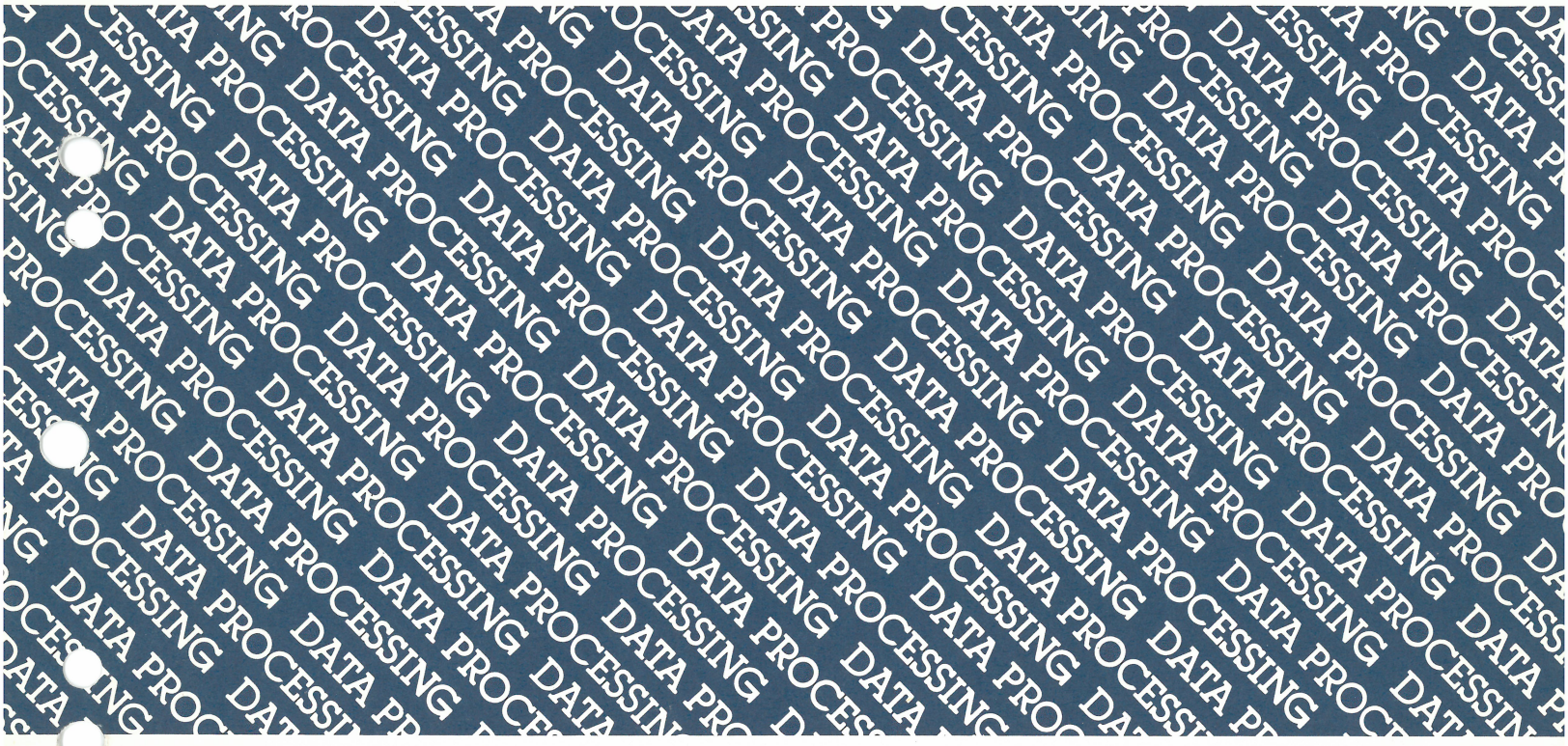
CPT®



CP/M®

COMPUPAK CP/MØ OPERATOR'S MANUAL

June 16, 1980

CPT Corporation
8100 Mitchell Road
P.O. Box 295
Minneapolis, MN 55440
U.S.A.

(612) 937-8000 Telex 29-0775

Marketing Identifier: 6619
Price: $10.

Disclaimer

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

Table of Contents

1.0 INTRODUCTION.

This manual describes the concept and features of the Control Program Monitor (CP/M®) disk operating system. The disk operating system is used to control the job execution of the CPT 8000 when it is functioning as a computer. It may be viewed as the superviser of the computer. CP/M is a product of Digital Research Corporation. It has become one of the few industry-wide standard operating systems ever developed. Utilization of this operating system makes possible the exchange, with little or no modification, of applications programs with owners of many brands of computer equipment. This also means that there are many software suppliers that can produce programs for the CPT 8000 and many other brands of equipment at the same time. Thus, software developmental costs can be spread over many more users than ever before possible. Therefore, lower cost software will be available to the users of CPT 8000 hardware.

The disk operating system resides on the very beginning of the CompuPak CP/M disk (track 0 and track 1 of a floppy disk). The system is copied into the computer's memory when the STOP key is depressed. It is also copied into memory immediately after the CompuPak interface disk is replaced with the CompuPak CP/M disk. The system resides in high memory at all times. The disk operating system controls the disk drives and all peripheral devices such as CPT screen, printer, keyboard, etc. It also has several built in commands which allow the user to perform various disk management functions.

The user should become totally familiar with this manual before attempting to operate application programs. It is easy to become confused when operating a computer if the user is not comfortable with being able to control and understand each function the computer is performing. Extreme care should be exercised with regard to maintaining quality control over the data being entered, the handling of files and disks and, the operating procedure being utilized. This manual will attempt to advise the user of some of the procedures to follow that will save many painful hours of computer reruns due to careless quality control.

The disk operating system utilizes unique file naming conventions, has built in operator commands and manages user programs and files. Several utility programs are also furnished with the system which may be used for copying files, managing files and disks, and printing files on the printer. This manual covers these functions as well as describes the architecture of the disk operating system.

This manual is divided into three primary sections. The Overview, How To, and Detail Functional Specifications. The user may want to get started without reading the entire manual. The "How To" section of this manual is designed to allow the user to perform the most common procedures without reading and understanding the entire manual. The user may copy the command lines presented in the "How To" section without understanding them. Caution is advised here, for it is far better to fully understand the procedure than to perform it totally mechanically.

CP/M Control Program Monitor is a registered trademark of Digital Research

2.0 OVERVIEW

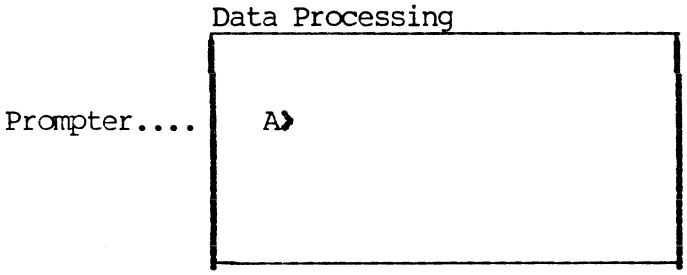
The user communicates with the computer by utilizing the Console Command Processor (CCP) of the disk operating system. The CCP acts as the supervisor of the computer. The built in commands reside in the CCP portion of the disk operating system. When the computer has finished copying the disk operating system into its memory, control is transfered to the CCP. When the CCP is ready it will give the operator a system prompter. The entire disk operating system requires about 16,000 characters (Bytes) of memory. The CPT 8000 has a total of 64,000 characters of memory. This means that 48,000 characters of memory are available for applications programs and data. This area of memory is known as the Transient Program Area (TPA). The TPA is used by the computer for storing and executing utility and application programs. The TPA is also used to store data.

The user will know when the CCP has control of the computer by observing the prompter symbol ">". The "greater than" symbol is preceded by a letter code. The letter code identifies the disk drive that is currently the primary reference drive. Disk drive devices are designated by a letter code "A or B". The CCP is ready to accept a console command when the greater than prompter is presented to the user.

The user may determine which version and modification level of CompuPak is in control of the CPT 8000 by depressing the CODE key and questionmark key simultaneously.

All commands given to the CCP are executed when the user depresses the RETURN key (labled "RETURN" on the key board).

In summary, the CCP signals it is ready to accept a console command by presenting the current disk letter followed by a greater than symbol.



The user may at this time execute either a built in command or a library command. The built in commands are part of the CCP which resides in high memory at all times. The library commands (refered to as transient commands by Digital Research) are copied from the files portion of the disk to the Transient Program Area (TPA) of the computers memory whenever the user executes a library command. The files portion of the disk contains the program library as well as user data files.

2.1 FILE NAMING CONVENTION

All files that reside in the files portion of the disk are identified by a file name. This name is place in the disk directory and is used by the disk operating system to locate the information that is stored on the disk. The file reference consists of two parts: The file name and the file type. The file type is optional.

A period is used to separate the file name from the file type. The file name may be from one to eight characters. The file type may be from one to three characters (when present). The file name and type cannot contain any of the following special characters:

< > . , ; : = ? *

Lower case letters may be used for the file name or file type, however; the CCP will convert them to upper case before searching or entering new file names into the directory.

Some file types have special meanings. These types have been assigned either by the CCP; by compiler programming languages such as BASIC, COBOL, FORTRAN, PASCAL; or by applications programs. Some of the most common types assigned are:

COM	Recognized by the CCP as a library program command. The CCP will automatically supply ".COM" for the user when the library command name is typed immediately following the prompter.
BAS	Recognized by interpretive BASIC programming languages as a program file and by compiler BASIC programming languages as a source program file.
INT	Used by intermediate interpretive BASIC programming languages as the program file. Must be used with a run time module such as CRUN2.COM
SUB	Used by the SUBMIT.COM library command program as a source of CCP commands. These commands are automatically executed by the SUBMIT library command program.
DAT	Recognized by CPT business application programs as a data file.
DF	Recognized by many CP/M application programs as a data file.
FIL	Same as above.
ASM	Recognized by assembly language programs as a source program file.

BAK — Used by text editors and some application programs as a back up file. The original file is renamed to a file type BAK so that if anything goes wrong the user may recover by using the BAK file.

$$$ — A temporary file established by utility programs. This file type is usually automatically erased by the utility program upon successful completion of its task. The user may find a file like this on the directory if the utility is interrupted before it can finish its work.

2.2 BUILT IN COMMANDS

Built in commands are located within the CCP. The built in commands are:

DIR — Display the file names located on the disk directory. Similar to CODE IN for word processing.

TYPE — Display the contents of a file on the console CRT. The display may be alternately stopped and started by depressing the CLEAR TAB key. The display may be terminated before the end of file is encountered by depressing any key twice. Note: It is not possible to overflow the screen buffer in data processing due to the fact that the screen scrolls and therefore only stores one line in computer memory at a time.

ERA — Remove the specified file or files from the disk directory.

REN — Rename the specified file from what currently is shown on the directory to the specified name.

d: — Change the current disk from whatever the letter code is in front of the greater than prompter to the letter code specified by d: (d: may either be equal to A: or B:).

EXAMPLES OF THE USE OF THE BUILT IN COMMANDS FOLLOW:

Data Processing

Display directory on drive A	A>DIR
May also be entered as .	A>DIR A:
Display directory on drive B	A>DIR B:
Display contents of a file named GLACCNT.DAT..	A>TYPE GLACCNT.DAT
Erase the file LIST.COM	A>ERA LIST.COM
Rename the file GLINFO.BAK to GLINFO.DAT	chg to chg from A>REN GLINFO.DAT=GLINFO.BAK
Switch current drive from A to B	A>B: B>
Switch current drive from B to A	B>A: A>

2.3 FILE CONTROL CHARACTERS

Many of the CCP built in commands and library program commands are designed to use two special file control characters. These characters are very useful when the user would like to handle similar files in a like manner. For instance the user may desire to copy all of the files from one disk to another. This would be a very laborous process if the user had to execute the copy program for each program transferred.

When using these characters the user may substitute the words "I don't care" wherever the special character appears in the file name and file type. The two characters to be used are:

- `*` This character may be substituted for either a file name or file type.
- `?` This character may be substituted for a single character within the file name or file type. The user may use from one "?" to eight "???????" in the file name and from one "?" to three "???" in the file type.

EXAMPLES OF THE USE OF THE FILE CONTROL CHARACTERS

Data Processing

Display all files that are file type .COM	A>DIR *.COM
Display all files beginning with G that are file type .INT	A>DIR G???????.INT
Note: The above examples all use drive A because the drive was not specified and the current drive is A	A>
Display all GL files on drive B	A>DIR B:GL??????.*
Erase all BAS source beginning with GL and ending with SRT	A>ERA GL??SRT.BAS

Note: File control characters may not be used with TYPE or REN Commands.

2.4 LINE EDITING, SCREEN, KEYBOARD AND PRINTER CONTROL

The CCP allows certain line editing functions while typing command lines. Other functions have been assigned to match some of the most common keys provided on a standard data processing terminal. The keys marked by an asterisk are used by the CCP. All other keys will function only when the application programs utilize them. These functions may be invoked either by depressing the key marked "CODE" and the specified key simultaneously or by depressing the specified function key. The operator should be aware that the labeled function keys normally used for word processing such as WORD, LINE, PARA, etc. have been changed to simulate control functions of a normal data processing terminal.

FUNCTION KEY	CODE KEY	DESCRIPTION
ADJ	N	Used as an "insert line" in some text editors. Not used when CCP is in control.
BACKSPACE	H	Causes the cursor to back up one space and erase the previously typed character in Microsoft Basic.
CHAR	G	Rings bell if RETURN key is depressed after CHAR key. It is also a delete character on some text editors.
*CLEAR TAB	S	This key will cause the display on the CRT or PIP program printouts to stop or freeze momentarily. Program execution and output to the CRT or printer will continue when any character is depressed on the keyboard. This key is active when the CCP or the PIP program is in control of the computer.
*DOWN	J	This causes the cursor to go down vertically on the screen. It performs the function of a line feed.
*ERASE	-	This key corresponds to the RUBOUT key on a standard data processing terminal. This key performs the function of backspace (erase left). The operator should notice, however, that this key may function differently under the control of some application programs.
HAT	H	Same as backspace key.
*HOLD	-	Capital Lock. This key capitalizes the letters "a" through "z", all other keys remain in lower case mode. This key simulates CAPS LOCK on a standard computer terminal. It is a toggle type key, that is, to turn of CAPS LOCK depress the key a second time.

IN	J	Used as a down arrow key on some text editors. Not used when CCP is in control (see down key).
INSERT	F	Used as a character insert command by some text editors. Not used when CCP is in control.
LINE	Y	Used as a command to delete a line by some text editors. Not used when CCP is in control.
MAR REL	Z	Used as end of video entry by some text editors. Also used as end of entry by the PIP program. Can be used as a clear screen command by depressing MAR REL then RETURN.
MAR SET	-	No response, not implemented.
MOVE	L	This key will cause the printer to form feed if the PRINT key has been set to cause the printer to follow the CRT. Depress MOVE followed by a RETURN.
OUT	-	Same as the erase key.
*PAGE	C	Same as stop key.
*PARA	R	Retype current command line when CCP is in control. This key will also cause machine to reverse page (scroll backwards) in some text editors.
*PRINT	P	Copy all subsequent console (CRT) output to the currently assigned list device (usually the printer). Output is sent to both the printer and the CRT console device until the PRINT key is depressed a second time (or when STOP key is depressed).
PROG	-	No response, not implemented.
*RETURN	M	Carriage return.
SKIP	;	Corresponds to the ESCape key on many terminals.
*STOP	C	Disk operating system reload (sometimes referred to as a warm boot, or restart). Can be used to return to the CCP from an application program when the program has paused to receive console input. This key should always be depressed immediately after changing diskettes. If this is not done a BDOS error will occur. If a BDOS error occurs, depress the STOP key at that time. Then resume normal operation.
TAB	I	Will cause a tab of 8 spaces in some applications.

UP	K	Will cause the cursor to move up the screen vertically in some text editors.

*These keys are utilized when the CCP is in control of the computer. This condition is signaled to the operator by the greater than prompter preceded by the active drive designator. For example

A>

The Key to the right of the letter "P" key is called the "Pa" key. It has the "!" character on its surface. In CompuPak CP/M this key represents the following characters:

Pa KEY	Backslash.	(Prints on screen \ )
SHIFT Pa KEY	Up-side-down v.	(Prints on screen ʌ)
CODE Pa KEY	Accent.	(Prints on screen `)
CODE SHIFT Pa	Underline.	(Prints on screen _)

Note: CCP command lines can generally be up to 255 characters in length; they are not acted upon until RETURN is depressed. All keys of the keyboard repeat when held down.

Remember ... The CPT 8000 has the ability to store keystrokes. When this feature is coupled with the fact that all keys will go into auto-repeat after being held down for approximately 2 seconds, confusing results may be experienced by the operator. The operator may experience machine operations that did not appear to be initiated by his actions. To avoid this situation, always depress keys with a quick firm positive action - DO NOT LINGER ON THE KEYBOARD!

2.5 LIBRARY COMMANDS

Library commands are loaded from the current disk (unless otherwise specified) and executed in the TPA. The library commands furnished with the disk operating system that may be executed under the CCP are shown below. Additional commands may be easily defined by the programmer. Many of the commands may be purchased from software suppliers throughout the United States.

APPLICATION USER COMMANDS

STAT — List the number of characters (Bytes) of storage remaining on the current disk, provide statistical information about particular files, and display or alter device assignment. This command uses the file control characters.

PIP — Load the file copy program (Peripheral Interchange Program-PIP) from the disk for subsequent disk copy and device management operations. This is a very versatile program. The detail function specification section of this manual describes the full range of capability of this program. This command uses the file control characters.

SYSMOV — Copy the disk operating system from a source disk or from memory and place it on the receiving disk. This program transfers the system (from track 0 and 1) of an existing disk to a new disk. It is used to create a new diskette. This command does not use file contol characters.

SUBMIT — Begin execution of an automatic command execution process. The submit command assumes that the user has already prepared a command file (type.SUB) with a text editor. This command does not use the file control characters.

ED — Load and execute the text editor program furnished with the disk operating system. This program is not easily understood by the user because it does not take advantage of the CPT video screen. For this reason, it is recommended that the user perform text editing in the word processing mode and then use the copy utilities (word processing to data processing and data processing to word processing) to convert files from CPT word processing format to CP/M file format. This program may not use the file control characters.

DSCOPY — This is a file copy utility. It will copy the entire disk on drive A to drive B.

The following commands are used by machine language programmers and systems software programmers. Manuals are available on the use of these commands under separate cover for ASM and DDT.

ASM — Load the assembler language program and assemble the specified source language instructions (file type .ASM) from disk. The application user will seldom be concerned with this command. This command does not use the file control characters.

LOAD — Load the file in Intel "hex" machine code format and produce a file in machine executable form (type .COM) which can be loaded into the TPA (this loaded program becomes a new command under the CCP) and is added to the program library on the disk. This command does not use file control characters.

DDT — Load the debugging aid into high memory within the TPA and start execution. File control characters may not be used with this program. A manual is available on the use of this command under separate cover.

DUMP — Dump the contents of a disk file in hex code.

MOVCPM — Regenerate the disk operating system for a particular memory size.

There are desirable commands available from the dealer at reasonable cost. These commands and their functions are listed below.

SORT — Sorts disk files into specified sequences. It can also select data based on predefined criteria and then sort the selected data into the specified sequence. This sort will handle up to 32 input files and sort on up to 32 input keys ascending or descending. It will handle fixed or variable length records and data elements. Manual is available under separate cover.

CRUN2 — Run-time module of Commercial BASIC. This module is used to initiate the execution of application programs written in CBASIC. Sometimes it is renamed to RUN.

CBAS2 — Commercial BASIC compiler. This program translates BASIC source lanaguage program statements into an intermediate (.INT) language file. The run-time module CRUN2 is then used to execute the application program that has been written in BASIC. A manual is available under separate cover.

Library commands are specified in the same manner as built-in commands, and additional commands can be easily defined by the progammer. As an added convenience, the library command can be preceded by a drive name, which causes the library command to be loaded from the specified drive into the TPA for execution. Thus, the command

B:STAT

causes the CCP to temporarily use drive B for the source of the STAT library command, and then return to the current disk for subsequent processing.

LIBRARY COMMAND EXAMPLES (Type .COM)

	Data Processing
Initiate the command to copy the disk operating system onto a new disk	A>SYSMOV
Copy all files from drive A to Drive B	A>PIP B:=*.*
Copy a basic source program from drive B to drive A	A>PIP A:=B:GL1.BAS
Determine the remaining amount of storage left on drive B	A>STAT B:
Edit (build) a submit parameter file	A>EDIT INITIAL.SUB
Determine the storage of each file on the disk	A>STAT *.*
Copy just the .COM files from drive A to drive B.	A>PIP B:=*.COM

Detail descriptions of each library command (except those described in a separate manual) may be found in the Detail Functional Specifications section of this manual.

2.6 CPT COPY UTILITY

The user should refer to the CompuPak interface manual (under separate cover) for detailed operating instructions of the CPT word processing (WP) to data processing and data processing (DP) to word processing file conversion utility. It can only be used just after the computer has been reset and the interface disk loaded. Several operational considerations should be kept in mind:

1. The exclamation mark on the Pa KEY will be converted to an upside exclamation mark (an ASCII backslash) when going from WP to DP. This is used for commercial basic programs.

2. The underscore will seem to disappear when converted from WP to DP but will reappear when converted back from DP to WP.

3. Hat codes are not converted. They are discarded and the text is compressed.

3.0 HOW TO

This section of the manual has been prepared for the purpose of presenting some of the most often used procedures. These procedures show the proper combination of command steps for performing complete functions; such as, creating a new disk and establishing the desired files on the disk, building an automatic job step execution sequence, etc. The user will need to read the "overview" of this manual and in some cases the "Detail Functional Specifications" section of the manual in order to fully understand the suggested command sequences.

3.1 COPY A NEW DISK

Most disks are sold already formatted for immediate use (IBM 3740 single density soft sectored). This procedure assumes that a new disk has been formatted. The user must purchase formatted disks or CompuPak will not work properly.

This procedure is used when the user desires an exact copy of the source disk. Place the source disk in drive A and the new disk in drive B.

METHOD 1

	Data Processing
Copy the system from drive A to drive B	A>SYSMOV
Computer States ----------	SYSMOV VERSION 2.1
Enter A ----------	SOURCE DRIVE NAME (OR RETURN TO SKIP)A
Depress RETURN key ---------	SOURCE ON A THEN TYPE RETURN
Computer States ----------	FUNCTION COMPLETE
Enter B ----------	DESTINATION DRIVE NAME (OR RETURN)B
Depress RETURN key ---------	DESTINATION ON B THEN TYPE RETURN
Computer States ----------	FUNCTION COMPLETE
Computer States ----------	DESTINATION DRIVE NAME (OR RETURN)
Depress RETURN key ---------	A>
Copy all of the files from drive A to Drive B	A>PIP B:=*.*
Computer States ----------	COPYING -
	STAT.COM
	etc.

METHOD 2

Data Processing

Copy the entire source disk on drive A to the disk on drive B. WARNING Do not place the wrong disk in drive B as it will be entirely written over by the diskette in drive A.

Computer States ----------

```
A>DSCOPY
  SOURCE ON A, DEST ON B, THEN RETURN
```

Display the directory to be sure the files are present ..

```
A>DIR B:
```

Note: Drive A corresponds to station 1 (leftmost disk drive) and Drive B corresponds to station 2 (rightmost disk drive) on the CPT.

The user may not wish to copy all of the files. Frequently only the command files are desired. When this is true substitute the following command for the PIP command line in method 1 above.

```
A PIP B:=*.COM
```

This causes only the command files (Type .COM) to be copied. All other files are left on the disk in drive A.

3.2 ERASE ALL FILES ON THE DISK

This procedure removes the files, but not the system. The user may then place a new set of files on the disk.

Place the old disk in drive B.

Data Processing

Switch disks .....

```
A>B:
```

Erase all files .....

Computer prompts ----------

Enter Y ----------

```
B>ERA *.*
B>ALL FILES Y/N
B>ALL FILES Y/N Y
```

Verify procedure .....

Computer states ----------

```
B>DIR
B>NOT FOUND
```

Switch disks .....

```
B>A:
A>
```

Note: The reason the disk drives were switched is because the user could forget to give the disk drive designation in which case the CCP will automatically use the current drive. This could cause the files on drive A to be inadvertently ERASED!

3.3 BACK UP A DATA FILE

This procedure is used when the user would like to back up the data files on another disk.

Place the source disk in Drive A and the backup disk in drive B.

	Data Processing
Determine that the files are present on the source disk	A>DIR *.DAT
Copy the files to the backup disk	A>PIP B:=*.DAT
Computer states ----------	COPYING - LEDGER.DAT ENTRY.DAT
Determine that the files are now on the backup disk	A>DIR B:

Note: Be sure to place a date on the backup disk label indicating when the last backup occured.

3.4 PRINT A FILE ON THE PRINTER

The user may print a file on the printer either by the use of PIP or PRINT commands. Care must be exercised to only print ASCII files on the printer. DO NOT print files with the type .COM, .INT or .HEX on the printer. The user may want to set top of form on the printer by causing a form feed. This is done by depressing the following sequence of keys:

PRINT (Code S)
MOVE (Code L)
RETURN
PRINT (Code S)

	Data Processing
Print data file using the PIP command ..... (Will number each line, works on the Rotary V only)	A>PIP PRN:=GLACCNT.DAT
Print data file without line numbering	A>PIP LPT:=GLACCNT.DAT[P55]

DO NOT TRY TO PRINT files of the type .COM or .INT. These are machine language files and contain many characters which are not within the ASCII character set and therefore are not displayable.

3.5 CHANGE FROM A ROTARY V TO A ROTARY IV PRINTER

CompuPak is initially set to use a Rotary V printer. The user may change this at any time by executing the following procedure.

Data Processing

Change to a Rotary IV Printer by typing	A>STAT LST:=UL1:
Change to a Rotary V (only required if printer has been changed without reloading the reference disk)	A>STAT LST:=LPT:

4.0 DETAIL FUNCTIONAL SPECIFICATIONS

This section of the manual presents the full range of capability for the library commands that are furnished with the operating system.

Files will generally be referenced in this section by two terms:

1. Unambiguous File Name (ufn). This term is used to indicate that there is only one file that fits this name.

   For example: GL1.BAS

2. Ambiguous File Name (afn). This term is used to indicate that more that one file may fit the conditions of this name. This occurs when file control characters are employed. For example: Using the file reference GL?.* would cause the computer to select all files beginning with GL in the first two positions, any character in the third position and any file type.

4.1. STAT

The STAT command provides general statistical information about file storage and device assignment. It is initiated by typing one of the following forms:

```
STAT
STAT "command line"
```

Special forms of the "command line" allow the current device assignment to be examined and altered as well. The various command lines which can be specified are shown below, with an explanation of each form shown to the right.

STAT

If the user types an empty command line, the STAT command calculates the storage remaining on all active drives, and prints a message

A: R/W, SPACE: 120K

or

A: R/O, SPACE: 120K

for each active drive A or A and B, where R/W indicates the drive may be read or written, and R/O indicates the drive is read only (a drive becomes R/O by explicitly setting it to read only, as shown below, or by inadvertantly changing diskettes without performing a warm start). The space remaining on the diskette in drive A or B is given in thousands of characters

by nnn, i.e. 120 means 120,000 characters (Kilobytes) are available on drive A.

STAT x:

If a drive name is given, then the drive is selected before the storage is computed. thus, the command "STAT B:" could be issued while logged into drive A, resulting in the message

BYTES REMAINING ON B: 120K

STAT afn
Example:
STAT *.COM

The command line can also specify a set of files to be scanned by STAT. The files which satisfy afn labels are listed in alphabetical order, with storage requirements for each file under the heading

RECS BYTS EX D:FILENAME.TYP

rrrr bbbK ee d:pppppppp.sss

where rrrr is the number of 128-character records allocated to the file, bbb is the number of characters (bytes) in thousands allocated to the file, (bbb=rrrr X 128 / 1024), ee is the number of 16K extensions (ee=bbb/16), d is the drive name containing the file (A or B), pppppppp is the (up to) eight-character file name, and sss is the (up to) three-character file type. After listing the individual files, the storage usage is summarized. The information that is of interest to the operator is underscored.

STAT x:afn
Example:
STAT B:*.COM

As a convenience, the drive name can be given ahead of the afn. In this case, the specified drive is first selected, and the form "STAT afn" is executed.

STAT x:=R/O
Example:
STAT A:=R/O

This form sets the drive given by x to read only, which remains in effect until the next warm or cold start takes place. When a disk is read-only, the message

BDOS ERR ON x: READ ONLY

will appear if there is an attempt to write to the read-only disk x. The CCP waits until a key is depressed before performing an automatic warm start (at which time the disk becomes R/W).

The STAT command also allows control over the physical to logical device assignment. In general, there are four logical peripheral devices which are, at any particular instant, each assigned to one of

several physical peripheral devices. The four logical devices are named:

CON:	The system console device (used by CCP for communication with the operator-keyboard and screen of CPT 8000).
RDR:	The paper tape reader device. Not implemented on CPT 8000).
PUN:	The paper tape punch device. Not Implemented on CPT 8000)
LST:	The output list device (Printer on CPT 8000 Rotary IV or Rotary V).

The actual devices attached to the CPT 8000 are driven by subroutines in the I/O portion of disk operating system (BIOS). Thus, the logical LST: device, for example, could actually be a Rotary IV or a Ratory V printer. In order to allow some flexibility in device naming and assignment, several physical devices are defined, as shown below:

TTY:	Teletype device. Used as CPT 8000 Console. Word Processing 45 line.
CRT:	Cathode ray tube device. Not implemented. DP 53 line.
BAT:	Batch processing. Not implemented.
UC1:	User-defined console. Not implemented.
PTR:	Paper tape reader. Not implemented.
UR1:	User-defined reader #1. Not implemented.
UR2:	User=defined reader #2. Not implemented.
PTP:	Paper tape punch. Not implemented.
UP1:	User-defined punch #1. Not implemented.
UP2:	User-defined punch #2. Not implemented.
LPT:	Line printer. Rotary V.
UL1:	User-defined list device #1. Rotary IV.

It must be emphasized that the physical device names may or may not actually correspond to devices which the names imply. That is, the CRT: device may be implemented as a high speed printer at some time in the future. The exact correspondence and driving subroutine is defined in the I/O (BIOS) portion of the disk operating system. This portion

of the operating system is modified by CPT Corporation as new products are implemented.

The possible logical to physical device assignments can be displayed by typing

```
STAT VAL:
```

The STAT prints the possible values which can be taken on for each logical device:

```
CON: = TTY:  CRT:  BAT:  UL1:
RDR: = TTY:  PTR:  UR1:  UR2:
PUN: = TTY:  PTP:  UP1:  UP2:
LST: = TTY:  CRT:  LPT:  UL1:
```

The underscored devices are currently implemented in the disk operating system.

In each case, the logical device shown to the left can take any of the four physical assignments shown to the right on each line, provided they have been implemented by CPT Corporation. The current logical to physical assignments are displayed by typing the command

```
STAT DEV:
```

which produces a listing of each logical device to the left, and the current corresponding physical device to the right. For example, the list might appear as follows:

```
CON: = TTY:
RDR: = UR1:
PUN: = PTP:
LST: = LPT:
```

The current logical to physical device assignment can be changed by typing a STAT command of the form

```
STAT LD(1) = PD(1), LD(2) = PD(2) , ... , LD(n) = PD(n)
```

where LD(1) through LD(n) are logical device names, and PD(1) through PD(n) are compatible physical device names (i.e., CON: can utilize TTY:, CRT:, BAT:, or UC1:). The following are valid STAT commands which change the current logical to physical device assignments:

```
STAT LST:=LPT:
STAT LST:=UL1:
```

4.2 LOAD ufn. (For Programmers only).

The LOAD command reads the file ufn, which is assumed to contain "hex" format machine code, and produces a memory image file which can be sub-

sequently executed. The load command is used with the ASM command (Manual is provided under separate cover for full line dealers). The assembler command (ASM) produces a hex output file. The file name ufn is assumed to be of the form

x.HEX

and thus only the name x need be specified in the command. The LOAD command creates a file named

x.COM

which marks it as containing machine executable code. The file is actually loaded into memory and executed when the user types the file name x immediately after the prompting character ">" printed by the CCP.

In general, the CCP reads the name x following the prompting character and looks for a built-in function name. If no function name is found, the CCP searches the system disk directory for a file by the name

x.COM

If found, the machine code is loaded into the TPA, and the program executes. Thus, the user need only LOAD a hex file once; it can be subsequently executed any number of times by simply typing the library name. In this way, the progammer can "invent" new commands in the CCP. (Distribution disks contain the library commands as .COM files, which can be deleted at the user's option.) The load operation can take place on an alternate drive if the file name is prefixed by a drive name. Thus,

LOAD B:PRINT

brings the LOAD program into the TPA from the currently logged disk and operates upon drive B after execution begins.

It must be noted that the PRINT.HEX file must contain valid Intel format hexadecimal machine code records (as produced by the ASM program, for example) which begin at storage location 100H, the beginning of the TPA. Further, the addresses in the hex records must be in ascending order; gaps in unfilled memory regions are filled with zeroes by the LOAD command as the hex records are read. Thus, LOAD must be used only for creating CP/M standard ".COM" files which operate in the TPA. Programs which occupy regions of memory other than the TPA can be loaded under the DDT library command.

4.3 PIP

PIP is the Peripheral Interchange Program which implements the basic media conversion operations necessary to load, print, punch, copy, and combine disk files. Only the underscored functions have been implemented at this time. The PIP program is initiated by typing one of the following forms

(1) PIP
(2) PIP "command line"

In both cases, PIP is loaded into the TPA and executed. In case (1), PIP reads command lines directly from the console, prompted with the "*" character, until an empty command line is typed (i.e., a single carriage return is issued by the operator). Each successive command line causes some media conversion to take place according to the rules shown below. Form (2) of the PIP command is equivalent to the first, except that the single command line given with the PIP command is automatically executed, and PIP terminates immediately with no further prompting of the console for input command lines. The form of each command line is

destination = source #1, source #2, ... , source #n

where "destination: is the file or peripheral device to receive the data, and "source #1, ..., source #n" represents a series of one or more files or devices which are copied from left to right to the destination.

When multiple files are given in the command line, i.e.

PIP A:=PARA1.TXT,PARA2.TXT

the individual files are assumed to contain ASCII characters, with an assumed file end-of-file character (CODE-Z) at the end of each file (see the O parameter to override this assumption). All files produced by text editors and application programs normally automatically satisfy this condition. Lower case ASCII alphabetics are internally translated to upper case to be consistent with CP/M file and device name conventions. Finally, the total command line length cannot exceed 255 characters (CODE-E can be used to force a physical carriage return for lines which exceed the console width).

The destination and source files may can be referenced with a preceding drive name (A: or B:) which defines the particular drive where the file may be obtained or stored. When the drive is not designated, the currently logged disk is assumed. Further, the destination file can also be the same name as one or more of the source files, in which case the source file is not altered until the entire joining or concatenation is complete. If the destination file already exists, it is removed if the command line is properly formed (it is not removed if an error condition arises), therefore the operator is protected from inadvertent typagraphical errors. The following command lines (with explanations to the right) are valid as input to PIP:

PIP PAR2.TXT=PAR1.TXT	Copy to the file named PAR2.TXT from the file named PAR1.TXT. PAR1.TXT remains unchanged.
PIP LTR.TXT=PAR1.TXT,PAR2.TXT	Join or concatenate files PAR1.TXT and PAR2.TXT and copy to file

	LTR.TXT, with PAR1.TXT and PAR2.TXT unchanged.
PIP PAR1.TXT=PAR1.TXT,PAR2.TXT	Concatenate files PAR1.TXT and PAR2.TXT into PAR1.TXT. PAR2.TXT is unchanged, however, PAR1.TXT now contains both files in the order of PAR1.TXT then PAR2.TXT.
PIP LETTER=B:PAR1.TXT	Move a copy of PAR1.TXT from drive B: to the current drive (A:) and rename PAR1.TXT to LETTER.
PIP B:LTR.TXT=B:PAR1,PAR2,PAR3	Concatenate file PAR1 from drive B: with PAR2 from current drive A: and PAR3 from current drive A: into the created file LTR.TXT on drive B:

For more convenient use, PIP allows the use of file control charcters for transferring files between disk drives. Examples of this use are

1. PIP B:=afn
   Example: PIP B:=*.COM

2. PIP A:=B:afn
   Example: PIP B:=A:*.COM

3. PIP ufn=B:
   Example: PIP LTR.TXT=B:

4. PIP A:ufn=B:
   Example: PIP A:LTR.TXT=B:

The first form copies all files from the currently logged disk which satisfy the afn to the same file names on drive B. The second form is equivalent to the first, where the source for the copy is drive A. The third form is equivalent to the command "PIP ufn=B:ufn" which copies the file given by ufn from drive B to the file ufn on drive A. The fourth form is equivalent to the third, where the source disk is explicitly given by B.

Note that the source and destination disks must be different in all of these cases. If an afn is specified, PIP lists each ufn which satisfied the afn as it is being copied. If a file exists by the same name as the destination file, it is removed upon successful completion of the copy, and replaced by the copied file.

The following PIP commands give examples of valid disk-to-disk copy operations:

PIP B:=*.COM	Copy all files which have the file type ".COM" to drive B from the current drive.

PIP A:=B:LTR.*	Copy all files which have the file name "LTR" to drive A from drive B.
PIP PRINT.ASM=B:	Equivalent to PRINT.ASM=B:PRINT.ASM
PIP B:PIP.COM=A:	Equivalent to B:PIP.COM=A:PIP.COM
PIP B:=ENTRY.BAS	Same as B:ENTRY.BAS=ENTRY.BAS
PIP B:=A:ENTRY.BAS	Same as B:ENTRY.BAS=A:ENTRY.BAS

PIP also allows reference to physical and logical devices which are attached to the system. The device names are the same as given under the STAT command, along with a number of specially named devices. The logical devices given in the STAT command are

CON: (console), RDR: (reader), PUN: (punch), and LST: (list)

while the physical devices are

TTY: (console, reader, punch, or list)
CRT: (console, or list), UC1: (console)
PTR: (reader), URl: (reader), UR2: (reader)
PTP: (punch), UPl: (punch), UP2 (punch)
LPT: (list), ULl: (list)

The destination device must be capable of receiving date (i.e., data cannot be sent to the reader), and the source devices must be capable of generating date (i.e., the LST: device cannot be read).

The additional device names which can be used in PIP commands are

PRN:	Same as LST:, except that tabs are expanded at every eighth character position, lines are numbered, and page ejects are inserted every 60 lines, with an initial eject (same as [T8NP]). This device name may be used to print a file on the printer.
NUL: (For Programmers)	Send 40 "nulls" (ASCII 0's) to the device. This can be issued at the end of punched output.
EOF: (For Programmers)	Send a end-of-file (ASCII CODE-Z) to the destination device (sent automatically at the end all ASCII data transfers through PIP).
OUT: (For Programmers)	Special PIP output destination which can be patched into the PIP program: PIP CALLs location 106H with data in register C for each character to transmit. Note that locations 109H through

1FFH of the PIP memory image are not used and can be replaced by special purpose drivers using DDT (see the DDT operator's manual).

File and device names can be interspersed in the PIP commands. In each case, the specific device is read until end-of-file (CODE-Z for ASCII files, and a real end of file for non-ASCII disk files). Data from each device or file is concatenated from left to right until the last data source has been read.

The destination device or file is written using the data from the source files, and an end-of-file character (CODE-Z) is appended to the result for ASCII files. Note if the destination is a disk file, then a temporary file is created ($$$ file type) which is changed to the actual file name only upon successful completion of the copy. Files with the extension ".COM" are always assumed to be non-ASCII (machine language).

The copy operation can be aborted at any time by depressing any key on the keyboard (a space bar suffices). PIP will respond with the message "ABORTED" to indicate that the operation was not completed. Note that if any operation is aborted, or if an error occurs during processing, PIP removes any pending commands which were set up while using the SUBMIT command (See library command SUBMIT).

Valid PIP commands are shown below.

PIP LST:=LTR.TXT	Copy LTR.TXT to the LST device and terminate the PIP program. Print the file LTR.TXT and add line numbers.
PIP	Start PIP for a sequence of commands (PIP prompts with "*").
*CON:=PAR1,PAR2,PAR3	Concatenate three files and displays them on the CON device, which is the CPT Screen.
*A.LTR=CON:,PAR1,PAR2	Create a LTR file by reading the CON (until a CODE-Z is typed), followed by data from PAR1, followed by data from PAR2.
*Depress RETURN	Single carriage return stops PIP.

The user can also specify one or more PIP parameters, enclosed in left and right square brackets, separated by zero or more blanks. Each parameter affects the copy operation, and the enclose list of parame-

ters must immediately follow the affected file or device. Generally, each parameter can be followed by an optional decimal integer value (the S and Z parameters are exceptions). The valid PIP parameters are listed below.

Dn Delete characters which extend past column n in the transfer of data to the destination from the character source. This parameter is used most often to truncate long lines which are sent to a (narrow) printer or console device.

E Echo all transfer operations to the console as they are being performed.

F Filter form feeds from the file. All imbedded form feeds are removed. The P parameter can be used simultaneously to insert new form feeds.

L Translate upper case alphabetics to lower case.

N Add line numbers to each line transferred to the destination starting at one, and incrementing by 1. Leading zeroes are suppressed, and the number is followed by a colon. If N2 is specified, then leading zeroes are included, and a tab is inserted following the number. The tab is expanded if T is set.

Pn Include page ejects at every n lines (with an inital page eject). If n = 1 or is excluded altogether, page ejects occur every 60 lines. If the F parameter is used, form feed suppression takes place before the new page ejects are inserted.

Qs z Quit copying from the source device or file when the string s (terminated by CODE-Z) is encountered.

Ss z Start copying from the source device when the string s is encountered (terminated by CODE-Z). The S and Q parameter can be used to "abstract" a particular section of a file (such as a paragraph). The start and quit strings are always included in the copy operation.

NOTE - the strings following the s and q parameters are translated to upper case by the CCP if form (2) of the PIP command is used. Form (1) of the PIP invocation, however, does not perform the automatic upper case translation.

(1) PIP
(2) PIP "command line"

Tn Expand tabs (CODE-I characters) to every nth column during the transfer of characters to the destination from the source.

U Translate lower case alphabetics to upper case during the copy operation.

V Verify that data has been copied correctly by rereading after the write operation (the destination must be a disk file).

The following are valid PIP commands which specify parameters in the file transfer:

PIP SYSMOV.COM=B:[V] Copy SYSMOV.COM from drive B to the current drive and verify that the data was properly copied.

PIP LPT:=PAR1[NT8U] Copy PAR1 to the LPT: device; number each line, expand tabs to every eighth column, and translate lower case alphabetics to upper case.

PIP LTR.A=PAR1 [sINTROz qEND TXTz]
Copy from the file PAR1 into the file LTR. Start the copy when the string "INTRO" has been found, and quit copying after the string "END TXT" is encountered.

PIP PRN:=LTR[P50] Send LTR to the LST: device, with line numbers, tabs expanded to every eighth column, and page ejects at every 50th line. Note that NT8P60 is the assumed parameter list for a PRN file; P50 overrides the default value.

4.4 ED ufn

The ED program is a system context editor, which allows creation and alteration of ASCII files in the disk operating system environment. This is not a video text editor. The user may find it easier to create files under word processing, then use the file utilities to convert the word processing files to data processing files. Complete details of operation are given in the ED user's manual, "ED: a Context Editor for the CP/M Disk System." In general, ED allows the operator to create and operate upon source files which are organized as a sequence of ASCII characters, separated by end-of-line characters (a carriage-return line-feed sequence). There is no practical restriction on line length (no single line can exceed the size of the working memory), which is instead defined by the number of characters typed between carriage returns. The ED program has a number of commands for character string searching, replacement, and insertion, which are useful in the creation and correction of programs or text files under the disk operating system. Although the disk operating system has a

limited memory work space area (approximately 20,000 characters in the CPT 8000), the file size which can be edited is not limited, since data is easily "paged" through this work area.

Upon initiation, ED creates the specified source file, if it does not exist, and opens the file for access. The programmer then "appends" data from the source file into the work area, if the source file already exists (see the A command), for editing. The appended data can then be displayed, altered, and written from the work area back to the disk (see the W command). Particular points in the program can be automatically paged and located by context (see the N command), allowing easy access to particular portions of a large file.

Given that the operator has typed

```
ED LTR.TXT
```

the ED program creates an intermediate work file with the name

```
LTR.$$$
```

to hold the edited data during the ED run. Upon completion of ED, the LTR.TXT file (original file) is renamed to LTR.BAK, and the edited work file is renamed to LTR.TXT. Thus the LTR.BAK file contains the original (unedited) file, and the LTR.TXT file contains the newly edited file. The operator can always return to the previous version of a file by removing the most recent version, and renaming the previous version. Suppose, for example, that the current LTR.TXT file was improperly edited; the sequence of CCP command shown below would reclaim the backup file.

```
DIR LTR.*                 Check to see that BAK file is
                          available.

ERA LTR.TXT               Erase most recent version.

REN LTR.TXT=LTR.BAK       Rename the BAK file to LTR.TXT.
```

Note that the operator can abort the edit at any point (reboot, power failure, CODE-C or Q command) without destroying the original file. In this case, the BAK file is not created, and the original file is always intact.

The ED program also allow the user to "ping-pong" the source and create backup files between two disks. The form of the ED command in this case is

```
ED ufn B:
```

where ufn is the name of a file to edit on the currently logged disk, and B is the name of the alternate drive. The ED program reads and processes the source file, and writes the new file to drive B, using the name ufn. Upon completion of processing, the original file becomes

the backup file. Thus, if the operator is addressing disk A, the following command is valid:

```
ED LTR.TXT B:
```

which edits the file LTR.TXT on drive A, creating the new file LTR.$$$ on drive B. Upon completion of a successful edit, A:LTR.TXT is renamed to A:LTR.BAK and B:LTR.$$$ is renamed to B:LTR.ASM. For user convenience, the currently logged disk becomes drive B at the end of the edit. Note that if a file by the name B:LTR.TXT exists before the editing begins, the message

```
FILE EXISTS
```

is printed at the console as a precaution against accidently destroying a source file. In the case, the operator must first ERAse the existing file and then restart the edit operation.

Similar to other library commands, editing can take place on a drive different from the currently logged disk by preceding the source file name by a drive name. Examples of valid edit requests are shown below

ED A:LTR.TXT	Edit the file LTR.TXT on drive A, with new file and backup on drive A.
ED B:PAR1 A:	Edit the file PAR1 on drive B to the temporary file PAR1.$$$ on drive A. On termination of editing, change PAR1 on drive B to PAR1.BAK, and change PAR1.$$$ on drive A to PAR1.

4.5 SYSMOV

The SYSMOV library command allows generation of an initialized diskette containing the operating system. The SYSMOV program prompts the console for commands, with interaction as shown below.

SYSMOV	Initiate the SYSMOV program.
SYSMOV VERSION m.m	SYSMOV sign-on message.
SOURCE DRIVE NAME (OR RETURN TO SKIP)	Respond with the drive name (one of the letters A or B of the disk containing the disk operating system; usually A. Typing drive name A will cause the response:
SOURCE ON A THEN TYPE RETURN	Place a diskette containing the operating system on drive B.

	Answer with carriage return when ready.
FUNCTION COMPLETE	System is copied to memory. SYSMOV will then prompt with:
DESTINATION DRIVE NAME (OR RETURN TO REBOOT)	If a diskette is being initialized, place the new disk into drive B and answer with the drive name. Otherwise, depress the carriage return and the program will return to the operating system. Typing drive name B will cause SYSMOV to prompt with:
DESTINATION ON B THEN TYPE RETURN	Place new diskette into drive B; type return when ready.
FUNCTION COMPLETE	New diskette is initialized in drive B.

The "DESTINATION" prompt will be repeated until a single carriage return is typed at the console, so that more than one disk can be initialized.

Upon completion of a successful system move, the new diskette contains the operating system, and only the built-in commands are available. A factory-fresh IBM-compatible diskette appears to the disk operating system as a diskette with an empty directory; therefore, the operator must copy the appropriate COM files from an existing diskette to the newly constructed diskette using the PIP or DSCOPY library commands.

The user can copy all files from an existing diskette by typing the PIP command

```
PIP B: = A:*.*[V]
```

which copies all files from disk drive A to disk drive B, and verifies that each file has been copied correctly. The name of each file is displayed at the console as the copy operaton proceeds.

It should be noted that a SYSMOV does not destroy the files which already exist on a diskette; it results only in construction of a new operating system. Further, if a diskette is being used only on drive B and will never be the source of a bootstrap operation on drive A, the SYSMOV need not take place. In fact, a new diskette needs absolutely no initialization to be used with CP/M.

4.6. SUBMIT ufn parm#1 ... parm#n (For Programmers).

The SUBMIT command allows the disk operating system commands to be batched together for automatic processing. The ufn given in the SUBMIT

command must be the file name of a file which exists on the currently logged disk, with an assumed file type of "SUB." The SUB file contains operating system prototype commands, with possible parameter substitution. The actual parameters parm #1 ... parm #n are substituted in the prototype commands, and, if no errors occur, the file of substituted commands are processed sequentially by the operating system.

The prototype command file is created using a text editor, with interspersed "S" parameters of the form

```
$1 $2 $3 ... $n
```

corresponding to the number of actual parameters which will be included when the file is submitted for execution. When the SUBMIT command is executed, the actual parameters parm#1 ... parm#n are paired with the formal parameters $1 ... $n in the prototype commands. If the number of formal and actual parameters does not correspond, the submit function is aborted with an error message at the console. The SUBMIT function creates a file of substituted commands with the name

```
$$$.SUB
```

on the logged disk. When the system reboots (at the termination of the SUBMIT), this command file is read by the CCP as a source of input, rather than the console. If the SUBMIT function is performed on any disk other than drive A, the commands are not processed until the disk is inserted into drive A and the system reboots. Further, the user can abort command processing at any time by typing a spacebar when the command is read and echoed. In this case, the $$$.SUB file is removed, and the subsequent commands come from the console. Command processing is also aborted if the CCP detects an error in any of the commands. Programs which execute under the system can abort processing of command files when error conditions occur by simply erasing any existing $$$.SUB file.

In order to introduce dollar signs into a SUBMIT file, the user may type a "$$" which reduces to a single "$" within the command file. Further, an up-arrow symbol may precede an alphabetic character x, which produces a single CODE-X character within the file.

The last command in a SUB file can initiate another SUB file, thus allowing chained batch commands.

Suppose the file ASMBL.SUB exists on disk and contains the prototype commands

```
ASM $1
DIR $1.*
ERA *.BAK
PIP $2:=$1.PRN
ERA $1.PRN
```

and the command

```
SUBMIT ASMBL X PRN
```

is issued by the operator. The SUBMIT program reads the ASMBL.SUB file, substituting "X" for all occurrences of $1 and "PRN" for all occurrences of $2, resulting in a $$$.SUB file containing the commands

```
ASM X
DIR X.*
ERA *XBAK
PIP PRN:=X.PRN
ERA X.PRN
```

which are executed in sequence by the CCP.

The SUBMIT function can access a SUB file which is on an alternate drive by preceding the file name by a drive name. Submitted files are only acted upon, however, when they appear on drive A. Thus, it is possible to create a submitted file on drive B which is executed at a later time when it is inserted in drive A.

4.7 DUMP ufn (For Programmers)

The DUMP program types the contents of the disk file (ufn) at the console in hexadecimal form. The file contents are listed sixteen bytes at a time, with the absolute byte address listed to the left of each line in hexadecimal. Long type outs can be aborted by pushing the typebar key during printout. (The source listing of the DUMP program is given in the "CP/M Interface Guide" as an example of a program written for the CP/M environment).

4.8 DSCOPY

This program will copy the entire diskette on Drive A to the diskette on Drive B. To initiate the DSCOPY program the user should type

```
DSCOPY
```

The program will sign on with the message

```
DSCOPY RELEASE A

INSERT SOURCE IN A, DESTINATION IN B, HIT RETURN
```

The user should depress RETURN to execute the copy program. The program will return to the CCP as soon as it is finished.

5.0 ERROR CONDITIONS AND MESSAGES

Most all error conditions are accompanied with an error message. There are two conditions which are not accompanied with an error message:

1. System hang up. This will occur if a data processing disk does not have a system on track 0 and 1, it is placed in drive A and the STOP key is depressed or an application program has completed its function and is returning to the disk operating system (CCP). The user must reset the CPT and reload the reference disk in order to restart.

2. End of printer ribbon. The computer will stop without indicating to the operator that an end of ribbon condition has been encountered. Change ribbons and depress the printer reset button to continue.

5.1 DISK ERROR CONDITIONS AND MESSAGES

There are three error situations which the Basic Disk Operating System (BDOS) intercepts during file processing. When one of these conditions is detected, the BDOS prints the message:

BDOS ERR ON x: error

where x is the drive name, and "error" is one of the three error messages:

BAD SECTOR
SELECT
READ ONLY

BAD SECTOR. The BAD SECTOR message may indicate one of the following errors:

1. The disk controller electronics has detected an error condition in reading or writing the diskette. This condition is generally due to a malfunctioning disk controller, or an extremely worn diskette. If you find that your system reports this error more than once a month, you should check the state of your controller electronics, and the condition of your media.

2. A disk is not present in the selected drive. This error will be accompanied with a CPT error message. Place the desired disk in the empty drive and depress STOP to continue.

3. A word processing disk has been placed in the drive. Remove the word processing disk and replace with any data processing disk, depress STOP to continue.

4. The operator changed disks without depressing the STOP key before continuing. The STOP key (warm start) must be

depressed any time diskettes are changed unless the application program prompts the operator to change the diskette.

Recovery from the BAD SECTOR ERROR condition is accomplished by typing a CODE-C (STOP KEY) to reboot (this is the safest!), or a return, which simply ignores the bad sector in the file operation. Note, however, that typing a return may destroy your diskette integrity if the operation is an electronic failure and the computer is performing a directory write, so make sure you have adequate backups in this case.

SELECT. The SELECT error occurs when there is an attempt to address a drive beyond the A through B range. In this case, the value of x in the error message gives the selected drive. The system reboots following any input from the console.

READ ONLY. The READ ONLY message occurs when there is an attempt to write to a diskette which has been designated as read-only in a STAT command, or has been set to read-only by the BDOS. In general, the operator should reboot either by using the warm start procedure (CODE-C) or by performing a cold start whenever the diskettes are changed. If a changed diskette is to be read but not written, BDOS allows the diskette to be changed without the warm or cold stat, but internally marks the drive as ready-only. The status of the drive is subsequently changed to read/write if a warm or cold start occurs. Upon issuing this message, CCP waits for input from the console. An automatic warm start takes place following any input.

5.2 CBASIC II RUN TIME ERRORS

These errors may be encountered when running CBASIC II programs. The run time module named CRUN2 is in control of the computer when these errors are encountered.

NO INTERMEDIATE FILE

A file name was not specified with the CRUN2 command, or no file of the type .INT with the specified file name was found on the disk.

IMPROPER INPUT - REENTER

This message occurs when the fields entered from the console do not match the fields specified in the program. This can occur when field types do not match or the number of fields entered is different from the number of fields specified. Following this message all values required for input must be reentered.

Other errors detected cause a 2 letter code to be printed. If the code is preceded by the word WARNING, execution of the program will continue. If the code is preceded by the word ERROR, execution of the program is terminated. The possible error codes are:

CODE	DESCRIPTION
WARNING CODES	
DZ	A number was divided by zero. The result is set to the largest valid CBASIC number. This is a very large number.
FL	A field length greater than 255 characters was encountered during a READ LINE input statement in the program. The first 255 characters of the record are retained; the other characters are ignored.
LN	The argument given in the LOG function was zero or negative. The value of the argument is returned.
NE	A negative number was specified before the raise to a power operator. The absolute value of the number is used in the calculation.
OF	A calculation produced a number too large. The result is set to the largest valid CBASIC number.
SQ	A negative number was specified in the SQR function. The absolute value is used.
ERROR CODES	
AC	The string used as the argument in an ASC function evaluated to a null string.
BN	The value following the BUFF option in an OPEN or CREATE file definition statement is less than 1 or greater than 52.
CC	A chained program's code area is larger than the main program's code area. Use the %CHAIN directive in the main program to correct the problem. This is a programmer problem.
CD	A chained program's data area is larger than the main program's data area. Use the %CHAIN directive in the main program to solve this problem. This is a programmer problem.
CE	An error occured while closing a disk file.
CF	A chained program's constant area is larger than the main program's constant area. Use the %CHAIN directive in the main program. This is a programmer problem.
CP	A chained program's variable storage area is larger than the main program's variable storage area. Use the %CHAIN directive in the main program. This is a programmer problem.
CS	A chained program reserved a different amount of memory with a SAVEMEM statement than the main program. This is a programmer problem.

CU A close statement specified a file number that was not active. This is a programmer problem.

DF An OPEN or CREATE was specified with a file number than was already active. This is a programmer problem.

DU A DELETE statement specified a file number that was not active.

DW An error occurred while writing to a file. This occurs when either the directory or the disk is full.

EF A read past the end of file occurred on a file for which no IF END statement had been executed in the program. This can occur if the file is not present on the disk. This may also occur in some programs if the file does not have a proper trailer record which identifies the end of file. A sequential file will cause this error if any of the records in the file contains fewer or more than the expected number of fields.

ER An attempt was made to write a record of length greater than the maximum record size specified in the associated OPEN, CREATE or FILE statement. Check the data to see if it overflowed the allotted number of characters specified in the file formats for the program.

FR An attempt was made to rename a file that already exists in the directory of the disk.

FT A file statement was executed when 20 files were already active. This is a programmer problem.

FU An attempt was made to read or write to a file that was not active.

IF An invalid character was found in a disk file name. A colon may never appear inbedded in the name. Question marks and asterisks may only appear in ambiguous file names.

IR A record number less than one was specified.

IV An attempt was made to execute an .INT file created by a version 1 compiler. To use CRUN2 a program must be recompiled using the version 2 compiler, CBAS2.

IX A FEND statement was encountered prior to executing a RETURN statement. All multiple line functions must exit with a RETURN statement.

LW A line width less than 1 or greater than 133 was specified in an LPRINTER WIDTH statement.

ME An error occured while creating or extending a file because the disk directory was full.

MP The third parameter in a MATCH function was zero or negative.

NF The file number specified was less than 1 or greater than 20.

NM There was insufficient memory to load the program.

NN An attempt was made to print a number with a PRINT USING statement but there was not a numeric data field in the USING string.

NS An attempt was made to print a string with a PRINT USING statement but there was not a string field in the USING string.

OD A READ statement was executed with no DATA statement, or all data statements having already been read.

OE An attempt was made to OPEN a file that doesn't exist. Check the directory to see that all required files are present.

OI The expression specified in an ON...GOSUB or an ON...GOTO statement evaluated to a number less than 1 or greater than the number of line numbers contained in the statement.

OM The program ran out of memory during execution.

QE An attempt was made to PRINT a string containing a quotation mark to a file. Quotation marks can only be written to files when using the PRINT USING option of the PRINT statement.

RB Random access was attempted to a file activated with the BUFF option specifying more than 1 buffer.

RE An attempt was made to read past the end of a record in a fixed file.

RG A RETURN occurred for which there was no GOSUB.

RU A random read or print was attempted to other than a fixed file.

SB An array subscript was used which exceeded the boundaries for which the array was defined.

SL A concatenation operation resulted in a string of more than 255 bytes. No single field can exceed 256 bytes.

SO The file specified in a SAVEMEM statement could not be located on the referenced disk. The expression specifying the file name must include the type if one is present. A type of .COM is not forced.

SS The second parameter of a MID$ function was zero or negative.

TF An attempt was made to have more than 20 active files simultaneously.

TL A TAB statement contained a parameter less than 1 or greater than the current line width.

UN A PRINT USING statement was executed with a null edit string or an escape character was the last character in an edit string.

WR An attempt was made to write to a file after it had been read, but before it had been read to the end of the file.

6.0 APPENDIX

The appendicies contain the glossary of terms and the index to the manual. The glossary of terms will be expanded as the CPT receives user feedback from the field organizations.

6.1 CRT Display Driver (For Programmers)

The display driver is the program which controls the way in which characters are displayed on the screen. It is written to support the Lear Siegler ADM-3A terminal protocol. The screen is 54 lines long and each line contains 80 characters. The top line is the status line and is used to display error messages. It is unavailable to CP/M users. Therefore there are 53 usable lines for application programs. ASCII character codes are supported in the following ways:

The printable characters with ASCII codes 32 through 127 are supported as standard with these exceptions:

Code 92 is displayed as "/"
Code 94 is displayed as "@"
Code 95 is displayed as "-"
Code 123 is displayed as down arrow
Code 124 is displayed as "1/2"
Code 125 is displayed as up arrow
Code 126 is displayed as "1/ "

The first 32 character codes may contain screen control characters. The following codes are supported:

CODE F (CTL 6) Return the cursor to the beginning of the current line.

CODE G (CTL 7) Beep the buzzer.

CODE H (CTL 8) Backspace, back the cursor to the current column position of the last line.

CODE I (CTL 9) Tab, advance the cursor to the next position which is a multiple of eight.

CODE J (CTL 10) Line feed, advance the cursor to the current column position of the next line (down one line).

CODE K (CTL 11) Vertical tab, move cursor up one line.

CODE L (CTL 12) Forespace, advance the cursor to the right one position.

CODE M (CTL 13) Carriage return, advance the cursor to the begining of the next line.

CODE Z (CTL 26) Clear Screen, advance the cursor to the first column of the first line (home cursor).

CODE ; (CTL 27) ESCape, the ASCII escape code followed by the character "=" represent absolute cursor addressing. The next two characters following the ESC = sequence represent the

absolute cursor row and column respectively (Y,X). The row and column assigned values begin with the ASCII code for "space" (CODE 32).

CODE ; (CTL 27) ESCape, the ASCII escape code followed by the character "*" will clear the screen and home the cursor.

CODE ↠ (CTL 30) Home cursor, position the cursor to the begining of the screen.

6.2 GLOSSARY OF TERMS

ASCII	A national standard that has been set for representing data in computer systems.
BDOS	Basic Disk Operating System. This is the portion of the disk operating system that controls the disk drive input/output.
BINARY	Binary notation is a mathematical system based on a number system to the base 2. This is contrasted to the decimal number system which is a number system to the base 10.
BIOS	Basic Input Output System. This is the portion of the disk operating system that controls the keyboard, screen and printer of the CPT 8000.
BIT	Binary Digit. The smallest divisable unit of data representation in a computer. A binary digit may only represent an on or off condition.
BOOT	The process of loading the disk operating system into memory from track 0 and 1 of the diskette and transfering control to the Command Control Processor. The operator is signaled that the transfer is complete when the prompter A is displayed on the screen.
BOOTSTRAP	Synonymous to BOOT.
BYTE	A byte equals eight binary digits. The byte is used to represent data in the computer. It is generally equivalent to one character; that is, one letter or number or special symbol (period, comma, etc.).
CCP	Command Console Processor. The CCP is that portion of the disk operating system that interprets the operator's commands for the computer.

CHARACTER — A character is one letter, one number or one special symbol (period, comma, etc.). A character is generally equivalent to one byte.

COLD BOOT — The process of loading and transferring control to the disk operating system just after the machine has been turned on or the reset button at the rear of the CPT 8000 has been pressed. A cold boot or start requires the operator to use both the CompuPak interface disk and the CompuPak CP/M disk.

COLD START — See Cold Boot.

COMMAND, BUILT IN — Built in commands are available to the operator at all times. They are part of the operating system that resides in high memory. They may be executed by typing the command and pressing the RETURN key.

COMMAND, LIBRARY — Library commands reside on the diskette. They are part of the overall library files stored on the disk. They may be recognized by the file type .COM. They may be executed by typing the name of the command and pressing the RETURN key.

CONCATENATION — The joining of one file onto the end of another. PIP is capable of concatenating files together.

CP/M — A registered trademark of Digital Research Corporation. It is the Control Program Monitor of a computer system. It supervises the operation of application programs and operator functions.

CPU — Central Processor Unit. That portion of the computer that controls its functions. The CPU provides the arithmetic logic, input/output instructions, logical operators, data manipulation instructions and miscellaneous control instructions to the computer.

CRT — Cathode Ray Tube. This is the screen of the CPT 8000.

CURRENT DISK — The current disk (currently logged disk) is the disk that is designated ahead of the greater than prompter, either A or B. This will be the drive used if the user does not give the disk drive designator in the command line of a console instruction.

DIRECTORY	That portion of the diskette that is used to locate files that have been stored in the diskette library. The directory may be displayed by typing DIR a pressing RETURN.
EXTENT	An extent is equivalent to one entry in the diskette directory. It holds approximately 16,000 characters. If the file is larger than this the operating system will automatically enter as many additional extents in the directory as are needed to store the file.
ECHOES	A computer is said to echo data when the character typed on the keyboard appears twice on the screen or the printer follows the keyboard output as it is entered.
FILE	One or more records that have been set aside on a disk for a specific purpose.
FILE CONTROL CHAR	A character used to aid the operator in referencing groups of similar files. The two file contol characters are * and ?.
HARD DISK	A storage device similar to the floppy disk in that it is a random access device, but stores many times more data and operates at very high speeds. Usually the disk storage is non removable.
HEXADECIMAL	A number system to the base 16. This system is used because it can represent all combinations of a byte (eight binary digits). The characters 0-9 and A-F (16 symbols) are used to represent the digits of hexadecimal.
KILOBYTE	A kilobyte is equivalent to one thousand bytes.
MEGABYTE	A megabyte is equivalent to one million bytes.
MEMORY	That place in the CPT 8000 where data, application programs and the disk operating system temporarily reside. The CPT 8000 has 64,000 characters of internal memory. The operator should understand that the memory used in the CPT is a volatile type. This means that if the computer is turned off or there is a temporary power interruption all information in the memory will be lost!
PROGRAM	A set of computer instructions logically arranged for the purpose of performing a set

	of specific functions. Programs are usually written in an assembler, BASIC, FORTRAN or COBOL language.
READ ONLY (R/O)	This is a condition set by STAT that will only allow programs to read a diskette. If a write is attempted a BDOS error will occur.
READ/WRITE (R/W)	This condition exists initially. It may be changed by STAT. This condition allows applications programs to both read and write a file on the diskette.
RECORD, LOGICAL	A logical record may be whatever length the application programmer established. The operating system and application programs automatically keep track of how many physical records are required for each logical record or visa versa.
RECORD, PHYSICAL	A physical record contains 128 characters in single density format. There are 3328 records on a single density IBM 3740 diskette.
SECTOR	A term for referencing information on the disk. There are 26 sectors on an IBM 3740 standard disk.
SUBROUTINE	A portion of a program that is utilized two or more times. Subroutines may also be common to two or more programs. Subroutines save the programmer the effort of writing the same code every time a similar function is required. They also save space in the computer due to the elimination of repetitive code.
TRACK	A term for referencing information on the disk. There are 77 tracks on an IBM 3740 standard disk.
TPA	Transient program area. This area is used to store application programs for computer execution. The operating system copies the application program from the disk into the transient program area just before execution.
WARM BOOT	The process of loading and transfering control to the disk operating system when CODE C or the STOP key is depressed. A warm boot must be used when diskettes are changed. During the warm boot the CCP reads the directory of the new diskette and places it in memory. The CCP refers to the directory it has in memory

for reading and writing information on the diskette. Careless changing of diskettes without intiating a warm boot can cause disasterous results!

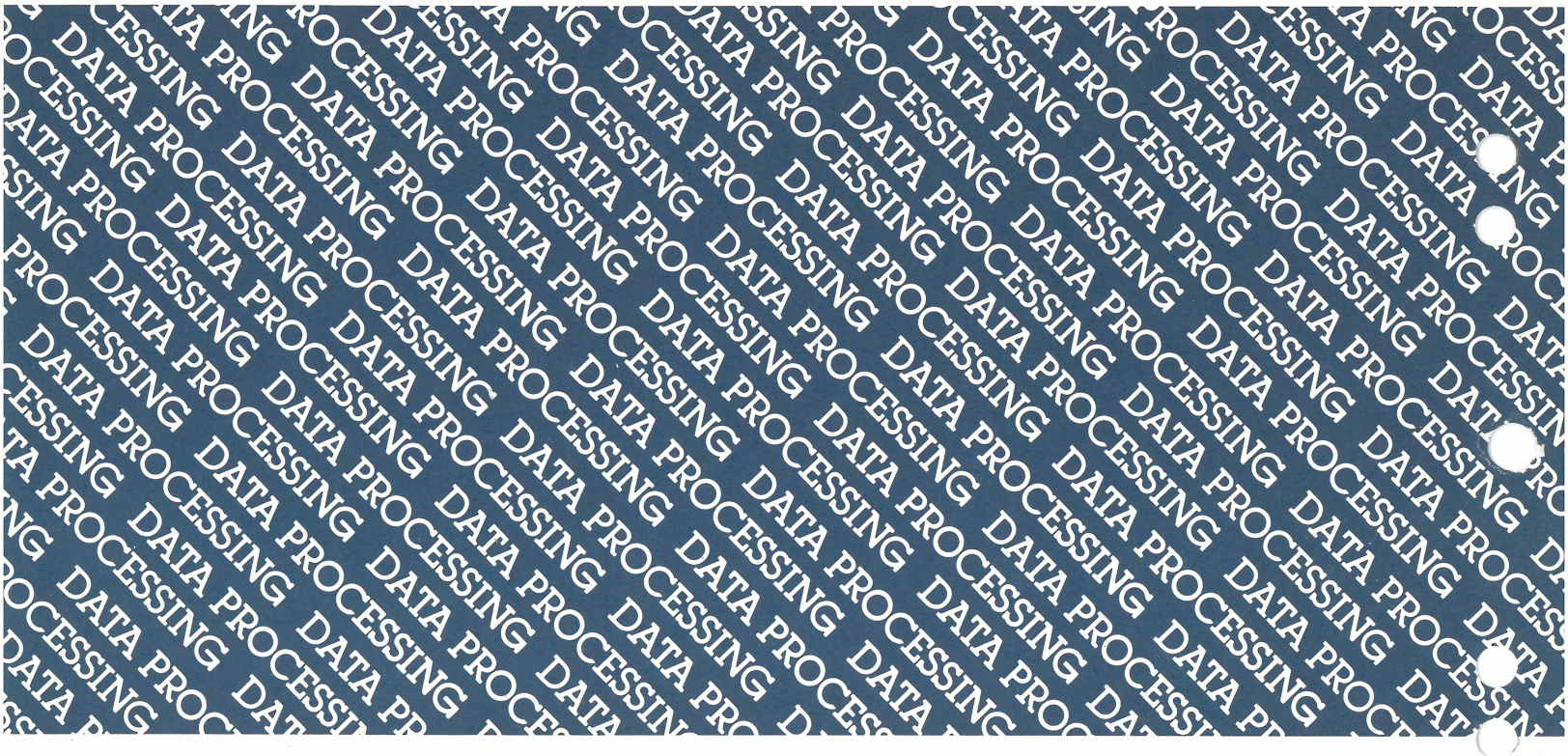
WARM START — See warm boot.

ASCII CHARACTER CODES

Graphics y,x	Decimal	Hex	Code Key	Character	Cursor Commds
	000	0		NUL	
	001	1	A	SOH	
	002	2	B	STX	
	004	4	D	EOT	
	005	5	E	ENQ	
	006	6	F	ACK	
	007	7	G	BEL	
	008	8	H	BS	←
	009	9	I	HT	TAB
	010	A	J	LF	↓
	011	B	K	VT	↑
	012	C	L	FF	
	013	D	M	CR	→
	014	E	N	SO	
	015	F	O	SI	
	016	10	P	DLE	
	017	11	Q	DC1	
	018	12	R	DC2	
	019	13	S	DC3	
	020	14	T	DC4	
	021	15	U	NAK	
	022	16	V	SYN	
	023	17	W	ETB	
	024	18	X	CAM	
	025	19	Y	EM	
	026	1A	Z	SUB	CLR
	027	1B		ESCAPE	
	028	1C		FS	
	029	1D		GS	
	030	1E		RS	HOME
	031	1F		US	
1	032	20		SPACE	
2	033	21		!	
3	034	22		"	
4	035	23		#	
5	036	24		$	
6	037	25		%	
7	038	26		&	
8	039	27		'	
9	040	28		(	
10	041	29		)	
11	042	2A		*	
12	043	2B		+	
13	044	2C		,	
14	045	2D		-	
15	046	2E		.	
16	047	2F		/	
17	048	30		0	
18	049	31		1	
19	050	32		2	
20	051	33		3	
21	052	34		4	
22	053	35		5	
23	054	36		6	
24	055	37		7	
25	056	38		8	
26	057	39		9	
27	058	3A		:	
28	059	3B		;	
29	060	3C		<	
30	061	3D		=	
31	062	3E		>	
32	063	3F		?	
33	064	40		@	

Graphics y,x	Decimal	Hex	Code Key	Character	Cursor Commds
34	065	41		A	
35	066	42		B	
36	067	43		C	
37	068	44		D	
38	069	45		E	
39	070	46		F	
40	071	47		G	
41	072	48		H	
42	073	49		I	
43	074	4A		J	
44	075	4B		K	
45	076	4C		L	
46	077	4D		M	
47	078	4E		N	
48	079	4F		O	
49	080	50		P	
50	081	51		Q	
51	082	52		R	
52	083	53		S	
53	084	54		T	
54	085	55		U	
55	086	56		V	
56	087	57		W	
57	088	58		X	
58	089	59		Y	
59	090	5A		Z	
60	091	5B		[	
61	092	5C		\	
62	093	5D		]	
63	094	5E		^	
64	095	5F		←	
65	096	60		'	
66	097	61		a	
67	098	62		b	
68	099	63		c	
69	100	64		d	
70	101	65		e	
71	102	66		f	
72	103	67		g	
73	104	68		h	
74	105	69		i	
75	106	6A		j	
76	107	6B		k	
77	108	6C		l	
78	109	6D		m	
79	110	6E		n	
80	111	6F		o	
	112	70		p	
	113	71		q	
	114	72		r	
	115	73		s	
	116	74		t	
	117	75		u	
	118	76		v	
	119	77		w	
	120	78		x	
	121	79		y	
	122	7A		z	
	123	7B		{	
	124	7C		|	
	125	7D		}	
	126	7E		~	
	127	7F		DEL	

CPT®



CPT CORPORATION • 8100 MITCHELL ROAD, P.O. BOX 295 • MINNEAPOLIS, MN 55440 U.S.A. • TELEPHONE (612) 937-8000, TELEX 29-0775