lines 1-23, 26-28, 30-end from the old file. All update input is logged to the printer.

```
UP,E 1.X,1.X
```

Source file X on unit one is updated. The output replaces the old file with one of the same name, the old file being erased. The output label is blank.

VERIFY -- Verify File/Diskette Label

VERIFY is used in batch (cards, tape, file command input mode) to verify that a diskette has the expected DSN (i.e., that the correct diskette is loaded). The command is:

```
VERIFY <unit> , <dsn>
VE <unit>, <dsn>
```

<unit> is the unit to be checked.

<dsn> is the disk serial number expected.

VERIFY can also be used to verify that a file is present and has the expected label with the following command form:

```
VERIFY, option <file> , < label >
VE, option < file > ,< label >
```

<file> is the name of the file to be checked.

<label> is the label to be checked for.

VERIFY signals an error condition if the expected DSN or file label does not match the actual one.

The option in the file case is one of the following:

| | |
|---|---|
| D | File is a data file. |
| O | File is an object file. |
| R | File is a relocatable file. |
| S | File is a source file. |

Examples:

VE 1,DISKØ5

Verify that the diskette on unit one has serial number DISKØ5.

VERIFY,S 1.K,LM

Verify that source file K on unit one has label LM.

XDISK -- Examine Disk

XDISK (examine disk) is used online in keyboard mode to inspect and modify diskette data at the sector level. The command is:

XDISK,options
XD,options

The display includes the following:

UNIT-X Current unit, may be modified by entering U followed by a single digit.

TRACK-XX Current track, may be modified by entering T followed by two decimal digits.

SECTOR-XX Current sector, may be modified by entering S followed by two decimal digits.

POS-XX Cursor position in the sector data. May be modified by entering P followed by two hexadecimal digits or by entering R (right) to increment or L (left) to decrement. The latter two keys repeat automatically.

ERROR-X Status of read/write operation

Ø No error
1 Search check (sector not found)
2 Read error or verify error
3 Device inoperable
4 Track number out of range
5 Unit is write protected

Buffer Data 128 bytes of the diskette buffer sector data area displayed in hexadecimal and character form, or optionally (see F option), the full 256 byte disk buffer in which positions X'03-X'82' contain the sector data.

A sector is read onto the display by setting appropriate unit, track, and sector values. The buffer data as displayed may be modified by entering hexadecimal data from the keyboard.

W is keyed to set the buffer from the display and write the sector data to the currently specified sector. The alarm sounds if a unit error (status not zero) occurs.

H is keyed to obtain a hard copy replica of the current screen on the system printer.

X is keyed to cause return to the SPD/DOS nucleus.

Z is keyed to clear the sector data area of the buffer to hexadecimal zeroes.

The following letter options are recognized by XDISK:

F Full buffer. Normally the display includes only the 128 byte. sector data area. Specification of the F option causes the entire 256 byte diskette buffer to be displayed.

V Verify. Writes performed in response to a W command are performed with a reread check (verify).

Examples:

XDISK

Enter XDISK, writes will not be verified.

XD, VF

Enter XDISK, all writes will be verified. The display will include the full 256 bytes of the diskette buffer.

## ZAP -- Patch/Examine Object Program

ZAP is used to patch core images or object files for permanent correction or temporary debugging purposes. In the most general case, there are three sets of data involved:

(a) The input object file (first parameter) representing the program to be patched. In the case where ZAP is used after a manual boot, this parameter may be omitted in which case the L command may not be used.

(b) The current core image. This is a complete core image including the auto-exec locations. All patch modifications (M commands) modify the current core image. Following a manual boot (ZAP entered following a manual nucleus load), this core image is that which is saved during the load. Otherwise it is initially undefined until appropriate L commands are given. ZAP uses the scratch area (track two) of unit zero to store part of the core image. Thus ZAP itself is not part of this image.

(c) The output file (parameters two and three). This is used to store the resulting patched program in response to S commands. These parameters may be omitted (S commands not allowed) if the result is not to be saved or if modifications are to be written back to the input file (in-place modification).

There are four forms of the ZAP command:

ZAP, options < ifile> , < ofile > , <olabel>
ZA, options <file> , < ofile > , <olabel >

Object file< ifile >is to be loaded in using L commands, modified and then stored using S commands into a separate output file to be created called< ofile >with< olabel > as its label.

ZAP, options < ifile>
ZA, options <ifile >

Object file< ifile> is to be loaded in using L commands, modified and stored back in place using S commands.

ZAP, options
ZA, options

This form is used to inspect, modify and possibly restart the core image which was saved on a manual boot.

ZAP, options , < ofile >, <olabel>
ZA, options , < ofile> , <olabel>

This form is used when the saved core image is to be written to the specified output file. This is used if a program currently loaded (e.g., from the cassette library program) is to be saved on disk for future executions.

The following commands are accepted by ZAP. They are always taken from the current command source.

Aaaaa — Set address. The current address is set to aaaa (3 or 4 hexadecimal digits). The contents of the current address is continually displayed.

B — Place a breakpoint instruction at the current address.

Baaaa — Place a breakpoint instruction at the specified address. Proper operation of the breakpoint commands requires that the D&POWR routine be included in the program. See the SPD/DOS Programmer's Reference Manual (Order No. MS-7178) for further details.

Cssss — This command is used to specify a logical core size smaller than the actual physical size. This smaller size will be the one written to an output file (S command) or cassette tape (W command). The operand is 1FFF (for 8K) or 3FFF (for 16K) or 7FFF (for 32K). Also COOOO may be specified for a SIZE O program.

D <text> — The entire core image is dumped on the printer. <text> , if given, is used as a header. The blank separates D from < text >.

Daaaa. aaaa, aaaa. aaaa, . . . . aaaa. aaaa <text>
The specified regions of the core image (one or more as shown) are dumped on the printer. < text >, if given, is used as a header and must be separated from the last address by one blank.

E — End ZAP run. The output file (if any) is closed and control returns to the nucleus.

F — Get effective address. The contents of the current address is treated as a one-word instruction and its first level effective address is set as the current address. Command error is signalled if more than five levels of indirect addressing is encountered.

I — Image. The current core image is physically loaded into core but not started. ZAP terminates processing and the machine waits with the core image loaded. The machine waits in a disabled spin with the core image. The output file, if any, is closed before the wait occurs.

Jaaaa — Control is passed to location aaaa (3 or 4 hexadecimal digits), in the current core image. The output file, if any, is closed before the jump.

J — Control is passed to the program entry point as read from the input file. Valid only if an input file parameter was specified.

Jaaaa, options parameters
Pass control to aaaa with options and parameters made available to the program just as if a nucleus load command had been given (with Jaaaa the program name). Options and the comma may be omitted if no options are to be specified.

J, options parameters
As above, but control is passed to the program entry point read from the input file.

Lsn — The segment whose number is given (2 hexadecimal digits) is loaded into the current core image. If sn is omitted the main (or only) segment is loaded.

Mdddd, dddd, . . . dddd
The current location and following words modified to contain dddd (4 hexadecimal digits). The current location is then incremented to point past the last word specified.

N — Next. The current address is incremented by two (next word).

P — Previous. The current address is decremented by two (previous word).

Saaaa.aaaa,aaaa.aaaa,.....,aaaa.aaaa,eeee,sn
Store segment. The regions represented by aaaa.aaaa (one or more as shown), are stored on the output file as segment number sn or as the main segment if sn is omitted. eeee is the segment entry point. A segment may only be stored once.

Raaaa — Set relocation base. The address given (3-4 hexadecimal digits) is set as a relocatable base. Subsequently, any address appearing in a command may be given as Raaaa and the effect is to add in the specified relocation base. For example, R230 will set relocatable address 230. Both the current absolute and relocatable addresses are displayed. The Raaaa construction may also be used in Modify data, e.g., M8C00, R120 to set a jump to address 120 + relocation base.

Ssn — This abbreviated form of the S command is used if the limits and entry point are the same as for the input file. This form is the only allowed form when there is no output file (in-place modification), in which case it is permissable to store a segment more than once.

T — Test current core image for breakpoint condition. When ZAP is loaded with no input file specified, then an automatic T command is issued which will, if a breakpoint condition exists, cause the breakpoint location and the values of the registers to be displayed. Pressing the space bar clears this message. The T key may be used to redisplay the message at a later point.

Vdddd — Verify that the current location contains dddd (4 hexadecimal digits). Terminate ZAP with error if not.

Waaaa.aaaa,aaaa.aaaa,.....,aaaa.aaaa,eeee,c

This command is used on the 20/20 only to write the specified regions (up to 7) on the 20/20 cassette load tape in debug dump format suitable for boot loading. eeee is the program entry point. C is a digit giving number of copies to be written (from 1 to 7, normally 5 is used.)

X — Causes immediate return to the nucleus. The output file is not closed.

Z — Clears the entire core image area to hexadecimal zeros.

When the output file is closed (J or E commands) and if there is an input file specified, then all segments of the input file which have not been stored by S commands are copied unchanged to the output file. In the case of the J command, subsequent segment loads will be from the new output file.

Note that when an S command is given with no address ranges, for example in the case of an in place update, only those areas originally in the program are stored. In particular, the following areas will not be stored:

BSS areas

Areas skipped by ORG commands

Bytes skipped by implicit or explicit work alignment

Areas outside the bounds of the program

Attempts to modify such areas and then store the result do not work and give no error indication. Always use the full form of the Store

command if any such areas are to be modified.

ZAP uses a scratch area (track two) on unit zero to store the current core image, thus unit zero must not be write protected. Neither the utility programs nor the nucleus disturb this image (except on a manual boot). Thus if ZAP errors out, it may be restarted without losing the core image being worked on although segments already stored with S commands may be lost.

The following letter options are recognized by ZAP:

A — Alternate work file. This causes unit one to be used for the work file area on track two instead of unit zero.

E — Erase. An object file with the same name as the output file will be erased before starting, otherwise this is an error.

V — Verify. Perform all disk writes with verify (reread check) option.

Examples:

```
ZAP GAME
L
A3ØØ
M6161
S
E
```

The object file GAME on the currently selected unit is patched in place by modifying location X'Ø3ØØ' of the segment zero (the main segment) to contain X'6161'. Other (overlay) segments are unaffected.

```
(load program using cassette library utility)
(manual boot)
ZA ,SLOTS,GAME
S 1ØØ.FD3,FFC.FFD,4ØØ
```

A program loaded before a manual boot is stored as an executable object file called SLOTS on the currently selected unit with label GAME. The limits are X'Ø1ØØ' - X'ØFD3' and X'ØFFC' - X'ØFFD' with entry point X'Ø4ØØ'.

```
ZAP,E 1.X,0.Y,NEW
L1B
A1ØØ
M3636
S1D
L1D
A3ØØ
M46B
S1D
E
```

Segments 27 and 29 of object program X on unit 1 are modified as shown. The result is written to a new object file Y on unit zero with label NEW. All other segments are copied unchanged.

```
ZAP
D DUMP AFTER BOMB, 07 JULY
X
```

The current core image is dumped to the printer with the specified page title heading.

# APPENDIX A

## ERROR CODES

Error codes consist of two letters followed by two decimal digits. This section contains an alphabetical list of all error codes issued by the SPD/DOS nucleus and its utility programs.

Certain general types of errors in diskette input/output operations occur as follows:

| | |
|---|---|
| Search check | A sector cannot be found. Either the diskette is improperly formatted, the heads are "seeked" to the wrong track, the prior sector had a cyclic check error, no diskette is mounted, the drive's door is open, or the hardware is malfunctioning. |
| Read check | A cyclic check comparison repreatedly failed after several attempts at reading the data. This condition indicates that the data was recorded incorrectly. |
| Write check | A cyclic check comparison failed during a write operation. The error may be in the previous sector (indicating incorrectly recorded data) or in the sector being written if the verify option is used (indicating a damaged disk surface). |
| Unit inoperable | The diskette unit is inoperable. This condition is most usually caused by reloading the unit at a time when a reload was not permitted (e.g., reloading the source file diskette unit in the middle of an assembly. |

| | |
|---|---|
| Write protect | A write was attempted to a diskette which was write protected (write tab attached or write protect button depressed). |
| Segment load error | In an overlay program, an attempt to load an overlay segment failed due to a diskette input error. If this happens, the object program file should be abandoned and a backup copy used. |

## ASSEMBLE Error Codes

| | |
|---|---|
| ASØ1 | Error opening source file. Search check |
| ASØ2 | Error opening source file. Read check. |
| ASØ3 | Error opening source file. Unit inoperable. |
| ASØ4 | Error opening source file. File not found. |
| ASØ5 | Error creating object file. Search check. |
| ASØ6 | Error creating object file. Read check. |
| ASØ7 | Error creating object file. Unit inoperable. |
| ASØ8 | Error creating object file. Duplicate file name. |
| ASØ9 | Error creating object file. Write protect. |
| AS1Ø | Error creating object file. File area full. |
| AS11 | Error opening work file. Search check. |
| AS12 | Error opening work file. Read check. |
| AS13 | Error opening work file. Unit inoperable. |
| AS14 | Error opening work file. File not found. |
| AS15 | Work file size less than 6 tracks. |
| AS17 | Error reading source file. Search check. |
| AS18 | Error reading source file. Read check. |
| AS19 | Error reading source file. Unit inoperable. |
| AS2Ø | Error reading source file. Missing end of file. |
| AS21 | Error writing object file. Search check. |
| AS22 | Error writing object file. Write check. |

AS23 Error writing object file. Unit inoperable.

AS24 Error writing object file. File area exhausted.

AS25 Error writing object file. Write protect.

AS26 Error reading work file. Search check.

AS27 Error reading work file. Read check.

AS28 Error reading work file. Unit inoperable.

AS29 Error writing work file. Search check.

AS30 Error writing work file. Write check.

AS31 Error writing work file. Unit inoperable.

AS33 Error writing work file. Write protect.

AS34 Segment load error.

AS35 Parameter format error.

AS37 Error opening list file. Search check.

AS38 Error opening list file. Read check.

AS39 Error opening list file. Unit inoperable.

AS40 Error opening list file. File not found.

AS41 Error opening list file. Write protect.

AS43 Error writing list file. Search check.

AS44 Error writing list file. Write check.

AS45 Error writing list file. Unit inoperable.

AS47 Error writing list file. Write protect.

AS48 Incorrect file allocation for diskette reload.

AS49 Erroneous DOS option specified.

AS50 Wrong Release of DOS in use.

CNFG Error Codes

| | |
|---|---|
| CNØ1 | Attempted use in other than keyboard mode. |
| CNØ2 | Error reading label record. Search check. |
| CNØ3 | Error reading label record. Read check. |
| CNØ4 | Error reading label record. Unit inoperable. |
| CNØ5 | Error writing label record. Search check. |
| CNØ6 | Error writing label record. Write check. |
| CNØ7 | Error writing label record. Unit inoperable. |
| CNØ8 | Error writing label record. Write protect. |
| CN09 | Wrong release of DOS in use. |

COPY Error Codes

| | |
|---|---|
| CO∅1 | Error opening input file. Search check. |
| CO∅2 | Error opening input file. Read check. |
| CO∅3 | Error opening input file. Unit inoperable. |
| CO∅4 | Error opening input file. File not found. |
| CO∅5 | Error creating output file. Search check. |
| CO∅6 | Error creating output file. Read check. |
| CO∅7 | Error creating output file. Unit inoperable. |
| CO∅8 | Error creating output file. Duplicate file name. |
| CO∅9 | Error creating output file. Write protect. |
| CO1∅ | Error creating output file. File area full. |
| CO11 | Error reading input file. Search check. |
| CO12 | Error reading input file. Read check. |
| CO13 | Error reading input file. Unit inoperable. |
| CO14 | Error reading input file. Missing end of file. |
| CO15 | Error writing output file. Search check. |
| CO16 | Error writing output file. Write check. |
| CO17 | Error writing output file. Unit inoperable. |
| CO18 | Error writing output file. File area exhausted. |
| CO19 | Error writing output file. Write protect. |
| CO2∅ | Error reading external medium. |
| CO21 | Error writing external medium. |

CO22 BOOT mode program too long to write cassette (512 bytes).

CO23 Missing D, O, R, or S option or more than one option.

CO24 Parameter format error.

CO25 Segment load error.

CO26 Verify error.

## CREATE Error Codes

| | |
|---|---|
| CRØ1 | Error creating file. Search check. |
| CRØ2 | Error creating file. Read check. |
| CRØ3 | Error creating file. Unit inoperable. |
| CRØ4 | Error creating file. Duplicate file name. |
| CRØ5 | Error creating file. Write protect. |
| CRØ6 | Error creating file. File area full. |
| CRØ7 | Error initializing file. Search check. |
| CRØ8 | Error initializing file. Write check. |
| CRØ9 | Error initializing file. Unit inoperable. |
| CR1Ø | Error initializing file. File area exhausted. |
| CR11 | Error initializing file. Write protect. |
| CR12 | Parameter format error. |

## DCOPY Error Codes

| | |
|---|---|
| DCØ1 | Error opening input file. Search check. |
| DCØ2 | Error opening input file. Read check. |
| DCØ3 | Error opening input file. Unit inoperable. |
| DCØ4 | Error opening input file. File not found. |
| DCØ5 | Error creating output file. Search check. |
| DCØ6 | Error creating output file. Read check. |
| DCØ7 | Error creating output file. Unit inoperable. |
| DCØ8 | Error creating output file. Duplicate file name. |
| DCØ9 | Error creating output file. Write protect. |
| DC1Ø | Error creating output file. File area full. |
| DC11 | Error reading input file. Search check. |
| DC12 | Error reading input file. Read check. |
| DC13 | Error reading input file. Unit inoperable. |
| DC14 | Error reading input file. Missing end of file. |
| DC15 | Error writing output file. Search check. |
| DC16 | Error writing output file. Write check. |
| DC17 | Error writing output file. Unit inoperable. |
| DC18 | Error writing output file. File area exhausted. |
| DC19 | Error writing output file. Write protect. |
| DC23 | Missing D, O, R or S option or more than one option. |
| DC24 | Parameter format error. |
| DC25 | Segment load error. |

EDIT Error Codes

| | |
|---|---|
| ED01 | Error opening input file. Search check. |
| ED02 | Error opening input file. Read check. |
| ED03 | Error opening input file. Unit inoperable. |
| ED04 | Error opening input file. File not found. |
| ED05 | Error creating output file. Search check. |
| ED06 | Error creating output file. Read check. |
| ED07 | Error creating output file. Unit inoperable. |
| ED08 | Error creating output file. Duplicate file name. |
| ED09 | Error creating output file. Write protect. |
| ED10 | Error creating output file. File area full. |
| ED11 | Error writing output file. Search check. |
| ED12 | Error writing output file. Write check. |
| ED13 | Error writing output file. Unit inoperable. |
| ED14 | Error writing output file. File area exhausted. |
| ED15 | Error writing output file. Write protect. |
| ED16 | Segment load error. |
| ED17 | Parameter format error. |

ERASE Error Codes

| | |
|---|---|
| ERØ1 | Error reading directory. Search check. |
| ERØ2 | Error reading directory. Read check. |
| ERØ3 | Error reading directory. Unit inoperable. |
| ERØ4 | Error writing directory. Search check. |
| ERØ5 | Error writing directory. Write check. |
| ERØ6 | Error writing directory. Unit inoperable. |
| ERØ7 | Error writing directory. Write protect. |
| ERØ8 | Specified file not found and I option not set. |
| ERØ9 | Missing D, O, R, or S option. |
| ER1Ø | Parameter format error. |

## FORMAT Error Codes

| | |
|---|---|
| FOØ1 | Error formatting diskette. Search check. |
| FOØ2 | Error formatting diskette. Verify check. |
| FOØ3 | Error formatting diskette. Unit inoperable. |
| FOØ4 | Error formatting diskette. Write protect. |
| FOØ5 | Error writing diskette. Search check. |
| FOØ6 | Error writing diskette. Write check. |
| FOØ7 | Error writing diskette. Unit inoperable. |
| FOØ8 | Error writing diskette. Write protect. |
| FOØ9 | Parameter format error. |
| FO1Ø | Segment load error. |
| FO11 | Error reading label record. Search check. |
| FO12 | Error reading label record. Read check. |
| FO13 | Error reading label record. Unit inoperable. |

LIST Error Codes

| | |
|---|---|
| LIØ1 | Error opening file. Search check. |
| LIØ2 | Error opening file. Read check. |
| LIØ3 | Error opening file. Unit inoperable. |
| LIØ4 | Error opening file. File not found. |
| LIØ5 | Error reading file. Search check. |
| LIØ6 | Error reading file. Read check. |
| LIØ7 | Error reading file. Unit inoperable. |
| LIØ8 | Error reading file. Missing end of file. |
| LIØ9 | Missing D, O, R, or S option |
| LI1Ø | Parameter format error. |

## NUCLEUS Error Codes

This section contains codes generated from the system nucleus. These codes are prefixed by the letters NU.

NUØ1 Error reading directory. Search check.

NUØ2 Error reading directory. Read check.

NUØ3 Error reading directory. Unit inoperable.

NUØ4 Error reading label sector. Search check.

NUØ5 Error reading label sector. Read check.

NUØ6 Error reading label sector. Unit inoperable.

NUØ7 Error opening object file. Search check.

NUØ8 Error opening object file. Verify check.

NUØ9 Error opening object file. Unit inoperable.

NU1Ø Error opening object file. File not found.

NU11 Error loading program. Search check.

NU12 Error loading program. Read check.

NU13 Error loading program. Unit inoperable.

NU14 Error loading program. Improper object file format.

NU15 Error loading program. Wrong system or boot mode program.

NU16 Error loading program. Insufficient memory on 20/20 or 10/24.

NU17 Error opening command file. Search check.

| | | |
|---|---|---|
| NU18 | Error opening command file. | Read check. |
| NU19 | Error opening command file. | Unit inoperable. |
| NU20 | Error opening command file. | File not found. |
| NU21 | Error reading command file. | Search check. |
| NU22 | Error reading command file. | Read check. |
| NU23 | Error reading command file. | Unit inoperable. |
| NU24 | Error reading command file. | Improper format. |
| NU25 | Card reader not configured. | |
| NU26 | Tape reader not configured. | |
| NU27 | Command format error. | |
| NU28 | Error writing label sector. | Search check. |
| NU29 | Error writing label sector. | Write check. |
| NU3Ø | Error writing label sector. | Unit inoperable. |
| NU32 | Error writing label sector. | Write protect. |

PACK Error Codes

| | |
|---|---|
| PAØ1 | Error reading diskette. Search check. |
| PAØ2 | Error reading diskette. Read check. |
| PAØ3 | Error reading diskette. Unit inoperable. |
| PAØ4 | Error writing diskette. Search check. |
| PAØ5 | Error writing diskette. Write check. |
| PAØ6 | Error writing diskette. Unit inoperable. |
| PAØ8 | Error writing diskette. Write protect. |
| PAØ9 | Parameter format error. |

## RASSEMBL Error Codes

| | |
|---|---|
| RAØ1 | Error opening source file. Search check. |
| RAØ2 | Error opening source file. Read check. |
| RAØ3 | Error opening source file. Unit inoperable. |
| RAØ4 | Error opening source file. File not found. |
| RAØ5 | Error creating relocatable file. Search check. |
| RAØ6 | Error creating relocatable file. Read check. |
| RAØ7 | Error creating relocatable file. Unit inoperable. |
| RAØ8 | Error creating relocatable file. Duplicate file name. |
| RAØ9 | Error creating relocatable file. Write protect. |
| RA1Ø | Error creating relocatable file. File area full. |
| RA11 | Error opening work file. Search check. |
| RA12 | Error opening work file. Read check. |
| RA13 | Error opening work file. Unit inoperable. |
| RA14 | Error opening work file. File not found. |
| RA15 | Work file size less than 6 tracks. |
| RA17 | Error reading source file. Search check. |
| RA18 | Error reading source file. Read check. |
| RA19 | Error reading source file. Unit inoperable. |
| RA20 | Error reading source file. Missing end of file. |
| RA21 | Error writing relocatable file. Search check. |
| RA22 | Error writing relocatable file. Write check. |

RA23 Error writing relocatable file. Unit inoperable.

RA24 Error writing relocatable file. File area exhausted.

RA25 Error writing relocatable file. Write protect.

RA26 Error reading work file. Search check

RA27 Error reading work file. Read check.

RA28 Error reading work file. Unit inoperable.

RA29 Error writing work file. Search check.

RA30 Error writing work file. Write check.

RA31 Error writing work file. Unit inoperable.

RA33 Error writing work file. Write protect.

RA34 Segment load error.

RA35 Parameter format error.

RA37 Error opening list file. Search check.

RA38 Error opening list file. Read check.

RA39 Error opening list file. Unit inoperable.

RA40 Error opening list file. File not found.

RA41 Error opening list file. Write protect.

RA43 Error writing list file. Search check.

RA44 Error writing list file. Write check.

RA45 Error writing list file. Unit inoperable.

RA47 Error writing list file. Write protect.

RA48 Incorrect file allocation for diskette reload.

RA49 Erroneous DOS option specification.

RA50 Inproper release of DOS in use.

RENAME Error Codes

| | |
|---|---|
| RE01 | Error reading directory. Search check. |
| RE02 | Error reading directory. Read check. |
| RE03 | Error reading directory. Unit inoperable. |
| RE04 | Specified file not found. |
| RE05 | Error writing directory. Search check. |
| RE06 | Error writing directory. Write check. |
| RE07 | Error writing directory. Unit inoperable. |
| RE09 | Error writing directory. Write protect. |
| RE10 | New file name not unique. |
| RE11 | Missing or duplicate option |
| RE12 | Parameter format error |

TMOVE Error Codes

| | |
|---|---|
| TM01 | Tape unit is offline or formatter turned off. |
| TM02 | Attempt to backspace past BOT. |
| TM03 | Attempt to position past double end of file. |
| TM04 | Attempt to position past EOT. |
| TM05 | Read error during list option. |
| TM06 | Parameter format error. |

UPDATE Error Codes

UPØ1 Error opening input or merge file. Search check.

UPØ2 Error opening input or merge file. Read check.

UPØ3 Error opening input or merge file. Unit inoperable.

UPØ4 Error opening input or merge file. File not found.

UPØ5 Error creating output file. Search check.

UPØ6 Error creating output file. Read check.

UPØ7 Error creating output file. Unit inoperable.

UPØ8 Error creating output file. Duplicate file name.

UPØ9 Error creating output file. Write protect.

UP1Ø Error creating output file. File area full.

UP11 Error reading input or merge file. Search check.

UP12 Error reading input or merge file. Read check.

UP13 Error reading input or merge file. Unit inoperable.

UP14 Error reading input or merge file. Missing end of file.

UP15 Error writing output file. Search check.

UP16 Error writing output file. Write check.

UP17 Error writing output file. Unit inoperable.

UP18 Error writing output file. File area exhausted.

UP19 Error writing output file. Write protect.

UP2Ø Error reading command file. Search check.

UP21 Error reading command file. Read check.

UP22 Error reading command file. Unit inoperable.

UP23 Error reading command file. Improper source format.

UP24 End of file on command input.

UP25 Update command error. Unrecognized type.

UP26 Update command error. Syntax error.

UP27 Update command error. Line number error.

UP28 Parameter format error.

UP29 Segment load error.

VERIFY Error Codes

| | |
|---|---|
| VEØ1 | Error reading. Search check. |
| VEØ2 | Error reading. Read check. |
| VEØ3 | Error reading. Unit inoperable. |
| VEØ4 | Specified file not found. |
| VEØ5 | File label match verify error. |
| VEØ6 | Diskette serial number match verify error. |
| VEØ7 | Parameter format error. |

ZAP Error Codes

| | |
|---|---|
| ZAØ1 | Error opening input file. Search check. |
| ZAØ2 | Error opening input file. Read check. |
| ZAØ3 | Error opening input file. Unit inoperable. |
| ZAØ4 | Error opening input file. File not found. |
| ZAØ5 | Error creating output file. Search check. |
| ZAØ6 | Error creating output file. Read check. |
| ZAØ7 | Error creating output file. Unit inoperable. |
| ZAØ8 | Error creating output file. Duplicate file name. |
| ZAØ9 | Error creating output file. Write protect. |
| ZA1Ø | Error creating output file. File area full. |
| ZA11 | Error reading input file. Search check. |
| ZA12 | Error reading input file. Read check. |
| ZA13 | Error reading input file. Unit inoperable. |
| ZA14 | Error writing output file. Search check. |
| ZA15 | Error writing output file. Write check. |
| ZA16 | Error writing output file. Unit inoperable. |
| ZA17 | Error writing output file. File area exhausted. |
| ZA18 | Error writing output file. Write protect. |
| ZA19 | Error reading command file. Search check. |
| ZA20 | Error reading command file. Read check. |
| ZA21 | Error reading command file. Unit inoperable. |

ZA22 Error reading command file. Missing end of file.

ZA23 Error reading core image work file. Search check.

ZA24 Error reading core image work file. Read check.

ZA25 Error reading core image work file. Unit inoperable.

ZA26 Error writing core image work file. Search check.

ZA27 Error writing core image work file. Write check.

ZA28 Error writing core image work file. Unit inoperable.

ZA29 Error writing core image work file. Write protect.

ZA3Ø End of command file encountered.

ZA31 Parameter format error.

ZA32 ZAP command format error.

ZA33 L command invalid. (No input file.)

ZA34 S command invalid. (No output file, or segment already stored.)

ZA36 Wrong system (wrong model TPU, or external memory present, or BOOT mode program).

ZA37 Segment load error.

ZA38 ZAP V (Verify) reject due to data mismatch.

ZA39 Error writing load cassette.

# APPENDIX B

## FORM OF DISTRIBUTION

The SPD/DOS system is distributed as a single diskette with serial number SPDDOS. The following files are present on this diskette.

| | |
|---|---|
| ASSEMBLE | Object file for ASSEMBLE utility. |
| CNFG | Object file for CNFG utility. |
| COPY | Object file for COPY utility. |
| CREATE | Object file for CREATE utility. |
| EDIT | Object file for EDIT utility. |
| ERASE | Object file for ERASE utility. |
| FORMAT | Object file for FORMAT utility. |
| LIST | Object file for LIST utility. |
| PACK | Object file for PACK utility. |
| RASSEMBL | Object file for RASSEMBL utility. |
| RENAME | Object file for RENAME utility. |
| TMOVE | Object file for TMOVE utility. |
| UPDATE | Object file for UPDATE utility . |
| VERIFY | Object file for VERIFY utility. |
| XDISK | Object file for XDISK utility. |
| ZAP | Object file for ZAP utility. |

| | |
|---|---|
| DOSLIB | Source code for SPD/DOS library routines. |
| DOSLIB | Relocatable library of SPD/DOS routines. |

The configuration parameters are set to an unspecified state. They should be modified to match the actual configuration on receipt of the distribution disk by use of CNFG.

Copies of this disk can conveniently be made using the PACK utility.

INCOTERM®
CORPORATION
6 STRATHMORE ROAD
NATICK, MASS. 01760
(617) 655-6100

PRINTED IN U.S.A.