

CHAPTER 6  
ASSEMBLER PROGRAMMING UNDER ETOS

6.1 INTRODUCTION

This chapter contains information about assembly language programming. Coding is performed directly at the machine level. To develop assembler programs, you must utilize one of the assemblers distributed with the system (PAL8, SABR, RALF or MACREL). Your program must be developed under OS/8, but it may be executed under COS as a core image file. All "V" extension programs under COS (e.g., SYSGET, DFDIR) were developed in this manner. If you write a program which will operate under COS, make sure that you do not utilize an operating system dependent feature of OS/8, such as the user service routines.

All instructions, which may be used in assembly language coding, are listed in this chapter. Most instructions work identically (from the program's standpoint) to the way that they do on a stand-alone PDP-8. If you utilize a real time task which clears user mode, all input/output transfers (IOTs) work exactly as they do on a stand-alone PDP8. Due to the close correlation between ETOS IOT's and stand-alone PDP8 IOTs, the following manuals may be utilized for additional information: Introduction to Programming, the OS/8 Handbook and the OS/8 Software Support Manual. All of these manuals are reference books only. To learn assembler, there is an excellent student workbook and teacher's edition set called PDP-8 Programming, written by R. W. Southern. These books may be ordered from the Algonquin College Bookstore, 1385 Woodroffe Avenue, Ontario, Canada K26 1V8.

## 6.2 TERMINAL IOTs

### 6.2.1 KCF 6030 NOP

This instruction is ignored.

### 6.2.2 KSF 6031 SKIP ON NON-EMPTY INPUT BUFFER

If there are characters in the input buffer, skip the next instruction. If the input buffer is empty, and the next instruction is JMP `.-1`, the program ceases running until a break character has been entered.

### 6.2.3 KCC 6032 CLEAR AC AND ADVANCE IF CHARACTER READ

Clear the AC and advance the input buffer pointer past the current character, if the current character has already been read by a previous KRS. If the current character has not yet been read, the pointer is not advanced.

### 6.2.4 KCL 6033 CLEAR KEYBOARD BUFFER

Clear the keyboard buffer. All characters in the buffer are discarded.

### 6.2.5 KRS 6034 OR INPUT CHARACTER WITH AC

Inclusively OR the current input character with the AC; do not advance the buffer pointer beyond the current character. If no characters are in the input buffer, return to the program with the AC unchanged. Successive KRS's with no intervening KCC's will OR the same character.

### 6.2.6 KRB 6036 READ KEYBOARD BUFFER

Jam transfer the next input character to the AC, and advance the buffer pointer beyond that character. If no characters are in the input buffer, put the program in input and wait until a break character is entered. Note that when one uses KRB to read characters, "KSF; JMP .-1" is unnecessary.

### 6.2.7 GETSTAT 6037 GET STATUS WORDS

Read the status words KSTAT, BREAK,  $\hat{C}\hat{C}$  restart address, and  $\hat{P}$  restart address into the Instruction Field starting at the location whose address is in the AC. The AC is cleared upon return. For more information on the meaning of the status words, KSTAT and BREAK, see 3.7.6 and 3.7.5. The  $\hat{C}\hat{C}$  restart address is the location in field 0 to dispatch to, if you type  $\hat{C}\hat{C}$ . The  $\hat{P}$  restart address is the location in field 0 to dispatch to, if you type  $\hat{P}$ . After saving the current status with this IOT, you may wish to change the status via the IOT SETSTAT (see 6.2.14). A sample calling sequence is contained in Figure 6-1.

Figure 6-1  
Sample Use of GETSTAT

```
GETSTAT=6037
CLA      CLL
TAD      (ARGLST      /POINT TO ARGLST
GETSTAT      /SAVE THE STATUS WORDS
.
.
.
ARGLST, 0000      /SAVE KSTAT WORD HERE
          0000      /SAVE BREAK MASK HERE
          0000      /SAVE ADDRESS TO GO TO ON DOUBLE  $\hat{C}$ 
          0000      /SAVE ADDRESS TO GO TO ON  $\hat{P}$ 
```

### 6.2.8 TSF 6041 SKIP

Skip the next instruction unconditionally.

#### 6.2.9 TCF 6042 NOP

This instruction is ignored.

#### 6.2.10 TCL 6043 CLEAR OUTPUT BUFFER

Initialize the teleprinter and immediately clear the output buffer. This is useful for OS/8, for instance when ^C is entered, so that output, which may be up to 200 characters ahead, ceases immediately.

#### 6.2.11 TPC 6044 LOAD PRINTER BUFFER

Put the character in the AC into the output buffer for printing. If the output buffer is full, put the program in input/output, wait (stop it running) until the printer is within a system defined number of characters of emptying the buffer (see 1.2). Restarting the job before output finishes helps insure that output will be continuous. The AC is not cleared.

#### 6.2.12 TSK 6045 SKIP ON TERMINAL FLAG

Skip the next instruction unconditionally.

#### 6.2.13 TLS 6046 LOAD BUFFER PRINTER

Exactly the same as TPC.

#### 6.2.14 SETSTAT 6047 SET STATUS WORDS

Set the status words KSTAT, BREAK, ^C^C restart address, and ^P restart address using the information from the Instruction Field, starting at the location whose address is in the AC. The AC is cleared upon return. For more information on the status words KSTAT and BREAK, see 3.7.6 and 3.7.5. The ^C^C restart address

is the location in field 0 to dispatch to, if you type ^C^C. The ^P restart address is the location in field 0 to dispatch to if you type ^P. Before executing this IOT, you may wish to save the current status via the IOT GETSTAT (see 6.2.7). A sample calling sequence is contained in Figure 6-2.

Figure 6-2  
Sample Use of SETSTAT

```

KSTAT=WWWW
BREAK=XXXX
CTCRES=YYYY
CTPRES=ZZZZ
SETSTAT=6047
CLA      CLL
TAD      (ARGLST      /POINTER TO ARGLST
SETSTAT  /NEW STATUS WORDS
.
.
.
ARGLST, KSTAT      /ETOS STATUS
          BREAK     /BREAK MASK
          CTCRES    /ADDRESS IN FIELD 0 TO GO TO ON
                  /DOUBLE ^C
          CTPRES    /ADDRESS IN FIELD 0 TO GO TO ON ^P

```

### 6.3 EXTENDED MEMORY IOTs

All references to "data field" and "instruction field" refer to the virtual data and instruction fields. There is no way to determine which physical fields your program is using.

#### 6.3.1 CDF 62n1 CHANGE DATA FIELD

Change to Data Field n. This works from the program's standpoint exactly the way it does on a stand-alone PDP-8. However, a CDF to a non-existent field results in a CDF 0. You should be sure you have allocated enough memory or unpredictable results will occur.

### 6.3.2 CIF 62n2 CHANGE INSTRUCTION FIELD

Change to Instruction Field n. Like CDF, this appears to the program the same as CIF on a stand-alone PDP-8. However, execution of a JMP or JMS instruction after a CIF to a non-existent field results in the error message "?JMP/JMS TO NONEXISTENT FIELD".

### 6.3.3 CIF CDF 62n3 CHANGE DATA AND INSTRUCTION FIELDS

Change to Data and Instruction Field n. This is the microcoding of the previous two instructions.

### 6.3.4 RDF 6214 READ DATA FIELD

Inclusively OR the current data field with AC bits 6-8. This is the same affect as on a stand-alone PDP-8.

### 6.3.5 RIF 6224 READ INSTRUCTION FIELD

Inclusively OR the current instruction field with AC bits 6-8. This is the same affect as on a stand-alone PDP-8.

## 6.4 TIME SHARING IOTs

### SKETOS 6107 Skip on ETOS

Skip the next instruction if the program is running under ETOS/8 and change the 6107 to a 7410 (SKP) so only one IOT trap will occur. The AC is left unchanged by the execution of this IOT. On a stand-alone PDP-8 this instruction is a NOP, allowing the user to write programs that can be executed under both ETOS and stand-alone PDP-8 without modification. A sample use of the SKETOS IOT is contained in Figure 6-3.

Figure 6-3  
Sample Use of SKETOS

```
SKETOS=6107
CLA CLL
SKETOS
JMP STANDAL          /Jump to stand-alone routines
ETOS,                /ETOS routines
```

## 6.5 CARD READER IOTs

The ETOS card reader processes in a completely different manner from the stand-alone handler. The basic modes are similar but the ETOS handler is record oriented whereas the stand-alone handler is character oriented.

The following IOTs read a card into a buffer in the user's virtual data field. The IOT should be executed with the AC = pointer to the card reader buffer. If the card reader is not selected, the system returns to the next instruction. If the card reader is selected, the next instruction is skipped.

### 6.5.1 RCRA 6632 READ CARD IN ALPHANUMERIC MODE

Read a card into the card reader buffer in alphanumeric mode. Each column is decoded into a six-bit BCD representation and is transferred into the least six significant bits of its respective word.

### 6.5.2 RCRP 6633 READ CARD IN ALPHANUMERIC PACKED MODE

Read a card into the card reader buffer in a newly defined alphanumeric packed mode. As in the alphanumeric mode, each column is decoded into a six-bit BCD representation. However, in packed mode two columns are packed per word, with the columns in the order that they were read. Therefore, the first column of the card is deposited into the most significant six bits of the

first location of the buffer; the second column is deposited into the least significant six bits of the first location of the buffer; the third column is deposited into the most significant bits of the second location of the buffer; etc.

#### 6.5.3 RCRB 6634 READ CARD IN BINARY MODE

Read a card into the card reader buffer in binary mode. Each column is stored in one word, with the data transferred directly from the rows of the card to the bits in the word. The top row of the card (row 12) is deposited into bit 0 and the bottom row (row 9) is deposited into bit 11.

#### 6.5.4 RCRC 6636 READ CARD IN COMPRESSED MODE

Read a card into the card reader buffer in compressed alphanumeric mode. Each column is stored in one word. Rows 9, 12, 11, 0, and 8 are transferred directly to bit 4, bit 5, bit 6, bit 7, and bit 8, respectively, while rows 1 through 7 are decoded into a three-bit BCD representation in bits 9, 10, and 11. If a double punch occurs in rows 1 through 7, bit 0 is set to a one.

After you have issued one of the previous card reader IOTs in order to read a card, you must test to see if the card has been read in. The following IOT is used to test card completion.

#### 6.5.5 RCCR 6630 SKIP ON CARD READER COMPLETION

Skip if card done. The AC may be any value on entry. On exit, the AC is set equal to minus the number of columns in the card just read. If an error occurs, the AC is set equal to 0.

A sample use of the ETOS card reader IOTs is contained in Figure 6-4.



Figure 6-4  
Sample Use of ETOS Card Reader IOTs

```

RCRA=6632
RCCR=6630
TAD      (CDRBUF      /POINTER TO CARD READER BUFFER
RCRA     /READ CARD IN ALPHANUMERIC MODE
JMP      NOTRDY      /CARD READER NOT SELECTED
RCCR     /CARD DONE?
JMP      .-1         /NO, WAIT FOR IT
SNA     /CARD READ, ERROR?
JMP      CDERR       /YES, PRINT ERROR MESSAGE
DCA     CDCNT       /NO, SAVE COLUMN COUNT
.
.
.
CDRBUF, ZBLOCK 120   /CARD READER BUFFER
CDCNT, 0

```

#### 6.5.6 ILLEGAL CARD READER IOTS

All other stand-alone card reader IOTs are defined as illegal. Since execution of one of these IOTs will result in a virtual machine halt with the program counter printed out, debugging of a stand-alone handler should be simplified. A list of these illegal IOTs is contained in Table 6-1.

Table 6-1  
Illegal Card Reader IOTs

<u>Stand-alone Mnemonic</u>	<u>IOT</u>
RCSF	6631
RCNO	6635
RCNI	6637
RCSD	6671
RCSE	6672
RCRD	6674
RCSI	6675
RCTF	6677

## 6.6 LINE PRINTER IOTs

### 6.6.1 LSF 6661 SKIP ON LINE PRINTER FLAG

Skips the next instruction unconditionally.

### 6.6.2 LLS 6666 PRINT CHARACTER

Print character on the line printer. Program goes to sleep if output buffer is full.

### 6.6.3 Miscellaneous Line Printer IOTs

6662-6665,6667 are NOP's

## 6.7 SYSTEM FUNCTIONS

SYSCAL 6200 SYSTEM FUNCTIONS  
Extended system functions.

### SYSCAL Calling Parameters

MQ = Function number

AC = Argument or argument pointer

DF = Data field for transmission of ETOS information

A list of system functions is contained in Table 6-2.

Table 6-2  
System Functions

<u>Mnemonic</u>	<u>Function Number</u>
SSR	0
TOD	1
W100MS	2
WLMIN	3
WAIT	10
DATAPK	11
VERSION	12
TXTSET	13
LOGOUT	14
JOBNUM	15
XSCALE	16
ENDRUM	20
CORE	22

System functions 4, 5, 6, 7, 17, 20, 21 and 23 are privileged functions discussed in the System Manager's Guide. System functions 24-77 are invalid and generate an "?ILLEGAL IOT" error message. All valid non-privileged system functions are listed and documented in section 6.7.

### 6.7.1 SSR (Function 0)

Set virtual switch register to value contained in the AC. This value can then be read with an OSR. The AC is cleared on return. A sample calling sequence is contained in Figure 6-5.

Figure 6-5  
Sample Use of SSR

```
SYSCAL=6200
SSW=0           /SET SWITCH REGISTER IS FUNCTION 0
VALUE=XXXX     /DESIRED VALUE FOR SR
CLA CLL
TAD            (SSW
MQL           /MQ = FUNCTION NUMBER
TAD            (VALUE
SYSCAL        /NEW VALUE OF SWITCH REGISTER
/AC = 0 UPON RETURN
```

### 6.7.2 TOD (Function 1)

Return the time of day. The AC points to a four-word block in the user's data field before the execution of SYSCAL. AC = (address of block) minus one. The AC is cleared upon return. The block is of the form

Word 1 = minus number of ticks to next minute (1 tick = 100 (base 10) milliseconds).

Word 2 = minus number of minutes to next day (number of minutes-1440 (base 10)).

Word 3 = minus number of days to next year (number of days-365 (base 10)). December 31st = -1.

Word 4 = year (A.D.).

A sample calling sequence is contained in Figure 6-6.

Figure 6-6  
Sample Use of TOD

```

SYSCAL=6200
TOD=1                /TOD IS FUNCTION 1
BUFFLD=Y            /FIELD CONTAINING T.O.D. BUFFER
FLDCHG=BUFFLD^10
.
.
.
CLA CLL
TAD      (TOD
MQL                      /MQ = FUNCTION NUMBER
TAD      (TODBUF-1      /AC = BUFFER POINTER
CDF FLDCHG              /BUFFER IS IN DATA FIELD
SYSCAL                  /GET TIME OF DAY
/ AC = 0 UPON RETURN
.
.
.
FIELD BUFFLD
TODBUF,  0              /- MILLISECONDS TO NEXT MINUTE
          0              /- MINUTES TO NEXT DAY
          0              /- DAYS TO NEXT YEAR
          0              /+ YEAR A.D.

```

### 6.7.3 W100MS (Function 2)

Put job to sleep for n times 100 milliseconds. AC = -n an entry. AC is cleared upon return. Users who will be idle for a period of time may put themselves to sleep for a prescribed period. At the end of this period, the job will be awakened and begin processing. This IOT could be used for a job waiting to use a peripheral which is currently assigned to another user. Due to system overhead, the sleep time is not exact. It might be off by as much as three hundred milliseconds.

A sample calling sequence is contained in Figure 6-7.

Figure 6-7  
Sample Use of W100MS

```

SYSCAL=6200
W100MS=2            /WAIT ON 100 M.S. TIMER IS FUNCTION 2
COUNT=XXXX        /# INTERVALS TO SLEEP
CLA CLL
TAD      (W100MS
MQL                      /MQ = FUNCTION NUMBER
TAD      (-COUNT      /AC = - TICKS (100 M.S.) TO SLEEP FOR
SYSCAL                  /GO TO SLEEP
/ AC = 0 UPON RETURN

```

#### 6.7.4 WLMIN (Function 3)

Put job to sleep until the nth system minute. AC = -n an entry. The AC is cleared upon return. This function is analogous to function 2(WL00MS).

#### 6.7.5 WAIT (Function 10)

Wait until status bit cleared. The job goes to sleep until events occur that clear those status bits that are set. The job must be sure that such events will occur or else it may go to sleep forever. AC = word with bits set corresponding to forms of I/O wait. All of these bits must be cleared before the job will wake up. A list of the WAIT bits and their meanings is contained in Table 6-3.

Table 6-3  
WAIT Bits

<u>Bit</u>	<u>Meaning</u>
0	Waiting in SCALE mode
1	Terminal input wait
2	Terminal output wait
3	Line printer wait
4	Disk file input/output
5	Card reader wait
6	Waiting on 100ms timer
7	Waiting on 1 minute timer
8	Waiting for DMON
9	Unused
10	Unused
11	Job not resident

A sample calling sequence is contained in Figure 6-8.

Figure 6-8  
Sample Use of WAIT

```
SYSCAL=6200
WAIT=10           /WAIT IS FUNCTION 10
MASK=XXXX        /ONE BIT SET FOR STATUS
CLA CLL
TAD      (WAIT
MQL
TAD      (MASK
SYSCAL          /SLEEP UNTIL STATUS BIT CLEARED
/ RETURNS HERE WHEN STATUS BITS ARE CLEARED
/ AC = 0 UPON RETURN
```

### 6.7.6 DATAPK (Function 11)

Fetch data word. This IOT returns various information about a specific job. The high-order 6 bits of the AC contain the job number. If the job number is 0, the current job is assumed. The low order 6 bits contain the entry number of the desired information. This offset must be an octal number from 0 to 37. If the number is 40 or greater, it is interpreted modulo 37. On return, the data word is in the MQ. The AC = 0 unless the job number specified (JOB) was greater than the highest job number available (JOBMAX). In this case, AC = JOB-JOBMAX.

A list of the entry numbers and their meaning is provided below.

- Entry 0 = Pointer to the user map location in field 1. The high order 9 bits of the user map contain the swap track divided by 2. The low order 3 bits contain the field (0 if nonresident).
- Entry 1 = User status register. The layout of this register is given in function 10 (wait for status bit).
- Entry 2 = Pointer to the channel control block pointers.
- Entry 3 = Low order run time in system ticks (100ms intervals). The run-time for a job is stored in a double precision word, consisting of entries 3 and 4.
- Entry 4 = High order run time.
- Entry 5 = Privilege word. The format of this word is contained in Table 6-4.

Table 6-4  
PRIVilege Word Format

<u>Bits</u>	<u>Meaning</u>
0	0 if not privileged, 1 if privileged.
1	0 if not temporarily privileged, 1 if temporarily privileged.
2	Unused.
3 - 5	Core assigned to this job.
6 - 8	Priority in the run queue (0 = highest, 7 = lowest).
9 - 11	Offset into quantum table.

This word is set to 0040 upon LOGIN. It may be reset by a privileged user.

- Entry 6 = Nonzero if CIF pending.
- Entry 7 = Address of level 2 routine to run before starting job. Zero entry means there is no such routine.
- Entry 10 = User flag.
- Entry 11 = Unused.
- Entry 12 = Virtual program counter (PC).
- Entry 13 = Virtual accumulator (AC).
- Entry 14 = Virtual multiplier quotient (MQ).
- Entry 15 = EAE register. The format of this register is contained in Table 6-5.

Table 6-5  
EAE Register Format

Bits	Meaning
0 - 10	Virtual step counter.
11	EAE mode. 0 if Mode A. 1 if Mode B.

- Entry 16 = Console number.
- Entry 17 = Virtual switch register (SR). This register may be set with SYSCAL function 0 (SSR). It may be read with the OSR instruction (7404).
- Entry 20 = Account number. The project number is contained in the higher order six bits. The programmer number is contained in the lower order six bits.
- Entry 21 - 24 = Time of day when LOGIN was issued. These four words are in the same format as function 1 (time of day). Four IOT's must be executed to retrieve these words, as opposed to one for the current time.



Entry 25 = Pointer to keyboard device data block (DDB). The format of the data block is as follows.

Word 0= Keyboard status. The format of this register is specified in the section on the KSTAT command (3.7.6).

Word 1= Job number assigned to the DDB.

Word 2= Last character read.

Word 3= Saved break mask. The format of this register is specified in the section on the BREAK Command (3.7.5).

Word 4= DDB fill pointer.

Word 5= DDB empty pointer.

Word 6= Number of characters in the input buffer.

Word 7= Active break mask. The format of this register is specified in the section on the BREAK Command (3.7.5).

Entry 26 = Pointer to teleprinter device data block. The data block has the following format.

Word 0= Teleprinter status. The format of this word is contained in Table 6-6.

Table 6-6  
Teleprinter Status Format

<u>Bits</u>	<u>Meaning</u>
0	CTRL/O activated
1	XOFF activated
2-5	Unused
6-11	Jam output pointer

Word 1= Job number assigned to data block.

Word 2= ^C^C restart address. This address is specified via the SETSTAT IOT (see 6.2.14).

Word 3= CTRL/P restart address. This address is specified via the SETSTAT IOT (see 6.2.14).

Word 4= DDB fill pointer.

Word 5= DDB empty pointer.

Word 6= Number of characters in the output buffer.

Word 7= Unused.

Entry 27 = Work storage temporary number 1.

Entry 30 = Work storage temporary number 2.

Entry 31 = Protection code of UFD. This protection code is of the form specified in the ENTER Command (see 7.2.3 and 7.4.5).

Entry 32 = Reserved for future use.

Entry 33 - 37 = Program currently running. If COS is running, the program name is "COS" and the extension is unused. If the OS/8 keyboard monitor is running, the program name is "KBDMON" and the extension is "OS". If an OS/8 or ETOS program is running, the first three words are used for the program name, the fourth word is used for the extension, and the fifth word is unused. This text area is set via function 13 (TXTSET).

A sample calling sequence is contained in Figure 6-9.

Figure 6-9  
Sample Use of DATAPK

```
SYSCAL=6200
DATAPK=11           /DATAPK IS FUNCTION 11
ENTRY=XX           /ENTRY NUMBER
JOB=XX             /JOB NUMBER
CLA CLL
TAD      (DATAPK
MQL
TAD      (JOB
BSW
TAD      (ENTRY      /AC = JOB * 100 + ENTRY
SYSCAL
MQA
/DATA RETURNED NOW IN AC
```

### 6.7.7 VERSION (Function 12)

Return monitor version number. The AC may be any value on entry. The version of the monitor is returned in the AC in packed 6-bit ASCII. A sample calling sequence is contained in Figure 6-10.

Figure 6-10  
Sample Use of VERSION

```
SYSCAL=6200  
VERSION=12  
TAD      (VERSION      /VERSION IS FUNCTION 12  
MQL  
SYSCAL  
/ AC = VERSION OF THE SYSTEM  
/ E.G. 6502 FOR V5B)
```

### 6.7.8 TXTSET (Function 13)

Set text area. This function deposits a five word block into the user's data area. This area is then read with function 11, offset 33-37. Currently, only the first four words of this area are utilized. OS/8 sets up this area upon execution of a RUN, R, CCL, or ODT command. The COS bootstrap sets it to "COS". This area is then read by SYSTAT to print the current program. The AC points to the five-word block in the user's data field. AC = (address of block) minus one. The AC is cleared upon return.

A sample calling sequence is contained in Figure 6-11.

Figure 6-11  
Sample Use of TXTSET

```
SYSCAL=6200
TXTSET=13           /TXTSET IS FUNCTION 13
BUFFLD=Y           /DATA FIELD CONTAINING ARGUMENT BLOCK
FLDCHG=Y^10        /ARGUMENT FOR CDF
.
.
.
CLA CLL
TAD      (TXTSET
MQL
TAD      (BUFF-1     /AC = POINTER TO DATA BLOCK
CDF      FLDCHG      /ARGBLK IS IN DATA FIELD
SYSCAL           /0
/ AC = 0 UPON RETURN
.
.
.
FIELD BUFFLD
BUFF,      TEXT "FILENMEX"   /FILE NAME
```

### 6.7.9 LOGOUT (Function 14)

Log the current user off of the ETOS system. The AC is ignored. A sample calling sequence is contained in Figure 6-12.

Figure 6-12  
Sample Use of LOGOUT

```
SYSCAL=6200
LOGOUT=14           /LOGOUT IS FUNCTION 14
TAD      (LOGOUT
MQL
SYSCAL
/USER NOW LOGGED OUT
```

### 6.7.10 JOBNUM (Function 15)

Return current job number. The AC may be any value. Upon return, the AC = current job number. A sample calling sequence is contained in Figure 6-13.

Figure 6-13  
Sample Use of JOBNUM

```
SYSCAL=6200
JOBNUM=15                /JOBNUM IS FUNCTION 15
TAD      (JOBNUM
MQL
SYSCAL                /GET JOB NUMBER
/ AC CONTAINS CURRENT JOB NUMBER
.
.
```

### 6.7.11 XSCALE (Function 16)

Send a text string to SCALE, which is executed as a command. The text string must be in 8-bit ASCII, packed one character per word, and be terminated by a carriage return (ASCII 215). The maximum command length is 72 characters but the semicolon construction is not permitted. The AC contains a pointer to the text block in the user's data field. AC = (address of block) minus one. The AC is normally cleared upon return. If the ASSIGN command is specified and the device is busy, the AC contains the number of the job who owns the device. A sample calling sequence is contained in Figure 6-14.

Figure 6-14  
Sample Use of XSCALE

```
SYSCAL=6200
XSCALE=16                /XSCALE IS FUNCTION 16
CMDFLD=Y                /FIELD WHERE TEXT STRING IS STORED
FLDCHG=CMDFLD^10       /ARGUMENT FOR CDF
.
.
.
CLA CLL
TAD      (XSCALE
MQL
TAD      (CMD-1          /AC = (TEXT POINTER) - 1
CDF      FLDCHG
SYSCAL                /SEND THE COMMAND TO SCALE
/ AC = 0 ON RETURN UNLESS DEVICE ASSIGNED
.
.
FIELD CMDFLD
CMD,      "W;"H";"E;"R;"E;215
```

### 6.7.12 ENDRUN (Function 20)

Each job that is active in the system is scheduled for a particular period of time called a quantum. If the quantum has expired or a request for input/output to a device which is not available is issued, the job is dismissed and another job is scheduled. The ENDRUN IOT dismisses the job even if one of these two events have not occurred. The reason for executing this IOT is if you want to insure that you have an entire quantum for a particular process. If you were sampling a device which required a certain period of time to respond, you might execute this IOT immediately before you began your sampling. The AC is ignored when calling this instruction. A sample calling sequence is contained in Figure 6-15.

Figure 6-15  
Sample Use of ENDRUN

```
SYSCAL=6200
ENDRUN=20                /ENDRUN IS FUNCTION 20
CLA CLL
TAD      (ENDRUN
MQL
SYSCAL
```

### 6.7.13 CORE (Function 22)

Set number of memory fields assigned to the virtual machine (VM). This function does not modify existing memory fields if they are not deallocated. The CORE function can therefore be used to dynamically adjust the size of a VM. At entry, the AC contains the number of 4K memory fields to be allocated to the VM -1 in bits 9-11. AC bits 0-8 currently are ignored, but this may change in the future. Upon return, a zero AC indicates successful completion. A nonzero AC = minus number of fields that could not be allocated. A sample calling sequence is contained in Figure 6-16.

Figure 6-16  
Sample Use of CORE

```
SYSCAL=6200
CORE=22                /CORE IS FUNCTION 22
VMSIZE=X              /VM WILL HAVE THIS
                     /MANY MEMORY FIELDS

CLA CLL
TAD CORE
MQL
TAD      (VMSIZE-1    /AC = HIIGHEST FIELD
SYSCAL
SZA      /CHECK ERROR CODE
JMP ERROR /ERROR ROUTINE
```

If you utilize this function in an OS/8 program, you must also set the OS/8 software core size. Some OS/8 programs determine how much virtual memory is assigned to the user. All available memory can then be utilized by the program. The MEMORY command (see page 12, OS/8 Handbook Update) sets bits 6-8 in location 7777, field 0 to the highest field available for the program's use. This field value must be less than or equal to the amount of virtual memory assigned to the job. When OS/8 is BOOTed, bits 6-8 in 07777 (the software core size) are set to the amount of memory assigned to the account. If the user was assigned 8K memory, bits 6-8 are set to 001.

If you change the amount of memory assigned with the CORE system call, you must change the software core size to the highest field available. One way to accomplish this is to reboot the virtual machine by typing ^VS;BOOT<RET>. If you do not reboot OS/8, you must set the software core size directly. Do not modify any other bits in 07777. These bit are used for BATCH flags and the extended date. The routine contained in Figure 6-17 can be used to change a user's core allocation to 16K.

Figure 6-17  
OS/8 Core Usage

```

SYSCAL=6200
CORE=22
TAD      (CORE          /CORE IS FUNCTION NUMBER 22
MQL
TAD      (3             /HIGHEST FIELD USED
SYSCAL
SZA      /CHECK ERROR CODE IN AC
JMP ERROR /ERROR ROUTINE
/THE REST OF THE ROUTINE ONLY REQUIRED UNDER OS/8
/IF USER DOES NOT REBOOT VIRTUAL MACHINE
CDF 0
TAD I    (7777         /GET CONTENTS OF 07777
AND      (7707         /STRIP BITS 6-8
TAD      (0030         /SET HIGHEST FIELD TO 3
DCA I    (7777         /RESTORE LOCATION 07777

```

If you execute the CORE system call while running under BATCH, there are additional problems to consider. When the BATCH monitor is read in each time the keyboard monitor is called, it is read into the highest field assigned to the account. If the memory allocated is changed via the CORE SYSCAL, BATCH will not be aware of the memory change. If the memory size is reduced, BATCH will attempt to read itself into a non-existent field. If the memory size is extended, BATCH could overlay your program. The only way to circumvent this problem is to restrict a user program to use only memory which BATCH is not utilizing. The routine contained in Figure 6-18 illustrates a user requesting 16K.

Figure 6-18  
Batch Core Usage

```

SYSCAL=6200
CORE=22
CLA CLL
TAD I      (7777           /OS/8 SOFTWARE CORE SIZE
AND        (0070         /ISOLATE
TAD        (-30
SMA CLA                    /WANT AT LEAST 16K
JMP GOT16K
TAD I      (7777           /CHECK BATCH
RTL
SZL CLA                    /BATCH BIT IN LINK
JMP I      (7600         /EXIT IF NON-ZERO
TAD        (CORE
MQL
TAD        (3
SYSCAL                    /GRT 16K
SZA                    /CHECK ERROR CODE
JMP        ERROR        /ERROR ROUTINE

GOT16K, TAD I (7777       /ONE MORE PROBLEM
RTL                    /IF 16K, I CAN USE
SNL CLA                /ONLY 14.5K
JMP TELLOS
TAD I      (7777       /CHECK CORE ONCE MORE
AND        (70
TAD        (-30
SZA CLA                /16K=0-, 20=+
JMP OKGO              /20K +, WE ONLY USE 16, OK

```



## 6.8 MISCELLANEOUS IOTs

### 6.8.1 SGT 6606 SKIP IF GREATER THAN FLAG SET

Skip the next instruction if the EAE's GT flag is set.

### 6.8.2 HLT 7402 HALT

Stop the VM. Print the message "?%VM HLT" and put the TTY in SCALE mode. Microcoding HLT with other instructions is allowed.

### 6.8.3 OSR 7404 OR SWITCH REGISTER WITH AC

OR the contents of the virtual Switch Register with the AC. The virtual switch register is set with SYSCAL function 0 (see 6.7.1) or it is set with the SR SCALE command (see 3.74.).

## 6.9 UNIMPLEMENTED IOTs

All IOT's which have not been listed in this chapter are not supported and should be considered illegal. If an unsupported IOT is attempted, the message "ILLEGAL IOT" is displayed, and the terminal is placed in SCALE mode. This error message is also printed if you attempt to use a device which has not been ASSIGNED (see 3.10).



## CHAPTER 7

### ETOS CHANNELS AND ASSEMBLER PROGRAMMING OF CHANNELS

#### 7.1 INTRODUCTION

Each job under ETOS has eight channels associated with it. These channels are used to access ETOS files. These files may be system files, user account files, private disk files or public disk files. At LOGIN time, three channels are assigned by the system. Channel 0 is the virtual OS/8 system area, SYS:(DK0:[0,2]OS8.RTS). This area is read-only and is accessible to all users. Channel 1 is assigned to scratch blocks (DK0:[0,3]JOBXX.SBK). The OS/8 system scratch blocks and a temporary file for OS/8 BATCH are written on this channel. Channel 2 is assigned to the user's OS/8 account(OS8DISK.DSK). Channels 3 and 4 are available to the user as OS/8 device handlers. The system manager may also activate channels 5 through 7. These handlers are called "CHNx:" where "x" is the channel number.

Initiating the COSBO program to run COS causes channel 0 to be closed. The ETOS file COSDISK.RTS in the user's account is looked up on channel 0 and the file is BOOTED. This results in COS being started. Channel 1 is not closed so that the OS/8 scratch blocks can be left open and write enabled when switching between OS/8 and COS. Channel 2 is not closed and thus allows the user to transfer data from the OS/8 account to the COS account. Under COS, ETOS channels are called "DKx", where "x" is the channel number. Therefore, the user could transfer files from the OS/8 account area to the COS area by transferring from DK2 to DK0.

For an RK05 installation, the system pack is mounted in physical drive DK0 and non-system packs are mounted in physical drives DK1, DK2 and DK3. For a System Industries installation, the

system pack is the fixed pack in port 0 (DK0). The removable pack in port 1 is DK1. The fixed disk in port 2 is DK2. The removable pack in port 1 is DK3. These physical device names are used in the SCALE commands ASSIGN, LOOKUP, ENTER, MOUNT and DISMOUNT and in the ACCNT program. The logical channels (CHNx: for OS/8, DKx: for COS) are not necessarily related to physical drive numbers. The user's private account storage areas may be stored on the ETOS system disk as on any public disk. For example, if an account area were located on a public pack in physical DK1, it could be accessed as CHN2: or DSK: under OS/8 or DK0 under COS.

ETOS files are files accessed by Channel I/O SCALE commands (see 7.2) or Channel IOTs (see 7.4). ETOS files may be programs such as LOGIN.SAV. They may also be user accounts. Each account is actually an ETOS file with a size, protection, etc. assigned by the system manager via the ACCNT program, which uses Channel IOTs. ETOS files differ from OS/8 or COS files in that they are handled by a nonresident ETOS disk monitor (DMON). DMON also maintains the ETOS directory, allowing for multi-user access. OS/8 and COS are ETOS files. OS/8 files and COS files are not actually seen by ETOS. It appears to the system that an ETOS program, OS/8 or COS, is performing input/output operations on another ETOS file in the user's account disk area. It is not aware of considerations such as the OS/8 or COS directory structure. It is this flexibility that permits multiple operating systems to be implemented under ETOS.

## 7.2 CHANNEL INPUT/OUTPUT COMMANDS

SCALE provides the facility to manipulate ETOS files from the terminal. These commands correspond exactly to their corresponding channel IOTs in argument structure, restrictions, etc. Refer to section 7.4 to see how these IOTs work. Rather than specifying arguments in argument blocks within user programs, the SCALE commands accept arguments on the command delete line. The two exceptions to the exact correspondence between channel I/O IOTs and commands occur in the READ and WRITE functions. The IOT transmits data to and from the user's data field. The corresponding SCALE command allows the address specification to be five characters, where the first digit represents the data transmission field.

A list of CHANIO SCALE commands, with the form of the arguments and possible error messages, follows. These error messages are in the form "n? FUNCTION FAILURE" where "n" corresponds to the error code returned in the AC by the equivalent IOTS. Note: when you LOGOUT, all channels are closed.

For channel I/O commands, the defaults, which are assumed for parameters omitted by the user are contained in Table 7-1.

Table 7-1  
Channel I/O Defaults

<u>Item</u>	<u>Default</u>
device	SYS
[p,pn]	user's account
<protection>	<0>
size	0

The block numbers referred to in the SCALE commands are octal numbers in the range 0 to 37777777. However, an RK05 disk contains only 14152 octal blocks. For an RK05, the high order

number of blocks must be 0 or 1. A System Industries 30/40 fixed or removable pack contains 30324 octal blocks. For a System Industries disk, the high order number of blocks must be 0, 1, 2 or 3. The capability of 37777777 blocks is included only for possible implementation of a bigger disk under ETOS.

7.2.1 REA[D] channel,block,minus word count,buffer address

Read from a disk file associated with the specified channel into the data field specified by the first character of the buffer address and location specified by "buffer address". If you attempt to READ from a channel which is not LOOKed UP on an ETOS file, the message "?CHANNEL NOT OPEN" is printed. The READ starts at the specified "block", which is an octal number in the range 0 to 37777777. The word count is an octal number, which is rounded up to the nearest multiple of 400 (base 8) or 256 (base 10). See 7.4.1 for further details. A list of possible READ errors is contained in Table 7-2.

Table 7-2  
SCALE READ Errors

<u>Error</u>	<u>Meaning</u>
7 minus	Read beyond end of file Hard error

For example, to READ blocks 0 through 4 from the file LOOKed UP on channel 3 into locations 4000 through 5777 in field 1, enter the command specified in Figure 7-1.

Figure 7-1  
Sample SCALE READ

```
!READ 3,0,6000,14000
!
```

### 7.2.2 WR[ITE] channel,block,minus word count,buffer address

Write to a disk file associated with the specified channel from the data field specified by the first character of the buffer address and location specified by "buffer address". If you attempt to WRITE to a channel which is not LOOKed UP on an ETOS file, the message "?CHANNEL NOT OPEN" is printed. The WRITE starts at the specified "block", which is an octal number in the range 0 to 37777777. The word count is rounded up to the nearest multiple of 400 (base 8) or 256 (base 10). See 7.4.3 for further details. A list of possible WRITE errors is contained in Table 7-3.

Table 7-3  
SCALE WRITE Errors

<u>Error</u>	<u>Meaning</u>
1	Channel logically write-protected
7	Write beyond end of file
minus	Hard error

For example, to WRITE data from locations 0 through 3777 in field 0 onto blocks 10 through 17 in the file LOOKed UP on channel 4, enter the command specified in Figure 7-2.

Figure 7-2  
Sample SCALE WRITE

```
!WRITE 4,10,4000,0  
!
```

### 7.2.3 EN[TER]channel=[device:][[p,pn]]filename.ext[<protection>][,size]

Enter a file into an ETOS file directory and associate it with the specified channel. If you attempt to ENTER on a channel which is already LOOKED UP on an ETOS file, the message "?CHANNEL IN USE" is printed. You must CLOSE this channel before you can

use it. This directory is distinct from the OS/8 directory. The format of the protection code is contained in Table 7-4.

Table 7-4  
Protection Code Format

<u>Bits</u>	<u>Meaning</u>
0	Privileged file if set
1-5	Unused
6	Write protect against different project number
7	Read protect against different project number
8	Write protect against same project number, different programmer number
9	Read protect against same project number, different programmer number
10	Write protect against same project number, same programmer number
11	Read protect against same project number, same programmer number

The "size" is the length of the file, expressed as an octal number in the range 0 to 37777777. Under OS/8, the largest file size is 4096 blocks. An ETOS file can be up to 8,388,608 blocks long, and is limited only by the medium on which ETOS is running. This number is the largest that can be represented in 23 bits.

If the file is variable length, this value is positive. If the file is fixed length, this value is negative. When negative, the value is the two's complement of the actual file size. A fixed length file of 17 octal blocks is represented as having length = 77777761. See 7.4.5 for further details. A list of possible ENTER errors is contained in Table 7-5.



Table 7-5  
SCALE ENTER ERRORS

<u>Error</u>	<u>Meaning</u>
1	Unknown device
2	Device not owned by user
4	File already exists. File is looked up on specified channel.
5	Account number not found
6	Master file directory not found. (This should never happen.)
7	Not enough room on device. If fixed length ENTER, no file is created. If variable length ENTER, all available blocks are assigned to the file.
10	File directory is write-protected
11	Map block overflow. Disk is too fragmented to accommodate mapping the entire file. Unused files should be deleted.

For example, to create a 15 (octal) block variable length file called "TEST.TST" with a protection code of 50 on the system disk in account [10,11], enter the command specified in Figure 7-3. This file is associated with channel 7.

Figure 7-3  
Sample SCALE Variable Length Enter

```
!ENTER 7=DK0:[10,11]TEST.TST<50>,15
!
```

To create the same file as a fixed length file, enter the command specified in Figure 7-4.

Figure 7-4  
Sample SCALE Fixed Length Enter

```
!ENTER 7=DK0:[10,11]TEST.TST<50>,7777763
!
```

7.2.4 LOO[KUP] channel=[device:][[p,pn]]filename.ext  
 for LOOKING up an ETOS file on an ETOS file-structured device

Look up a file. This function is used to search an ETOS directory for the existence of the specified file and associate it with the specified channel. If you attempt to LOOKUP a channel which is already LOOKED UP on an ETOS file, the message "CHANNEL IN USE" is printed. You must CLOSE the channel before you can use it. See 7.4.6 for further details. A list of possible file structured LOOKup errors is contained in Table 7-6.

Table 7-6  
SCALE File Structured LOOKUP Errors

<u>Error</u>	<u>Meaning</u>
1	Unknown device
2	Device not owned by user
4	File not found
5	Account number not found
6	Master file directory not found. (This should never happen)
10	File is read-protected

For example, to associate the file "OS8DISK.DSK" on DK2 in account [15,70] with channel 4, enter the command specified in Figure 7-5.

Figure 7-5  
Sample SCALE File Structured LOOKUP

```
!LOOKUP 4=DK2:[15,70]OS8DISK.DSK
!
```

LOO[KUP] channel=device:[,offset]  
 for accessing a non ETOS file-structured data pack

This form of the LOOKUP command is used to associate a non-file structured device with the specified channel. The offset is an

octal number in the range 0 to 37777777. Input/output to the device is performed with the offset appearing as block 0. All READS and WRITES to the device will thus be relative to the specified offset. This method could be used to set up multiple OS/8 directories on a private OS/8 format pack. A list of possible non file-structured LOOKUP errors is contained in Table 7-7.

Table 7-7  
SCALE Non-File Structured LOOKUP Errors

<u>Error</u>	<u>Meaning</u>
1	Unknown device
2	Device not owned by user

For example, to associate the "B" side of an RK05 pack in drive 1 with channel 3, enter the command specified in Figure 7-6.

Figure 7-6  
Sample SCALE Non-File Structured Lookup

```
!LOOKUP 3=DK1:,6260
!
```

#### 7.2.5 EXT[END] channel,base block,length

Extend a file currently open on the specified channel. If you attempt to EXTEND a channel which does not have an ETOS file LOOKED UP on it, the message "?CHANNEL NOT OPEN" is printed. This function is used to add a number of blocks to a file, starting at a base block. It can be used to extend the file from the end of the file, to insert a number of blocks into the middle of a file, or to add a number of blocks to the beginning of the file. The base block is an octal number in the range 0 to 37777777. The specified number of blocks is inserted before this base block. The starting block number of a file is 0, so that the highest block number in a 15 block file is block number 14.

Bit 0 of the base block is used as an indicator. If this bit is not set (0), no special action is taken. If this bit is set (1), the base block is ignored and the extend automatically takes place from the end of the file. Using this method, you do not have to know the exact length of the current file. To extend from the end of the file, enter EXTEND chan,4000,length.

If a file has a zero length, block 0 does not exist. Therefore, a base address is illegal and will result in an error message. In this case, the user must utilize the extend from the end technique. The length of the blocks to extend is an octal number in the range from 0 to 37777777. See 7.4.7 for further details. A list of possible EXTEND errors is contained in Table 7-8.

Table 7-8  
SCALE EXTEND Errors

<u>Error</u>	<u>Meaning</u>
2	Non file-structured
3	File is write-protected
4	File is fixed length
5	Base block out of range
6	Map block overflow. Disk is too fragmented to accommodate mapping the entire file. Unused files should be deleted.
7	No room for EXTEND. All available blocks are assigned to the file.
10	File directory is write-protected

For example, to EXTEND a 15 (base 8) block file LOOKed UP on channel 6 by 22 (base 8) blocks, enter the command specified in Figure 7-7.

Figure 7-7  
Sample SCALE EXTEND

```
!EXTEND 3,16,22  
!
```

### 7.2.6 RED[UCE] channel,base block,length

Reduce a file currently open on the specified channel. If you attempt to REDUCE a channel which does not have an ETOS file LOOKED UP on it, the message "?CHANNEL NOT OPEN" is printed. This function is used to reduce a number of blocks from the end of the file, or to reduce a number of blocks from the middle of a file, or to reduce a number of blocks from the beginning of the file. The base block is an octal number in the range 0 to 37777777. The specified number of blocks is reduced, starting at the base block. The length of the blocks to reduce is an octal number from 0 to 37777777. The starting block number of a file is 0, so that the highest block number in a 15 block file is block number 14. See 7.4.8 for further details. A list of possible REDUCE errors is contained in Table 7-9.

Table 7-9  
SCALE REDUCE Errors

<u>Error</u>	<u>Meaning</u>
1	Channel not open
2	Non file-structured device
3	File is write-protected
4	File is fixed-length
5	Base block out of range
6	Map block overflow. Disk is too fragmented to accommodate splitting the file. Unused files should be deleted. This error can only occur if the REDUCE is not done at the end of the file.

For example, to REDUCE a 72 (base 8) block file LOOKED UP on channel 3 by 16 (base 8) blocks, enter the command specified in Figure 7-8.

Figure 7-8  
Sample SCALE REDUCE

```
!REDUCE 3,54,16  
!
```

### 7.2.7 DEL[ETE] channel

Delete a file currently open on the specified channel. If you attempt to DELETE a channel which does not have an ETOS file LOOKED UP on it, the message "?CHANNEL NOT OPEN" is printed. This is identical to a REDUCE to a file length of 0, except it also removes the file's directory entry. See 7.4.9 for further details. A list of possible DELETE errors is contained in Table 7-10.

Table 7-10  
SCALE DELETE Errors

<u>Error</u>	<u>Meaning</u>
2	Non file-structured device
3	File is write-protected
4	File in use (Variable length file is REDUCed to zero length but not deleted. Fixed length file is untouched.)

For example, to DELETE the file currently LOOKed UP on channel 2 from the ETOS directory, enter the command specified in Figure 7-9

Figure 7-9  
Sample SCALE DELETE

```
!DELETE 2  
!
```

### 7.2.8 REN[AME] channel=filename.ext[<protection>]

Rename a file currently open on the specified channel. If you attempt to RENAME a channel which does not have an ETOS file LOOKED UP on it, the message "?CHANNEL NOT OPEN" is printed. This function is used to change the file name and/or protection code. See 7.4.10 for further details. A list of possible RENAME errors is contained in Table 7-11.

Table 7-11  
SCALE RENAME Errors

<u>Error</u>	<u>Meaning</u>
2	Non file-structured device
3	File is write-protected

For example, to RENAME the ETOS file LOOKED UP on channel 4 to the name "OS8DSK2.DSK" with a protection code of 50, enter the command specified in Figure 7-10.

Figure 7-10  
Sample SCALE RENAME

```
!RENAME 4=OS8DSK2.DSK<0050>  
!
```

### 7.2.9 CL[OSE] channel

Close an open channel. This operation frees the channel for association with another ETOS file. CLOSE does not make the file permanent; the ENTER function has already accomplished that. Closing a channel which is already closed is redundant but not an error. When you LOGOUT, all channels are automatically closed. See 7.4.13 for further details.

There is no error condition for this command.

For example, to free channel 6 from its current association with an ETOS file, enter the command specified in Figure 7-11.

Figure 7-11  
Sample SCALE CLOSE

```
!CLOSE 6  
!
```

#### 7.2.10 I[NQUIRE] channel

Inquire about a file open on a channel. If you attempt to INQUIRE about a channel which does not have an ETOS file LOOKED UP on it, the message "?CHANNEL NOT OPEN" is printed.

INQUIRE returns arguments in the form

device:[p,pn] filename.ext <protection> size.

If a channel is associated with a non file-structured device, arguments are returned in the form

device:

For this form, offsets are not displayed. See 7.4.14 for further details.

For example, to INQUIRE which ETOS file is LOOKED UP on channel 2, enter the command specified in Figure 7-12.

Figure 7-12  
Sample SCALE INQUIRE

```
!INQUIRE 2  
SYS:[0,3]OS8DISK.DSK <0050> 1000
```



### 7.3 RESTRICTIONS ON THE USE OF THE CHANNEL COMMANDS

Certain channel commands are illegal unless the VM (virtual machine) is halted. They are dangerous to the user, to the system, or to both if they are executed while the VM is running. These are CLOSE, DELETE, ENTER, EXTEND, INQUIRE, LOOKUP, READ, REDUCE, RENAME, and WRITE. Attempting to execute these commands while the VM is running will be unsuccessful, and the message

?PLEASE TYPE <^VS><CR> FIRST

will be displayed on the terminal. The commands that are listed above are by no means the only ones which might be dangerous to a user program. For example, entering ^VPC n while the VM is running is very likely to produce unwanted results. Certain commands may be executed when the terminal is not logged in. They are ATTACH, HELLO, LOGIN, SYS, and VERSION. Attempting to execute other commands when the terminal is not logged in will be unsuccessful, and the message

?LOGIN PLEASE

will be displayed on the terminal. A summary of commands and when they may be executed is provided in Appendix A.

## 7.4 ASSEMBLER PROGRAMMING OF CHANNEL INPUT/OUTPUT INSTRUCTIONS

CHANIO 6100 CHANNEL I/O TO OR FROM USER'S DATA FIELD

This IOT is used to perform various operations on ETOS channels. See 7.1 for an explanation of the concept of channels.

CHANIO Calling Parameters:

MQ = Function number plus channel number.

AC = Argument block address minus one

DF = Data field for input/output buffer

A summary of CHANIO functions is contained in Table 7-12.

Table 7-12  
CHANIO Functions

<u>Mnemonic</u>	<u>Function Number</u>
READW	0000
READC	0100
WRITEW	0200
WRITEC	0300
ENTER	0400
LOOKUP	0500
EXTEND	0600
REDUCE	0700
DELETE	1000
RENAME	1100
WAITW	1200
WAITC	1300
CLOSE	1400
INQUIRE	1500

### 7.4.1 READW (Function 0000)

Read from a channel into the data field. The AC is a pointer to a four-word argument block in the user's instruction field. AC = (address of block) minus one. The form of the argument block is

Word 1 = High order block number (these two words make up a double precision block number)

Word 2 = Low order block number

Word 3 = Minus word count (the count is rounded up to the nearest multiple of 400 (base 8) or 256 (base 10)).

Word 4 = First location of buffer into which data is read.

An error code is returned in the AC. Possible error codes are contained in Table 7-13.

Table 7-13  
CHANIO READW Errors

<u>AC</u>	<u>Meaning</u>
0	No error
2	Channel not open
7	Read beyond end of file
minus	Hard error

A sample calling sequence is contained in Figure 7-13.

Figure 7-13  
Sample CHANIO READW

```

CHANIO=6100           /READW IS FUNCTION 0
READW=0
CHN=X                /CHANNEL NUMBER
BUFFLD=Y             /FIELD CONTAINING BUFFER
FLDCHG=BUFFLD^10    /ARGUMENT FOR CDF
BLOCKCOUNT=Z        /# OF BLOCKS TO TRANSFER
WORDCOUNT=400^BLOCKCOUNT /# OF WORDS TO TRANSFER
CLA CLL
TAD      (READW+CHN
MQL
TAD      (ARGBLK-1    /MQ = FUNCTION + CHANNEL
                        /AC = POINTER TO ARGUMENT BLOCK
CDF FLDCHG           /FIELD FOR I/O BUFFER
CHANIO
SZA
JMP      ERROR       /CHECK ERROR CODE IN AC
                        /ERROR ROUTINE
.
.
.
ARGBLK,  HIGH BLOCK  /HIGH ORDER BLOCK
          LOW BLOCK   /LOW ORDER BLOCK
          -WORDCOUNT /-WORD COUNT
          BUF         /BUFFER ADDRESS

PAGE
FIELD BUFFLD
BUF,     ZBLOCK WORDCOUNT

```

### 7.4.2 READC (Function 0100)

Read from a channel into the data field. This function works exactly like function 0000, except the program continues execution while the read is taking place. OS/8 input/output must wait until completion to continue processing and therefore must only use READW and WRITEW functions.

### 7.4.3 WRITEW (Function 0200)

Write to a channel from the data field. The form of the argument block is identical to function 0000's. An error code is returned in the AC. Possible error codes are contained in Table 7-14.

Table 7-14  
CHANIO WRITEW Errors

<u>AC</u>	<u>Meaning</u>
0	No error
1	Channel logically write-protected
2	Channel not open
7	Write beyond end of file
22	Hard error

A sample calling sequence is contained in Figure 7-14.

Figure 7-14  
Sample CHANIO WRITEW

```

CHANIO=6100
WRITEW=0200           /WRITEW IS FUNCTION 200
CHN=X                 /CHANNEL NUMBER
BUFFLD=Y              /FIELD CONTAINING BUFFER
FLDCHG=BUFFLD^10     /ARGUMENT FOR CDF
BLOCKCOUNT=Z        /# OF BLOCKS TO TRANSFER
WORDCOUNT=400^BLOCKCOUNT /# OF WORDS TO TRANSFER
CLA CLL
TAD      (WRITEW+CHN
MQL
TAD      (ARGBLK-1    /MQ = FUNCTION + CHANNEL
                        /AC = POINTER TO ARGUMENT BLOCK
CDF FLDCHG          /FIELD FOR I/O BUFFER
CHANIO
SZA              /CHECK ERROR CODE IN AC
JMP      ERROR    /ERROR ROUTINE
ARGBLK,  HIGH BLOCK /HIGH ORDER BLOCK
          LOW BLOCK /LOW ORDER BLOCK
          -WORDCOUNT /-WORD COUNT
          BUF        /BUFFER ADDRESS

PAGE
FIELD BUFFLD
BUF,           /BUFFER CONTAINING DATA

```

#### 7.4.4 WRITEC (Function 0300)

Write to a channel from the data field. This function works exactly like function 0200, except the program continues execution while the write is taking place. OS/8 input/output must wait until completion to continue processing and therefore must only use READW and WRITEW functions.

#### 7.4.5 ENTER (Function 0400)

Enter a file into an ETOS file directory and associate it with the specified channel. This directory is distinct from the OS/8 directory. The AC is a pointer to a twelve-word argument block in the user's data field. AC = (address of block) minus one. The argument block has the form

- Word 1 - 2 = Name of the device on which to create file. The name is in packed 6-bit ASCII.
- Word 3 = Account number. The project number is contained in the high order six bits. The programmer number is contained in the lower order six bits.
- Word 4 - 8 = File name. The first three and one-half words contain the file name. The last one and one-half words contain the file extension.
- Word 9 = Protection code. The format of the protection code is contained in Table 7-15.

Table 7-15  
Protection Code Format

<u>BITS</u>	<u>MEANING</u>
0	Privileged file
1-2	For account number entries in the master file directory, this is the amount of additional memory beyond 8K to assign to a job on LOGIN. 00 = 8K. 01 = 12K. 10 = 16K. 11 = 20K.
3-5	Unused
6	Write protect against other project no.
7	Read protect against other project no.
8	Write protect against same project number, different programmer number
9	Read protect against same project number, different programmer number
10	Write protect against same project number, same programmer number
11	Read protect against same project number, same programmer number

Word 10 = Date word. The format is contained in Table 7-16.

Table 7-16  
Date Word Format

<u>Bits</u>	<u>Meaning</u>
0-3	Current month
4-8	Current day
9-11	Current year - 1970 for 1974 to 1977
9-11	Current year - 1978 for 1978 to 1981

The above format is similar to the format of the OS/8 date word. The date March 17, 1979 is equal to 001101111001 (base 2) or 1571 (base 8).

Word 11 - 12 = File size. These two words contain a double-precision value indicating file size. Under OS/8, the largest file size is 4096 blocks. An ETOS file can be up to 8,388,608 blocks long, which is the largest number that can be represented in 23 bits.

If the file is variable length, this value is positive. If the file is fixed length, this value is the two's complement of the actual file size. A fixed length of 17 blocks = 77777761.

An error code is returned in the AC. Possible error codes are contained in Table 7-17.

Table 7-17  
CHANIO ENTER Errors

<u>AC</u>	<u>Meaning</u>
0	ENTER successful
1	Unknown device
2	Device not owned by user
3	Channel in use
4	File already exists.
5	Account number not found
6	Master file directory not found. (This should never happen.)
7	Not enough room on device. If fixed length ENTER, no file is created. If variable length, all available blocks are assigned to the file.
10	File directory is write-protected
11	Map block overflow. Disk is too fragmented to accommodate mapping the entire file. Unused files should be deleted.

There is no error code for hard errors. If a hard error occurs during an ENTER, the device probably has to be rebuilt.

A sample calling sequence is contained in Figure 7-15.

Figure 7-15  
Sample CHANIO Enter

```
CHANIO=6100
ENTER=400           /ENTER IS FUNCTION 400
CHN=X              /CHANNEL NUMBER
ARGFLD=Y           /FIELD CONTAINING ARGUMENT BLOCK
FLDCHG=ARGFLD^10  /ARGUMENT FOR CDF
.
.
.
CCL CLL
TAD      (ENTER+CHN
MQL
TAD      (ARGBLK-1  /MQ = FUNCTION + CHANNEL
CDF FLDCHG        /AC= POINTER TO ARGUMENT BLOCK
CHANIO
SZA
JMP      ERROR     /CHECK ERROR CODE IN AC
                        /ERROR ROUTINE
.
.
.

FIELD ARGFLD
ARGBLK, DEVICE DEVI /DEVICE NAME
                ACCNT /ACCOUNT NUMBER
                TEXT "FILENAMEXT" /FILE NAME
                *.-1
                PROT  /PROTECTION CODE
                DATE  /DATE WORD
                HSIZE /HIGH ORDER FILE SIZE
                LSIZE /LOW ORDER FILE SIZE
```

#### 7.4.6 LOOKUP (Function 0500)

Look up a file. This function is used to search an ETOS directory for the existence of the specified file and associate it with the specified channel. The AC is a pointer to an eight-word argument block in the user's data field. AC = (address of block) minus one. The format of the block is as follows.

Word 1- 2 = Name of the device on which the specified file is searched. The name is in packed 6-bit ASCII.

Word 3 = Account number. The project number is contained in the high order six bits. The programmer number is contained in the low order six bits.

Word 4- 8 = File name. The first three and one-half words contain the file name. The last one and one-half words contain the file extension.

An error code is returned in the AC. Possible error codes are contained in Table 7-18.

Table 7-18  
CHANIO File Structured LOOKUP Errors

<u>AC</u>	<u>Meaning</u>
0	LOOKUP succeeded
1	Unknown device
2	Device not owned by user
3	Channel in use
4	File not found
5	Account number not found
6	Master file directory not found. (This should never happen.)
10	File is read-protected

A sample calling sequence is contained in Figure 7-16.



Figure 7-16  
Sample CHANIO File Structured Lookup

```

CHANIO=6100
LOOKUP=0500           /LOOKUP IS FUNCTION 500
CHN=X                /CHANNEL NUMBER
ARGFLD=Y             /FIELD CONTAINING ARGUMENT BLOCK
FLDCHG=ARGFLD^10    /ARGUMENT FOR CDF
.
.
.
CLA CLL
TAD          (LOOKUP+CHN
MQL
TAD          (ARGBLK-1   /MQ = FUNCTION + CHANNEL
CDF          FLDCHG     /AC = POINTER TO ARGUMENT BLOCK
CHANIO
SZA          /CHECK ERROR CODE IN AC
JMP          ERROR     /ERROR ROUTINE
.
.
.
FIELD ARGFLD
ARGBLK, DEVI  /DEVICE NAME
ACCNT        /ACCOUNT NUMBER
TEXT "FILENAMEXT" /FILE NAME

```

LOOKUP may also be used to associate a non-file structured device with the specified channel.

Note: ETOS considers a standard single-user OS/8 pack or COS pack as a non-file structured device. The calling sequence is slightly different for this form of LOOKUP. The AC is a pointer to a six-word argument block in the user's data field. AC = (address of block) minus one. The format of the block is as follows

Word 1 - 2 = Device name.

Word 3 = Zero.

Word 4 = Zero. (Indicates non file-structured.)

Word 5 - 6 = Offset. These two words make up a double precision offset for accessing the device. Input/output to the device is performed with the offset appearing as block 0. All READs and WRITEs to the device are relative to the specified offset. This method could be used to set up multiple OS/8 directories on a private OS/8 format pack. Word 5 is the high order offset; word 6 is low order.

An error code is returned in the AC. Possible error codes are contained in Table 7-18.

Table 7-19  
CHANIO Non-File Structured LOOKUP Errors

AC	Meaning
0	LOOKUP succeeded
1	Unknown device
2	Device not owned by user
3	Channel in use

A sample calling sequence is contained in Figure 7-17.

Figure 7-17  
Sample CHANIO Non-File Structured Lookup

```

CHANIO=6100
LOOKUP=0500           /LOOKUP IS FUNCTION 500
CHN=X                /CHANNEL NUMBER
ARGFLD=Y            /FIELD CONTAINING AGREEMENT BLOCK
FLDCHG=Y^10        /ARGUMENT FOR CDF
CLA CLL
TAD      (LOOKUP+CHN
MQL      /MQ = FUNCTION + CHANNEL
TAD      (ARGBLK-1  /AC = POINTER TO ARGUMENT BLOCK
CDF      FLDCHG    /ARGBLK IS IN DATA FILED
CHANIO
SZA      /CHECK ERROR CODE
JMP ERROR /ERROR ROUTINE IN AC
.
.
.
FIELD    ARGFLD
ARGBLK,  DEVI DEVI /DEVICE NAME
        0
        0
        HIGH ORDER /HIGH ORDER OFFSET
        LOW ORDER  /LOW ORDER OFFSET

```

#### 7.4.7 EXTEND (Function 0600)

Extend a file currently open on the specified channel. This function is used to add a number of blocks to a file, starting at a base block. It can be used to extend a file from the end of the file, or to insert a number of blocks into the middle of a file, or to add a number of blocks to the beginning of the file. The AC is a pointer to a four-word block in the user's data field. AC = (address of block) minus one. The format of the block is as follows.

Word 1 - 2 = Base of starting block for extend. These two words make up a double precision block number. The specified number of blocks is inserted before this base block. Bit 0 of the high order word is used as an indicator. If this bit is not set (0), no special action is taken. If the bit is set (1), the base block is ignored and the extend automatically takes place from the end of the file. Using this method, you do not have to know the exact length of the current file. If a file has a zero length, block 0 does not exist. Therefore, a base address is illegal and results in an error message. In this case, the user must utilize the extend from the end technique.

Word 3 - 4 = Double precision number of blocks to add to the file.

An error code is returned in the AC. Possible error codes are contained in Table 7-20.

Table 7-20  
CHANIO EXTEND Errors

<u>AC</u>	<u>Meaning</u>
0	EXTEND succeeded
1	Channel not open
2	Nonfile structured device
3	File is write-protected
4	File is fixed length
5	Base block out of range
6	Map block overflow. Disk is too fragmented to accommodate mapping the entire file. Unused files should be deleted.
7	No room for EXTEND. All available blocks are assigned to the file.
10	File directory is write-protected

A sample calling sequence is contained in Figure 7-18.

Figure 7-18  
Sample CHANIO Extend

```

CHANIO=0100
EXTEND=0600           /EXTEND IS FUNCTION 600
CHN=X                /CHANNEL NUMBER
ARGFLD=Y            /FIELD CONTAINING ARGUMENT BLOCK
FLDCHG=ARGFLD^10    /ARGUMENT FOR CDF
.
.
.
CLA CLL
TAD      (EXTEND+ CHN
MQL
TAD      (ARGBLK-1    /MQ = FUNCTION + CHANNEL
CDF FLDCHG          /AC=ARGUMENT POINTER
CHANIO
SZA
JMP      ERROR      /ARGBLK IS IN DATA FIELD
.
.
.
FIELD ARGFLD
ARGBLK,  HIGH BASE   /HIGH ORDER BASE BLOCK
          LOW BASE   /LOW ORDER BASE BLOCK
          HIGH EXT   XXXX /HIGH ORDER EXTEND
          LOW EXT,   XXXX /LOW ORDER EXTEND

```

#### 7.4.8 REDUCE (Function 0700)

Reduce a file currently open on the specified channel. The calling sequence is identical to the EXTEND function. As in EXTEND, the user can delete blocks from the beginning, middle or end of the file. A negative base block cannot be used to reduce blocks from the end of the file. Reducing a file to zero length does not get rid of a file. DELETE must be used for this purpose. The AC is a pointer to a four-word block in the user's data field. AC = (address of block) minus one. The format of the block is as follows.

Word 1 - 2 = Base of starting block for reduce. These two words make up a double precision block number. The specified number of blocks are removed after this base block. If this block number is past the end of the file, the REDUCE is ignored.

Word 3- 4 = Number of blocks to subtract from the file size. These two words make up a double precision number of blocks. Bit 0 of the high order word is used as an indicator. If this bit is not set (0), no special action is taken. If the bit is set (1), the number of blocks is ignored and the reduce

removes all blocks after the specified base block. Using this method, you do not have to know the exact length of the current file.

An error code is returned in the AC. Possible error codes are contained in Table 7-21.

Table 7-21  
CHANIO REDUCE Errors

<u>AC</u>	<u>Meaning</u>
0	REDUCE succeeded
1	Channel not open
2	Non file-structured device
3	File is write-protected
4	File is fixed length
5	Base block out of range
6	Map block overflow. Disk is too fragmented to accommodate splitting the file. This error can only occur if the REDUCE is not done at the end of the file. Unused files should be deleted.

A sample calling sequence is contained in Figure 7-19.

Figure 7-19  
Sample CHANIO Reduce

```

CHANIO=6100
REDUCE=0700           /REDUCE IS FUNCTION 700
CHN=X                /CHANNEL NUMBER
ARGFLD=Y             /FIELD CONTAINING ARGUMENT BLOCK
FLDCHG=ARGFLD^10    /ARGUMENT FOR CDF
.
.
.
CLA CLL
TAD (REDUCE+CHN
MQL                  /MQ = FUNCTION + CHANNEL
TAD (ARGBLK-1        /AC = ARGUMENT POINTER
CDF FLDCHG           /ARGBLK IS IN DATA FIELD
CHANIO
SZA                  /CHECK ERROR CODE IN AC
JMP ERROR            /ERROR ROUTINE
.
.
.
FIELD ARGFLD
ARGBLK,  HIGH BASE   /HIGH ORDER BASE BLOCK
          LOW BASE    /LOW ORDER BASE BLOCK
          HIGH RED     /HIGH ORDER REDUCE
          LOW RED      /LOW ORDER REDUCE

```

#### 7.4.9 DELETE (Function 1000)

Delete a file currently open on the specified channel. This is identical to a REDUCE to a file length of 0, except it also removes the file's directory entry. The AC may be any value on entry. An error code is returned in the AC. Possible error codes are contained in Table 7-22.

Table 7-22  
CHANIO DELETE ERRORS

<u>AC</u>	<u>Meaning</u>
0	DELETE succeeded
1	Channel not open
2	Non file-structured device
3	File is write-protected
4	File in use. (Variable length file is REDUCED to zero length but not deleted. Fixed length file is untouched.)

A sample calling sequence is contained in Figure 7-20.

Figure 7-20  
Sample CHANIO DELETE

```
CHANIO=6100
DELETE=1000      /DELETE IS FUNCTION 1000
CHN=X           /CHANNEL NUMBER
CLA CLL
TAD (DELETE+CHN
MQL            /MQ = FUNCTION + CHANNEL
CHANIO
SZA           /CHECK ERROR CODE IN AC
JMP ERROR     /ERROR ROUTINE
```

#### 7.4.10 RENAME (Function 1100)

Rename a file currently open on the specified a channel. The AC is a pointer to a seven-word argument in the user's data field. AC = (address of block) minus one.

The argument block has the form

Word 1 - 5 = File name. The first three and one-half words contain the file name. The last one and one-half words contain the file extension.

Word 6 = Protection code. The format of the protection code is contained in Table 7-23.

Table 7-23  
Protection Code Format

<u>Bits</u>	<u>Meaning</u>
0	Privileged file
1-2	For account number, entries in the Master file directory, this is the amount of additional memory beyond 8K words to assign to a job on LOGIN. 00 = 8K. 01 = 12K. 02 = 16K. 03 = 20K.
3-5	Unused
6	Write protect against different project number
7	Read protect against different project number
8	Write protect against same project number, different programmer number
9	Read protect against same project number, different project number
10	Write protect against same project number, same programmer number
11	Read protect against same project number, same programmer number

Word 7 = Date word. The format is specified in Table 7-24.

Table 7-24  
Date Word Format

<u>Bits</u>	<u>Meaning</u>
0-3	Current month
4-8	Current day
9-11	Current year - 1970 for 1974 to 1977 Current year - 1978 for 1978 to 1981

The above format is similar to the OS/8 date word.

An error code is returned in the AC. Possible error codes are contained in Table 7-25.

Table 7-25  
CHANIO RENAME Errors

<u>AC</u>	<u>Meaning</u>
0	RENAME succeeded
1	Channel not open
2	Non file-structured device
3	File is write-protected
4	Bad file name. (First two characters are nulls.)

A sample calling sequence is contained in Figure 7-21.

Figure 7-21  
Sample CHANIO RENAME

```

CHANIO=6100
RENAME=1100           /RENAME IS FUNCTION 110
CHN=X                 /CHANNEL NUMBER
ARGFLD=Y              /FIELD CONTAINING AARGUMENT BLOCK
FLDCHG=ARGFLD^10     /AGRUMENT FOR CDF
CLA CLL
TAD (RENAME+CHN
MQL                   /MQ = FUNCTION + CHANNEL
TAD (ARGBLK-1         /AC = ARGUMENT POINTER
CDF FLDCHG            /ARGBLK IS IN DATA FIELD
CHANIO
SZA                   /CHECK ERROR CODE IN AC
JMP ERROR             /ERROR ROUTINE
FIELD ARGFLD
ARGBLK, TEXT "FILENAMEXT" /NEW FILE NAME
        *.-1
        PROT           /NEW PROTECTION CODE
        DATE           /DATE OF RENAMING

```

7.4.11 WAITW (Function 1200)

Wait for I/O completion. This function puts the job to sleep until the completion of an I/O operation, currently taking place on an open channel. WAITW may be used with the READC and WRITEC functions (codes 0100 and 0300). This function is not necessary when using READW and WRITEW. The AC may be any value on entry. An error code is returned in the AC. Possible error codes are contained in Table 7-26.

Table 7-26  
CHANIO WAITW Errors

<u>AC</u>	<u>Meaning</u>
0	I/O Complete
1	Channel not open
4000	Hard error

A sample calling sequence is contained in Figure 7-22.



Figure 7-22  
Sample CHANIO WAITW

```

CHANIO=6100
WAITW=1200           /WAITW IS FUNCTION 1200
CHN=X               /CHANNEL NUMBER
TAD      (WAIT+CHN
MQL                      /MQ = FUNCTION + CHANNEL
CHANIO
SZA                      /CHECK ERROR CODE IN AC
JMP      ERROR        /ERROR ROUTINE
.
.
.

```

7.4.12 WAITC (Function 1300)

Test for I/O completion. This function checks an open channel to see if I/O is currently taking place. The calling sequence is identical to the WAITW function. WAITC may be used with the READC and WRITEC functions (codes 0100 and 0300). This function is not necessary when used with READW and WRITEW. An error code is returned in the AC. Possible error codes are contained in Table 7-27.

Table 7-27  
CHANIO WAITC Errors

<u>AC</u>	<u>Meaning</u>
0	No I/O taking place
1	Channel not open
4000	Hard error
-n	n disk requests in queue (channel still busy)

A sample calling sequence is contained in Figure 7-23.

Figure 7-23  
Sample CHANIO WAITC

```

CHANIO=6100
WAITC=1300          /WAITC IS FUNCTION 1300
CHN=X              /CHANNEL NUMBER
TAD (WAITC+CHN
MQL                      /MQ = FUNCTION + CHANNEL
CHANIO
SNA
JMP      NOIO        /I/O COMPLETE
AND      (3777
SNA      CLA        /CHECK ERROR CODE
JMP      ERROR

```

#### 7.4.13 CLOSE (Function 1400)

Close an open channel. This operation frees the channel for association with another ETOS file. CLOSE does not make the file permanent, as the ENTER function has already accomplished that task. The AC is cleared on return. There is no error condition for this function. Closing a channel which is already closed is redundant but not an error. A sample calling sequence is contained in Figure 7-24.

Figure 7-24  
Sample CHANIO CLOSE

```
CHANIO=6100
CLOSE=1400           /CLOSE IS FUNCTION 1400
CHN=X                /CHANNEL NUMBER
TAD      (CLOSE+CHN
MQL
CHANIO
/AC = 0              /NO ERRORS POSSIBLE
```

#### 7.4.14 INQUIRE (Function 1500)

Inquire about a file open on a channel. The AC contains a pointer to a sixteen-word block for information deposited in the user's data field. AC = (address of block) minus one. The block of information, which is returned by this instruction is of the form

- Word 1 - 2 = Device name. The name is in packed 6-bit ASCII. If the device is non-file structured, only the device name is returned. The offset is not available to the user.
- Word 3 = Account number. The project number is contained in the high order six bits. The programmer number is contained in the low order six bits.
- Word 4 - 8 = File name. The first three and one-half words contain the file name. The last one and one-half words contain the file extension.
- Word 9 = Protection code. The format of the protection code is contained in Table 7-28.

Table 7-28  
Protection Code Format

<u>Bits</u>	<u>Meaning</u>
0	Privileged file
1-2	For UFDs, this is the amount of additional memory fields beyond 8K words to assign to an account at LOGIN time.
3-5	Unused.
6	Write protect against different project number.
7	Read protect against different project number.
8	Write protect against same project number, different programmer number.
9	Read protect against same project number, different programmer number.
10	Write protect against same project number, same programmer number.
11	Read protect against same project number, same programmer number.

Word 10 = Segment count. This is the number of physically contiguous segments into which the file is mapped.

Word 11 - 12 = System use. These words have no meaning to the user.

Word 13 - 14 = File size. These words make up a 23-bit double precision number representing the file size.

Word 15 = Entry number in user file directory (UFD).

Word 16 = Pointer to file control block (FCB).

An error code is returned in the AC. Possible error codes are contained in Table 7-29.

Table 7-29  
CHANIO INQUIRE Errors

AC	Meaning
0	INQUIRE succeeded
1	Channel not open
2	Non file-structured device. This is not necessarily an error.
3	DMON unable to obtain the user's data field because the user is simultaneously executing a scale command that also requests his data field (such as EXAMINE or DEPOSIT).

A sample calling sequence is contained in Figure 7-25.

Figure 7-25  
Sample CHANIO INQUIRE

```

CHANIO=6100
INQUIRE=1500           /INQUIRE IS FUNCTION 1500
CHN=X                   /CHANNEL NUMBER
BUFFLD=Y                /FIELD CONTAINING BUFFER
FLDCHG=BUFFLD^10       /ARGUMENT FOR CDF
.
.
CLA CLL
TAD      (INQUIRE+CHN
MQL
TAD      (BUF-1         /MQ = FUNCTION + CHANNEL
CDF      FLDCHG        /AC = POINTER TO BUFFER
CHANIO
SZA
JMP      ERROR        /CHECK ERROR CODE
                        /ERROR ROUTINE
.
.
FIELD BUFFLD
BUF,     0             /DEVICE
                        /DEVICE
ACCNT,   0             /ACCOUNT
NAME,    0             /FILE NAME CHARS 1,2
                        /FILE NAME CHARS 3,4
                        /FILE NAME CHARS 5,6
                        /FILE NAME CHARS 7,8
                        /FILE NAME CHARS 9,10
PROTEC,  0             /PROTECTION CODE
SEG,     0             /SEGMENT COUNT
SYSUSE,  0             /SYSTEM USE
                        /SYSTEM USE
SIZE,    0             /LOW ORDER FILE SIZE
                        /HIGH ORDER FILE SIZE
ENTRY,   0             /ENTRY # IN UFD
FCBPOI,  0             /POINTER TO FCB

```

APPENDIX A  
SCALE COMMANDS

Table A-1 contains a list of all non-privileged SCALE commands. There are restrictions as to when certain commands may be issued. The State column illustrates these restrictions. The code used for this column is

- L - Terminal must be logged in
- H - Virtual machine must be halted
- N - No restrictions
- O - Terminal must be logged out

The manual section is the section number in the User's Guide which explains the particular SCALE Command.

Table A-1  
SCALE Commands

<u>Command</u>	<u>Description</u>	<u>State</u>	<u>Manual Section</u>
AC [n]	set or print accumulator	L	3.7.1
AS[SIGN] dev[:]	assign device to job	L	3.10
AT[TACH]	attach console to a detached job; detach current job first if logged in.	N	3.13
BO[OT] [chan]	bootstrap channel "chan"	LH	3.5
B[REAK] [n]	set or display keyboard breakmask	L	3.7.5
CL[OSE] chan	close channel "chan"	LH	7.2.9
CON[TINUE] [[[df] if]loc]	resume execution	LH	3.3
COR[E] [n]	allocate n words to VM and/or display list of fields assigned to VM.	LH	3.4
DEA[SSIGN] dev[:]	deassign device "dev:" from VM	L	3.10

<u>Command</u>	<u>Description</u>	<u>State</u>	<u>Manual Section</u>
DEL[ETE] chan	delete the ETOS file that is open on channel "chan"	LH	7.2.7
DEP[OSIT] loc v1[, v2[,...]]	deposit v1,v2,... into virtual memory, starting at location "loc"	L	3.8
DET[ACH]	detach job from console	L	3.13
EN[TER] chan=filespc, length	create ETOS file filespc with specified length on channel "chan"	LH	7.2.3
E[XAMINE] loc[,n]	examine n words in virtual memory, starting at location "loc"	L	3.8
EXT[END] chan baseblk,n	extend ETOS channel "chan" by n blocks after block "baseblk"	LH	7.2.5
F[LAGS] [n]	set or print flag word	L	3.7.2
H[ELLO]	assign a VM to terminal	O	3.2
I[NQUIRE] chan	print data about ETOS file LOOKed UP on channel "chan"	LH	7.2.10
K[STAT] [n]	set or display terminal KSTAT register	L	3.7.6
LOGI[N]	assign a VM to terminal	O	3.2
LOGO[UT]	terminate job's active status	LH	3.2
LOO[KUP] chan=filespc	look up ETOS file "filespc" on channel "chan"	LH	7.2.4
MQ [n]	set or print multiplier quotient	L	3.7.1
PC [n]	set or print program counter	L	3.7.1
Q[UEUE] spldev:=filespc [,offset]	invoke the spooler to list an ETOS file	L	3.11
R filespc	run an ETOS program	LH	3.6
REA[D] chan,sector, mwc,addr	read from an ETOS channel to virtual core	LH	7.2.1

<u>Command</u>	<u>Description</u>	<u>State</u>	<u>Manual Section</u>
RED[UCE] chan, baseblk,n	reduce ETOS file on channel "chan" by n blocks starting at block "baseblk"	LH	7.2.6
REN[AME] chan=filenam .ext[<prot>]	rename ETOS file on channel "chan" as indicated	LH	7.2.8
SC [n]	set or display step counter	L	3.7.3
SR [n]	set or display switch register	L	3.7.4
S[TOP]	halt, go into SCALE mode	L	3.3
SY[STAT]	display ETOS system status (des- troys virtual core)	H	3.14
T[ALK] n message	print message on console n	L	3.9
V[ERSION]	display current version number	N	3.12
W[HERE]	display AC, PC, FLAGS, MQ, SC, SR	L	3.7.7
WR[ITE] chan,sector, mwc,addr	write from virtual core onto an ETOS channel	LH	7.2.2
^C	jump to current ^C restart address	L	3.15
^P	jump to current ^P restart address	L	3.15





APPENDIX B  
SCALE ERROR MESSAGES

Table B-1  
SCALE Error Messages

<u>Error Message</u>	<u>Meaning</u>
?ALREADY LOGGED IN	Attempt to issue LOGIN when not necessary.
?BAD CHANNEL FUNCTION	Incorrect use of a channel by a user program.
?BAD NUMBER	Too many digits were entered, no number was entered where one was required, or non octal digits were entered in an octal number.
?BAD QUEUE DEVICE	There is no spooler running that is initialized for the logical spooling device that was specified in the QUEUE command.
?CANNOT DETACH	There are no dummy consoles for the job.
?CHANNEL IN USE	LOOKUP or ENTER attempted on a channel number that is in use.
?CHANNEL NOT OPEN	The specified channel can not be used because it is closed.
?CHANNEL OPEN ON DEVICE	The device cannot be DISMOUNTed because there is at least one job with a file open on the device. Every channel of every job with a file open on the device must be CLOSED before the device can be DISMOUNTed.
?DEVICE OWNED BY JOB n CONSOLE m	JOB n has to release this device before it can be assigned.
?FILE SPECIFICATION ERROR	An illegal (bad syntax) ETOS file specification was found.

<u>Error Message</u>	<u>Meaning</u>
?ILLEGAL IOT	An illegal IOT (possibly random data) was executed by the user program, or a nonprivileged user attempted to run a privileged program.
?JMP/JMS TO NON EXISTENT FIELD	The user program executed a CIF to a non-existent field then attempted to execute a JMP or JMS. The PC contains the address of the JMP/JMS and bits 6-8 of the Flag word contains the Instruction Field the program tried to jump to. There is no indication of what field the JMP/JMS came from. If the user wishes to locate the JMP/JMS error, all existing virtual fields must be examined at the location contained in the PC for a JMP or JMS.
?JOBMAX EXCEEDED TRY LATER	The user cannot LOGIN or execute any commands that require the use of a temporary job because the maximum number of jobs that the system can handle are already running. Someone must LOGOUT before any new jobs can be initiated.
?LOGIN PLEASE	The terminal is currently logged out.
n? FUNCTION FAILURE	A channel I/O command had an error. See 3.13 for an explanation of the error code n.
?NO SWAP TRACKS AVAILABLE	The system requires 8K of virtual memory to run LOGIN. This space was not available. Other users must release it before this user can LOG IN.
?NON EXISTENT FIELD	The virtual machine does not have that memory field available for use. Re-issue the CORE command to allocate more memory.
?NOT ENOUGH SWAP TRACKS	There was not enough free virtual memory to accommodate the desired expansion. Another

Error Message	Meaning
?NOT PRIVILEGED	user must release some virtual memory.
?PLEASE TYPE <^V>S<CR>FIRST	The command issued is not allowed unless the account is privileged.
?SYSTEM SWAP ERROR	The user issued one of the commands mentioned in 3.15 while the VM was running. The request is ignored.
?SYSTEM R/W ERROR	Disk error during swap.
?UNKNOWN COMMAND	System device error. Retry.
?UNKNOWN DEVICE	SCALE did not recognize the command keyword. This error can be baffling when the user has embedded a nonprinting character such as ^Q or ^O in the command.
?%VM HLT	The given device name is illegal or unknown.
?WAIT FOR I/O	A HLT instruction was executed, stopping the virtual machine.
?WASN'T ASSIGNED	The Monitor cannot execute the command at this time.
?WHAT? WHERE?	The user attempted to DEASSIGN a device that was not assigned to his job.
	(1) An invalid command was entered; or (2) a required argument for the DEPOSIT or EXAMINE command was not entered (address or data for DEPOSIT, address for EXAMINE); or (3) a nonexistent memory field was referenced by some command; or (4) the programmed SYSCAL function CORE deallocated the VM's instruction or data field (see 6.7.13).



APPENDIX C  
OS/8 ERROR MESSAGES

This appendix is provided for your convenience. Error messages generated by OS/8 programs appear in alphabetical order and are identified by the system program by which they are generated. This appendix is only a summary. Refer to the appropriate manuals for more detailed information about error conditions.

TABLE C-1  
OS/8 ERROR MESSAGES

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
?0	SRCCOM	Insufficient core--this means that the differences between the files are too large to allow for effective comparison. Use of the /X option may alleviate this problem.
0	Editor	Editor failed in reading a device. Error occurred in device handler; most likely a hardware malfunction.
0000	Linking Loader	/I or /O specified too late.
0001	Linking Loader	Symbol table overflow; more than 64 subprogram names.
0002	Linking Loader	Program will not fit into core.
0003	Linking Loader	Program with largest common storage area was not loaded first.
0004	Linking Loader	Checksum error in input tape.
0005	Linking Loader	Illegal relocation code.

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
0006	Linking Loader	An output error has occurred.
0007	Linking Loader	An input error has occurred (either a physical device error, or an attempt was made to read from a write-only device such as LPT:).
0010	Linking Loader	No starting address has been specified and there is no entry point named MAIN.
0011	Linking Loader	An error occurred while the Loader was attempting to load a device handler.
0012	Linking Loader	I/O error on system device.
?1	SRCCOM	Input error on file #1 or less than two input files specified.
1	Editor	Editor failed in writing onto a device. Generally a hardware malfunction or WRITE-LOCKed device.
1	TECO	Illegal command.
?2	SRCCOM	Input error on file #2.
2	Editor	File close error occurred. The output file could not be closed; the file does not exist on that device.
2	TECO	Incomplete command.
2045 REFS	CREF	More than 2044 (decimal) references to one symbol were made.
?3	SRCCOM	Output file too large for output device.

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
3	Editor	File open error occurred. This error occurs if the output device is a read-only device or if no output file name is specified on a file-oriented output device.
3	TECO	Nonalphanumeric Q-register name.
?4	SRCCOM	Output error.
4	Editor	Device handler error occurred. The Editor could not load the device handler for the specified device.
4	TECO	Command iterations or macro calls nested too deeply.
?5	SRCCOM	Could not create output file.
5	TECO	Text buffer overflow.
6	TECO	Search string longer than 31 characters.
7	TECO	Numeric argument missing before comma, equal sign, U, or quote (").
8	TECO	Illegal filename in ER, EW, or EB command.
9	TECO	Semicolon or failing search encountered on command level.
10	TECO	Iteration close (>) without matching open (<).
11	TECO	Attempt to move pointer from text buffer.
12	TECO	Q-register storage overflow.

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
13	TECO	Incomplete command.
14	TECO	Output file too large, or else output parity error.
15	TECO	Input file parity error.
16	TECO	File error.
17	TECO	An output command was encountered which would have caused TECO to overflow its current output file. Users should close the current output file and write all further output onto one (or more) additional files. These files may be combined if necessary.
18	TECO	Attempt to execute an output command without opening an output file.
A	SABR	Too many or too few ARG statements follow a CALL statement.
AA	F4	More than six subroutine arguments are arrays.
ALOG	FORT Library	Attempt to compute log of negative number.
ALREADY EXITS (filename)	FOTP	An attempt was made to rename an output file with the name of an existing output file.
ARE YOU SURE?	PIP	Occurs when using the /S option. A response of Y will compress the files.
ARITHMETIC EXPRESSION TOO COMPLEX	FORT	Self-explanatory.
AS	F4	Bad ASSIGN statement.
?BAD ARG	BUILD	No device name was included in the LOAD.



<u>Message</u>	<u>Program</u>	<u>Explanation</u>
BAD ARG	FRTS	Illegal argument to library function.
BAD ARGS	Keyboard Monitor	The arguments to the SAVE command are not consistent and violate restrictions.
BAD-BLK	EPIC	When EPIC is punching a patch it checks the block specified by =n to see if it is within range. If the block is out of range EPIC outputs this error message and returns to the command decoder.
BAD CKECKSUM, FILE #n	ABSLDR	File number n of the input file list has a checksum error.
BAD CHECKSUM, FILE #n	BITMAP	File number n of the input file list had a checksum error.
BAD CORE IMAGE	Keyboard Monitor	The file requested was not a core image file.
BAD DATE	Keyboard Monitor	The date has not been entered correctly, or incorrect arguments were used, or the date was out of range.
BAD DEVICE	CCL	The device specified in a CCL command is not of the correct form.
?BAD DIRECTORY	RESORC	Input device directory cannot be read.
BAD DIRECTORY ON DEVICE #n	PIP	Error message occurs when: <ul style="list-style-type: none"> <li>1. PIP is trying to read the directory, but it is not an OS/8 directory.</li> <li>2. The output device does not have a</li> </ul>

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
		system directory; i.e., file storage begins at record 7 (occurs during a /y transfer).
		n is the number of the file in the input file list.
BAD EXTENSION	CCL	Either an extension was specified without a file name or two extensions were specified.
BAD FORMAT OR CHECKSUM—TRY AGAIN	LIBSET	Error in reading relocatable binary file.
?BAD INPUT	BUILD	An error was detected in the binary file; it is not a proper input for the LOAD command.
BAD INPUT DIRECTORY	DIRECT, FOTP	The directory on the specified input device is not a valid OS/8 directory.
BAD INPUT FILE	Loader	An input file was not a RALF module.
BAD INPUT, FILE #n	ABSLDR	Attempt was made to load a nonbinary file as file number n of the input file list; or a non-core image with /I option.
BAD INPUT, FILE #n	BITMAP	A physical end of file was reached before a logical end of file, or extraneous characters were found in binary file n.
#BAD LINE. JOB ABORTED	BATCH	The BATCH monitor detected a record in the input file that did not have one of the characters dot, slash, dollar sign, or asterisk as the first character of the record. The

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
		record is ignored, and BATCH scans the input file for the next \$JOB record.
?BAD LOAD	BUILD	An attempt was made to load a binary handler that is not in the correct format.
?BAD MONITOR	RESORC	The input device may be a system device but the Monitor cannot be accessed.
BAD MONITOR	CCL	The version of the Keyboard Monitor being used is not compatible with CCL. A newer version of the monitor must be obtained from DIGITAL before CCL can be used.
BAD NUMBER	CCL	A CCL command that uses the # construction does not have the full 16-digit specification required.
?BAD ORIGIN	BUILD	The origin in a binary file is not in the range 200-577.
BAD OUTPUT DEVICE	FOTP	This message usually appears when a non-file structured device is specified as the output device.
BAD OUTPUT DEVICE	Loader	The loader image file device was not a directory device, or the symbol map file device was a read-only device. The entire line is ignored.
BAD OUTPUT DIRECTORY	FOTP	The directory on the specified output device is not a valid OS/8 device directory.

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
BAD RECOLLECTION	CCL	An attempt was made to use a previously remembered argument when no argument was saved.
BAD SWITCH OPTION	CCL	The character used with a slash (/) to indicate an option is not a legal option.
BAD SYSTEM HEAD	PIP	If the /Y option is used and the area being transferred does not contain OS/8, this message results.
BATCH.SV NOT FOUND ON SYS:	BATCH	A copy of BATCH.SV must exist on the system device. Control returns to the OS/8 Monitor.
BD	F4	Bad dimensions (too big, or syntax) in DIMENSION, COMMON, or TYPE declaration.
BE	FLAP/RALF	Illegal equate. The symbol had been defined previously.
BE	PAL8	Two PAL8 internal tables have overlapped. Fatal error--assembly cannot continue.
BI	FLAP/RALF	Illegal index register specification.
BO	FRTS	No more file buffer available.
BS	FR	Illegal in BLOCK DATA program.

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
BX	FLAP/RALF	Bad expression. Something in the expression is incorrect, or the expression is not valid in this context.
C	SABR	An illegal character appears on the line.
CANNOT CHANGE CORE CAPACITY WHILE RUNNING BATCH	CCL	A CORE command was issued while the BATCH program was running.
CANNOT HANDLE VARIABLE LENGTH RECORDS	MCPIP	The records on the input and output files specified are not the same size.
% CAN'T--AT BOF	CAMP	A file mark was read before the specified number of records were read over in a BACKSPACE command. The device is moved forward so that it is positioned at the beginning of the file.
? CAN'T--AT BOT	CAMP	A BACKSPACE command cannot move the device backward the specified number of files because the device is positioned at the beginning of the first file.
% CAN'T-AT EOD	CAMP	The specified number of files cannot be advanced over because the end of data was encountered. The tape is positioned at the end of data.
% CAN'T-AT EOF	CAMP	A file mark was read before the specified number of records were advanced over in a SKIP command. The tape is moved backward one record to leave it positioned at the end of

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
		the file.
? CAN'T--DEVICE DOESN'T EXIST	CAMP and SET	The device specified in a CAMP or SET command is not present on the OS/8 system.
? CAN'T--DEVICE IS READY-ONLY	CAMP	The device specified in a CAMP command is a read-only device; e.g., PTR.
? CAN'T--DEVICE IS RESIDENT	SET	No modifications are allowed to the system handler.
? CAN'T--DEVICE IS WRITE-ONLY	CAMP	The device specified in a CAMP command is a write-only device; e.g., TTY.
? CAN'T FOR THIS DEVICE	CAMP	The operation specified does not make sense for the device specified.
? CAN'T I/O ERROR	CAMP	This message is followed by a brief explanation of the input/output error that occurred.
CAN'T LOAD INPUT DEVICE	RXCOPY	The name of the input device specified in the command line is not a permanent device name.
CAN'T LOAD OUTPUT DEVICE	RXCOPY	The name of the output device specified in the command line is not a permanent device name.
? CAN'T--OBSOLETE HANDLER	SET	The handler has an old version number.
CAN'T OPEN OUTPUT FILE	PIP	Message occurs due to one of the following: <ul style="list-style-type: none"> <li>1. Output file is on a read-only device.</li> <li>2. No name has been specified for the output file.</li> </ul>

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
		3. A /Y transfer has been attempted to a non-directory device.
		4. Output file has zero free blocks.
CAN'T READ IT	FRTS	I/O error on reading loader image file.
% CAN'T REMEMBER	CCL	The argument specified in a CCL command line is too long to be remembered or an I/O error occurred.
? CAN'T--UNKNOWN VERSION OF THIS HANDLER	SET	The version of the handler is one not recognized, possibly because it is a newer version.
CAUTION-DO DP	FRTS	The present hardware configuration does not include an FPP-12 Floating-Point Processor with double-precision option.
CCL #x OVERLAY & MONITOR INCOMPATIBLE	CCL	The version of CCL being used is not compatible with the Keyboard Monitor present on the system. Type R CCL to retry.
CH	BCOMP	Error in CHAIN statement.
CF	PAL8	Chain to CREF error--CREF-SV was not found on SYS:.
CHER	FORT Library	File specified as argument to CHAIN not found on system device.
CI	BRTS	Inquire failure in CHAIN. Device not found.
CL	BRTS	Lookup failure in CHAIN.

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
		Filename not found.
CL	F4	Bad COMPLEX literal.
CLOSE ERROR	MCPIP	MCPIP is not able to close the file. A bad file just created on magnetic tape or cassette must be removed by placing a sentinel file after the preceding file.
CLOSE FAILED	CRED	CLOSE on output file failed.
CO	F4	Syntax error in COMMON statement.
COMMAND LINE OVERFLOW	CCL	The command line specified with the @ construction is more than 512 characters in length.
COMMAND TOO LONG	CCL	The length of a text argument in a MUNG command is too long.
COMPARE ERROR	RXCOPY	When using the /m option all the areas that do not match are printed as COMPARE ERRORS. Since this is a nonfatal error, the RXCOPY operation continues.
COMPILER MALFUNCTION	FORT	The meaning of this message has been extended to cover various unlikely Monitor errors.
CONTRADICTION SWITCHES	CCL	Either two CCL processor switches were specified in the same command line or the file extension and the processor switch do not agree.
?CORE	BUILD	A CORE command specified more memory than is



<u>Message</u>	<u>Program</u>	<u>Explanation</u>
		physically available, or the BOOT command was issued on an 8K system with a two-page system handler active. Two-page system handlers require at least 12K of core to be present on the OS/8 system.
CX	BCOMP	Incompatible file extensions were used in BASIC CHAIN statement.
D	SABR	A device handler has returned a fatal condition.
DA	BRTS	Attempt to read past end of data list.
DA	F4	Bad syntax in DATA statement.
DE	BCOMP	Error in DEF statement.
DE	BRTS	Device driver error. Caused by hardware I/O failure.
DE	F4	This type of statement illegal as end of DO loop.
DE	PAL8	Device error. An error was detected when trying to read or write a device. Fatal error--assembly cannot continue.
DELETES PERFORMED ONLY ON INPUT DEVICE GROUP 1 CAN'T HANDLE MULTIPLE DEVICE DELETES	FOTP	More than one input device was specified with the /D option when no output specification (device or filename) was included.
?DEV IS NOT FILE STRUCTURE	RESORC	The input device specified is not a file structured device; e.g.,

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
		PTR.
DEV LPT BAD	CREP	The default output device, LPT, cannot be used because it is not available on this system.
DEV NOT IMPLEMENTED	BATCH	BATCH cannot accept input from the specified input device because its handler is not permanently resident (SYS: or coresident with SYS:). Control returns to the Command Decoder.
DEVICE DOES NOT HAVE A DIRECTORY	DIRECT	The input device is a nondirectory device; that is, PTR, DIRECT can only read directories from file structured devices.
DEVICE FULL	PIP10	DECsystem-10 ran out of space on the output file during a transfer.
DEVICE #n NOT A DIRECTORY DEVICE	PIP	<p>Message occurs when:</p> <ol style="list-style-type: none"> <li>1. Trying to list the directory of a nondirectory device.</li> <li>2. The input designated in a /Y transfer is not on a directory device.</li> </ol> <p>n gives the number of the device in the input list</p>
DF	PAL8	Device full. Fatal error--assembly cannot continue.
DF	F4	Bad DFINE FILE statement.
D.F. TOO BIG	FRTS	Product of number of records times number of blocks per record

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
		exceeds number of blocks in file.
DH	F4	Hollerith field error in DATA statement.
DI	BCOMP	Error in DIM statement syntax or string dimension greater than 72, or array dimensioned twice.
DIRECTORY ERROR	PIP	An error has occurred while reading or writing the directory during a /S option.
DIVIDE BY 0	FRTS	Attempt to divide by zero. The resulting quotient is set to zero and execution continues.
DIVZ	FORT Library	Division by zero; very large number is returned.
DL	F4	Data list and variable list are not same length.
DN	F4	DO-end missing or incorrectly nested. This message is not printed during pass 3. It is followed by the statement number of the erroneous statement rather than the ISN.
DO	BRTS	No more room for drivers. Too many different devices used in file commands.
DO	F4	Syntax error in DO or implied DO.
name DOES NOT EXIT	CCL, Command Decoder, MCPIP	The device with the name given is not present on the OS/8 system.
DP	F4	DO loop parameter not integer or real.

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
?DSK	BUILD	The device specified in a DSK command is not a file structured device.
DV	BRTS	Attempt to divide by 0. Result is set to zero (NF).
DV	FLAP/RALF	An attempt was made in an expression evaluation to divide by zero.
E	SABR	There is no END statement.
EF	BRTS	Logical end of file. Usually caused when I/O device runs out of medium.
EG	FLAP	The preceding line contains extra code that could not be used by the assembler.
EM	BRTS	Attempt to exponentiate a negative number to a power.
EN	BRTS	Enter error in opening file. Device is read-only, or there is already one variable file open on that device, or file not found.
END OF TAPE	EPIC	EPIC was expecting a block of tape and found end of tape instead. Press CONT to retry.
END OF TAPE ENTER NEXT	EPIC	When EPIC is reading a file that is segmented across a number of paper tapes and encounters the end of a segment, it outputs this message and halts with AC=7777 to allow the user to enter the next segment of paper tape.

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
		Press CONT to continue reading.
ENTER ERROR	MCPIP	Error occurred while trying to enter an output file. This message usually means that the cassette or magnetic tape has no sentinel file.
ENTER FAILED	CREP	Entering an output file was unsuccessful--possibly output was specified to a read-only device.
EOF ERROR	FRTS	End of file encountered on input.
EQUALS OPTION BAD	DIRECT	The =n option is not in the range 0-7.
ERROR CLOSING FILE	DIRECT	System error.
ERROR DELETING FILE	PIP, PIP10	An attempt was made to delete a file that does not exist.
ERROR IN COMMAND	CCL	A command not entered directly from the console terminal is not a legal CCL command. This error occurs when the argument of a UA, UB, or UC command was not a legal command.
ERROR ON INPUT DEVICE SKIPPING (filename)	FOTP	The file specified is not transferred, but any previous or subsequent files are transferred and indicated in the new directory.
ERROR ON OUTPUT DEVICE	BITMAP	Error occurred while writing on output device; i.e., output error on DECTape write.
ERROR ON OUTPUT DEVICE	FOTP	

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
SKIPPING (filename)		The file specified is not transferred, but any previous or subsequent files are transferred and indicated in the new directory.
ERROR READING INPUT DIRECTORY	DIRECT, FOTP	An error occurred while reading the directory.
ERROR WHILE WRITING OUTPUT FILE	LIBSET	Fatal output error occurred.
ERROR WRITING FILE	DIRECT	An error occurred while writing the output file.
ERROR WRITING OUTPUT DIRECTORY	FOTP	Self-explanatory.
ES	RALF	External symbol error.
EX	F4	Syntax error in EXTERNAL statement.
EXCESSIVE SUBSCRIPTS	FORT	Self-explanatory.
FB	FORT Library	Argument to EXP too large; very large number is returned.
FB	BRTS	FILE busy. Attempt to use a file already in use.
FC	BRTS	OS/8 error while closing variable file. Device is read-only on file already closed.
FE	BRTS	Fetch error in opening file. Device not found, or device handler too big for available space.
FETCH ERROR	MCPIP	Error occurred while trying to fetch an OS/8 device handler.
FI	BRTS	Attempt to close or use unopened file.
FILE ERROR	FRTS	Any of the following:

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
		<ul style="list-style-type: none"> <li>a. A file specified as an existing file was not found.</li> <li>b. A file specified as a nonexistent file would not fit on the designated device.</li> <li>c. More than one nonexistent file was specified on a single device.</li> <li>d. File specification contained "*" as name or extension.</li> </ul>
FILE NOT FOUND	PIPl0	The requested file was not found on the specified device.
FILE OVERFLOW	FRTS	Attempt to write outside file boundaries.
FIX	FORT Library	Attempt to fix a number >2047; 2047 is returned.
FL	FLAP/RALF	An error has occurred in the FPP or software floating conversion routines.
FLPW	FORT Library	Negative number raised to floating point power; absolute value taken.
FM	BRTS	Attempt to fix minus number. Usually caused by negative subscripts or file numbers.
FMT1	FORT Library	Invalid format statement.
FMT2	FORT Library	Illegal character in I format.
FMT3	FORT Library	Illegal character in F or E format.
FN	BCOMP	Error in file number of file name designation.
FN	BRTS	Illegal file number. Only 0, 1, 2, 3, 4 are

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
		legal.
FO	BRTS	Attempt to fix number greater than 4095. Usually caused by negative subscripts of the file numbers.
FORMAT ERROR	FRTS	Illegal syntax in FORMAT statement.
FP	BCOMP	Incorrect FOR loop parameters or FOR loop syntax.
FP	FLAP	A syntax error was encountered in a floating-point or extended-precision constant.
FPP ERROR	FRTS	Hardware error on FPP startup.
FR	BCOMP	Error in function arguments or function not defined.
FULL	Editor	The specified output device has become full. The file is closed; the user must specify a new output file.
GR	BRTS	RETURN without a GOSUB.
GS	BRTS	Too many nested GOSUBS. The limit is 10.
GT	F4	Syntax error in GO TO statement.
GV	F4	Assigned or computed GO TO variable must be integer or real.
HANDLER FAIL	CREF	This is a fatal error on output and can occur if either the system device or the selected output device is WRITE-LOCKed.



<u>Message</u>	<u>Program</u>	<u>Explanation</u>
?HANDLERS	BUILD	More than 15 handlers, including SYS and DSK, were active when a BOOT command was issued.
HO	F4	Hollerith field error.
I	SABR	Illegal syntax.
IA	BRTS	Illegal argument in UDEF function call.
IC	FLAP/RALF	The symbol or expression in a conditional is improperly used, or left angle bracket is missing. The conditional pseudo-op is ignored.
IC	PAL8	Illegal character. The character is ignored and the assembly continued.
ID	PAL8	Illegal redefinition of a symbol.
IE	F4	Error reading input file. Control returns to the Keyboard Monitor.
IE	PAL8	Illegal equals--an attempt was made to equate a variable to an expression containing an undefined term. The variable remains undefined.
IE	RALF	an entry point has not been defined, or is absolute, or also is defined as a common section, or external.
IF	BCOMP	THEN or GOTO missing from IF statement, or bad relational operator.
IF	BRTS	Illegal DEV:filename specification.

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
IF	F4	Logical IF statement cannot be used with DO, DATA, INTEGER, etc.
II	PAL8	Illegal indirect--an off-page reference was made; a link could not be generated because the indirect bit was already set.
IL	FLAP	A literal was used in an instruction that cannot accept one.
ILLEGAL*	DIRECT, FOTP	An asterisk (*) was included in the output file specification, or an illegal * was included in the input file name.
ILLEGAL * OR ?	CCL, MCPIP	An * or ? was used in a CCL command that does not accept the wild card construction.
ILLEGAL?	DIRECT, FOTP	A question mark (?) was included in the output file specification.
ILLEGAL ARG.	Keyboard Monitor	The SAVE command was not expressed correctly; illegal syntax used.
ILLEGAL ARITHMETIC EXPRESSION	FORT	Self-explanatory.
ILLEGAL BINARY INPUT, FILE #n	PIP	Self-explanatory; n is the number of the file in the input list.
ILLEGAL CONSTANT	FORT	Self-explanatory.
ILLEGAL CONTINUATION	FORT	Self-explanatory.
ILLEGAL EQUIVALENCING	FORT	Self-explanatory.
#ILLEGAL INPUT	BATCH	A file specification designated TTY or PTR as an input device when the initial dialogue indicated that an

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
		operator is not available. The current job is aborted, and BATCH scans the input file for the next \$JOB command record.
ILLEGAL OR EXCESSIVE DO NESTING	FORT	Self-explanatory.
ILLEGAL ORIGIN	Loader	A RALF routine tried to store data outside the bounds of its overlay.
ILLEGAL SPOOL DEVICE	BATCH	The device specified as a spooling output device must be file structured. Control returns to the Command Decoder.
ILLEGAL STATEMENT	FORT	Self-explanatory.
ILLEGAL STATEMENT NUMBER	FORT	Self-explanatory.
ILLEGAL SYNTAX	CCL, Command Decoder, MCPIP	The command line was formatted incorrectly.
?ILLEGAL WIDTH	SET	A width of 0 or a width too large was specified, or, for the TTY, a width of 128 or one not a multiple of 8 was specified.
ILLEGAL VARIABLE	FORT	Self-explanatory.
IN	BRTS	Inquire failure in opening file. Device not found.
INCOMPATIBLE?	ABSLDR	The versions of ABSLDR and the Keyboard Monitor being used are incompatible.
INPUT DEVICE READ ERROR	RXCOPY	Bad input, bad tracks or sectors. Since this is a nonfatal error, the RXCOPY operation continues.

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
?INPUT ERROR	RESORC	An input error occurred during a RESORC operation.
INPUT ERROR	CREF, MCPIP	An input error occurred while reading the file.
INPUT ERROR	FRTS	Illegal character received as input.
INPUT ERROR	LIBSET	Parity error on input.
INPUT ERROR, FILE #n	PIP	An input error occurred while reading file number n in the input file list.
INPUT ERROR READING INDIRECT FILE	CCL	CCL cannot read the file specified with the @ construction.
#INPUT FAILURE	BATCH	Either a hardware problem prevented BATCH from reading the next record of the input file, or BATCH read the last record of the input file without encountering a \$END command record.
INSUFFICIENT CORE FOR BATCH RUN	BATCH	OS/8 BATCH requires 12K of core to run. Control returns to the OS/8 Monitor.
IO	BCOMP	I/O error.
IO	BRTS	TTY input buffer overflow. Causes input buffer to be cleared and output another ? (NF).
IO	FLAP/RALF	Input/output error (fatal error).
IO	FORT	A device handler has signaled an I/O FORT error.
IOER	FORT Library	One of the following has occurred:

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
		<ol style="list-style-type: none"> <li>1. Device-independent input or output attempted without /I or /O options, or user attempted to specify a device requiring a two-page handler for device-independent I/O without using the /H option.</li> <li>2. Bad arguments to IOPEN or OOPEN.</li> <li>3. Transmission error while doing I/O.</li> </ol>
I/O ERR	BUILD	An error occurred while reading from an input device during a LOAD command.
I/O ERROR	FRTS	Error reading or writing a file: tried to read from an output device, or tried to write on an output device.
I/O ERROR	PIP10	I/O device error; e.g., parity, write-lock, out of paper.
I/O ERROR	EPIC	If EPIC encounters an error while reading or writing a mass storage device, or a paper tape read fails three consecutive times, it outputs this error message, deletes the output file if one exists, and returns to the Command Decoder.
I/O ERROR, FILE #n	ABSLDR, BITMAP	An I/O error has occurred in input file number n.
I/O ERROR IN (file name)-- CONTINUING	PIP	An error has occurred during a /S transfer.
I/O ERROR ON SYS:	CCL	An error occurred while

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
		doing I/O to the system device. The system must be restarted at 7600 or 7605. Do not press CONT, because that will cause further errors.
I/O ERROR TRYING TO RECALL	CCL	An I/O error occurred while CCL was trying to remember an argument.
IP	PAL8	Illegal pseudo-op--a pseudo-op was used in the wrong context or with incorrect syntax.
IR	FLAP	Invalid reference in a PDP-8 instruction.
IX	FLAP/RALF	An index register was specified for an instruction that cannot accept one.
IZ	PAL8	Illegal page zero reference--The pseudo-op was found in an instruction that did not refer to page zero. The Z is ignored.
L	SABR	/L or /G option was indicated, but the LOADER.SV file does not exist on the system device.
LD	PAL8	The /L or /G optins have been specified and ABSLDR is not present on the system.
LG	PAL8	Link Generated--only printed if the /E switch was specified to PAL8.
LI	F4	Argument of logical IF is not type Logical.
LIBRARY DIRECTORY OVERFLOW	LIBSET	Too many subroutines were specified.

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
LINE TOO LONG IN FILE #n	PIP	In ASCII mode, a line has been found greater than 140 characters.
LM	BRTS	Attempt to take log of negative number or 0.
LOADER I/O ERROR	Loader	Fatal error message during aUSR function.
LS	BCOMP	Missing equal sign in LET statement.
LT	BCOMP	Statement too long (greater than 80 characters).
LT	F4	Input line too long, too many continuations.
LT	FLAP/RALF	The line is longer than 128 characters. The first 127 characters are assembled and listed.
L/T ERROR	EPIC	EPIC was expecting leader/trailer and found non-leader/trailer while attempting to read a block. The program prints this error message and halts with AC=777 to allow the user to reposition the tape then press the CONT key.
M	SABR	A symbol has multiple definitions. Listings of programs with multiple definitions have unmarked errors.
#MANUAL HELP NEEDED	BATCH	BATCH is attempting to operate an I/O device, such as PRT or TTY, that will require operator intervention.
MD YY	BCOMP	Line number defined more than one. YY equals the line number before line in error.

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
MD	FLAP/RALE	The tag on the line has been previously encountered at another location or has been used in a context requiring an absolute expression.
ME	BCOMP	Missing END statement.
MIXED INPUT	Loader	The L option was specified on a line that contained some file other than a library file. The library file (if any) is accepted. Any other input file specification is ignored.
MIXED MODE EXPRESSION	FORT	Self-explanatory.
MK	F4	Misspelled keyword.
ML	F4	Line number with multiple definitions.
MM	F4	Mismatched parenthesis.
MO	BCOMP, F4	Operand expected, not found.
MONITOR ERROR 2 AT xxxx (DIRECTORY I/O ERROR	Keyboard Monitor	Attempt made to output to a WRITE-LOCKed device, usually DECTape; or an error has occurred reading/writing a directory.
MONITOR ERROR 5 AT xxxx (I/O ERROR ON SYS:)	Keyboard Monitor	An error occurred while doing I/O to the system device. This error is normally the result of not WRITE-ENABLEing the system device.
MONITOR ERROR 6 AT xxxx (DIRECTORY OVERFLOW)	Keyboard Monitor	A directory overflow has occurred (no room for tentative file entry in directory).



<u>Message</u>	<u>Program</u>	<u>Explanation</u>
#MONITOR OVERLAYED	BATCH	The Command Decoder attempted to call the BATCH monitor to accept and transmit a file specification, but found that a user program had overlaid part or all of the BATCH monitor. Control returns to the monitor level, and BATCH executes the next Keyboard Monitor command.
MORE CORE REQUIRED	FRTS	The space required for the program the I/O device handlers (I/O buffers), and the resident Monitor exceeds the available core.
MP	BCOMP	Missing parenthesis or error in expression within parentheses.
MT	BCOMP, F4	Operand of mixed type or operator does not match operands.
MULT SECT	Loader	Any combination of entry point, COMMON section (with the exception of multiple COMMONs) or program section of the same name causes this error, except as shown in the FORTRAN IV loader error messages table.
?NAME	BUILD	A device or filename was not designated in a command that requires one to be present.
NE	FLAP/RALF	Number error. A number out of range was specified, or an 8 or 9 occurred in octal radix.
NEED:n1 FOUND:n2	EPIC	EPIC read block n2 of the file when it was expecting block n1 of the file. EPIC halts

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
		with AC+7777 to allow the user to reposition the paper tape.
NEED:name1 FOUND:name2	EPIC	EPIC read a block of tape for the file NAME2 when it was expecting a block of the file NAME1.
NF	BCOMP	NEXT statement without corresponding FOR statement.
NM	BCOMP	Line number missing after GOTO, GOSUB, or THEN.
NO!!	Keyboard Monitor	The user attempted to start (with .ST) a program that cannot be started.
NO CCL!	Keyboard Monitor	CCL.SV is not present on the system device, or an I/O error occurred on the file. Refer to the OS/8 System Generation Notes for instructions on loading programs onto the system device.
NO DEFINE FILE	FRTS	Direct access I/O attempted without a DEFINE FILE statement.
NO END STATEMENT	FORT	The input to the compiler has been exhausted.
NO FILES OF THE FORM xxxx	FOTP	No files of the form (xxxx) specified were found on the current input device group.
NO /I	BITMAP	Cannot produce a bitmap of an image file.
NO /I!	ABSLDR	Use of /I is prohibited at this point.

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
NO INPUT	ABSLDR, BITMAP	No input or binary file was found on the designated device.
NO INPUT DEVICE	RXCOPY	No input device was specified.
NO INPUT FILE	MCPIP	No input file was specified when one was required.
NO MAIN	LOADER	No RALF module contained section #MAIN.
NO NUMERIC SWITCH	FRTS	The referenced FORTRAN I/O unit was not specified to the run-time system.
NO OUTPUT FILE	MCPIP	No output file was specified when one was required.
?NO ROOM	BUILD	Too many device handlers were present on the system when a LOAD or BUILD command was typed. The UNLOAD command must be used to remove a handler before another can be loaded.
NO ROOM FOR OUTPUT	FORT	The file FORTRN.TM cannot fit on the system device.
NO ROOM FOR OUTPUT FILE	DIRECT, PIP	Either room on device or room in directory is lacking.
NO ROOM IN (filename)--CONTINUING	PIP	Occurs during use of the /S option. The output device cannot contain all of the files on the input device.
NO ROOM, SKIPPING (filename)	FOTP	No space is available on the output device to perform the transfer. Predeletion may already have occurred.

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
NO SUCH DEVICE	PIPl0	Device name used is not legal in this OS/8 system.
% NON SYSTEM DEVICE	RESORC	The input device specified in a RESORC command line is not an OS/8 system device.
NOT A LOADER IMAGE	FRTS	The first input file specified to the run-time system was not a loader image file.
% NOT A SYSTEM HEAD	RESORC	The filename specified is not a system-head file.
name NOT AVAILABLE	Keyboard Monitor	The device with the name given is not listed in any system table, or it is not available for use at the moment, or the user tried to obtain input from an output-only device.
NOT ENOUGH CORE	CCL	The number specified in a CORE command is larger than the number of 4K core banks on the system.
name NOT FOUND	BUILD, CCL, Command Decoder, Keyboard Monitor	The device or file name designated in the command was not found.
file NOT FOUND	MCPIP	The file specified cannot be found.
NOT OS8 FILE	PIPl0	The output device specified with a /L or /F option was not an OS/8 device or file.
NOT PDP-10 FILE	PDP10	The output device specified with a /Z option was not a DECsystem-10 tape, or the input device specified with a /L or

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
		/F option was not a DECsystem-10 tape.
?NUMBER TOO BIG	CAMP	The "nnnn" specified in a BACKSPACE or SKIP command is greater than 4095.
?NUMBER TOO BIG	SET	The number specified was out of range.
OE	BRTS	Driver error while overlaying. Caused by SYS device hardware error.
OF	BCOMP	Output file error.
OF	F4	Error writing output file. Control returns to the Keyboard Monitor.
OP	F4	Illegal operator.
OS/8 ENTER ERROR	Loader	Fatal error during a USR function.
OT	F4	Type/operator use illegal (e.g., A.AND.B where A and/or B not typed Logical).
OUT DEV FULL	CREP	The output device is full (directory devices only).
OUT-IN	MCPIP	Both the input and the output devices were specified as the same cassette or magnetic tape drive.
?OUTPUT DEVICE FULL	RESORC	The output device specified does not have enough room to copy the RESORC file.
OUTPUT DEVICE FULL	MCPIP	Either room on device or room in the directory is lacking.
?OUTPUT DEVICE IS READ ONLY	RESORC	The output device

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
		specified is a read-only device; e.g., PTR.
OUTPUT DEVICE READ ERROR	RXCOPY	Bad data on output device, tracks and sectors bad. Since this is a nonfatal error, the RXCOPY operation continues.
?OUTPUT ERROR	RESORC	An error occurred while attempting to output during a RESORC operation.
OUTPUT ERROR	MCPIP	Output error--possibly a WRITE-LOCKed device, parity error, or attempt to output to a read-only device.
OUTPUT FILE OPEN ERROR	PIP10	The output file could not be opened. Check output directory to ensure that enough space exists on the device.
OV	BRTS	Numeric or input overflow.
OVER CORE	Loader	The loader image requires more than 32K of core memory.
OVER IMAG	Loader	Output file overflows in the loader image file.
OVER SYMB	Loader	Symbol table overflow. More than 253 (decimal) symbols in one FORTRAN JOB.
OVERFLOW	FRTS	Result of a computation exceeds upper bound for that class of variable. The result is set equal to zero and execution continues. This error is detected only if an FPP is present.
OVERLAY ERROR	FRTS	Error while reading

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
		overlay.
OVFL	FORT Library	Floating point overflow; very large number is returned.
PA	BRTS	Illegal argument in POS functin.
PARENS TOO DEEP	FRTS	Parens nested too deeply in FORMAT statement.
PARTIY ERROR	EPIC	EPIC failed to read a block correctly; e.g., the reader dropped some bits. EPIC halts with AC=7777 to allow the user to reposition the tape so that it can try the read again.
PD	BCOMP	Pushdown stack overflow. Result of either too complex a statement (or statements) or too many nested FOR-NEXT loops.
PD	F4	Compiler stack overflow; statement too big and/or too many nested loops.
PE	PAL8	Current non-zero page exceeded.
PH	F4	Bad program header line.
PH	PAL8	Phase error--a conditional assembly bracket is still in effect at the end of the input stream--this is caused by nonmatching angle bracket (< >) characters in the source.
PIPlØ CANNOT BE CHAINED	PIPlØ	Self-explanatory.
?PLAT	BUILD	The =n in a SYS command is too large for the device specified; e.g., RFØ8=5.

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
PO	FLAP	Page overflow. Literals and instructions have been overlapped.
PREMATURE END OF FILE, FILE #n	PIP	Message occurs in Binary Mode (/B) only. A physical end-of-file has been found before the final leader/trailer.
PTR:name IS TOO BIG FOR dev:	EPIC	The paper tape file name will not fit on the specified output device DEV:. EPIC aborts the command and returns to the Command Decoder. EPIC makes the check for size before writing on the output device.
QL	F4	Nesting error in EQUIVALENCE statement.
QS	BCOMP	String literal too long or does not end in quote.
QS	F4	Syntax error in EQUIVALENCE statement.
RD	F4	Attempt to redefine the dimensions of a variable.
RD	PAL8	Redefinition--a permanent symbol has been defined with =. The new and old definitions do not match. The redefinition is allowed.
RE	BRTS	Attempt to read past end of file (NF).
RE	RALF	Relocatability error. A relocatable expression has been used in context requiring an absolute expression.
RECORD SIZE TOO BIG	MCPIP	The output record size specified is greater than 1000 or an output



<u>Message</u>	<u>Program</u>	<u>Explanation</u>
		record size is 0.
RT	F4	Attempt to redefine the type of variable.
RW	F4	Syntax error on READ/WRITE statement.
S	SABR	Either the symbol table has overflowed, common storage has been exhausted, more than 64 different user-defined symbols occurred in a core page, or more than 64 external symbols have been declared. Could also indicate a system error such as overflowed output file.
SABR.SV NOT FOUND	FORT	The SABR assembler is not present on the system device.
SAVE ERROR	Keyboard Monitor	An I/O error has occurred while saving the program. The program remains intact in core.
SC	BRTS	String too long (greater than 72 characters) after concatenating.
SE	PAL8	Symbol table extended-- too many symbols have been defined for the amount of memory available. Fatal error-- assembly cannot continue.
SF	F4	Bad arithmetic statement function.
SL	BRTS	String too long or undefined.
?SLOTS	BUILD	More than 8 groups of non-system handlers have been inserted.

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
SN	F4	Illegal subroutine name in CALL.
SORRY--NO INTERRUPTIONS	PIP	1. ^C (CTRL/C) is typed while compressing a file onto itself; the transfer continues.  2. A /Y transfer is done with system device as the output, or if the transfer has both input and output on the same device.
#SPOOL TO FILE BTCH1	BATCH	Where the "A" may be any character of the alphabet and the "1" may be any decimal digit. This message indicates that BATCH has intercepted a nonfile structured output file and rerouted it to the spool device. This is not, generally, an error condition.
SQRT	FORT Library	Attempt to take square root of negative number; absolute value used.
SR	BRTS	Attempt to read string from numeric file.
SS	BCOMP	Subscript or function argument error.
SS	F4	Error in subscript expression; i.e., wrong number, syntax.
ST	BCOMP	Symbol table overflow due to too many variables, line number, or literals. Combine lines using backslash (\) to condense program.
ST	BRTS	String truncation on input. Stores maximum length allowed (NF).

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
ST	F4	Compiler symbol table full, program too big. Causes an immediate return to the Keyboard Monitor.
ST	FLAP/RALF	User symbol table overflow (fatal error).
SU	BRTS	Subscript out of DIM statement range.
SUBR. OR FUNCT. STMT. NOT FIRST	FORT	Self-explanatory.
% SUPERSEDED	CCL	The file specified in a MAKE command already exists. This is a warning message indicating that the file is being replaced.
SW	BRTS	Attempt to write string into numeric file.
SWITCH NOT ALLOWED HERE	CCL	Either a CCL option was specified on the left side of the < or was used when not allowed.
SY	BCOMP	System incomplete. System files such as BASIC.SV, BCOMP.SV, and BRTS.SV missing.
SY	F4	System error; i.e., PASS20. SV or PASS2.SV missing, or no room for output file. Causes an immediate return to the Keyboard Monitor.
SYM OVERFLOW	CREP	More than 896 (decimal) symbols and literals were encountered.
SYMBOL TABLE EXCEEDED	FORT	Self-explanatory.
?SYNTAX	BUILD	An illegal character was typed in a BUILD command line. The line must be retyped.

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
?SYNTAX ERROR	CAMP	An illegal character was typed in a CAMP command or a command was formatted incorrectly. The command must be retyped.
?SYNTAX ERROR	SET	Incorrect format used in SET command or NO specified when not allowed.
SYNTAX ERROR	FORT, PIP 10	Invalid command line.
?SYS	BUILD	This message appears when one of the following conditions exists: <ul style="list-style-type: none"> <li>a. A permanent name in a SYS command was not a system handler or coresident with one.</li> <li>b. A BOOT command was issued when two system handlers were active.</li> <li>c. A BOOT command was issued when an active handler, which must be coresident with a SYS handler, did not have the system handler active.</li> </ul>
#SYS ERROR	BATCH	A hardware problem prevented BATCH from performing an I/O operation.
SYS ERROR	BUILD	An I/O error occurred with a system handler. The computer halts. Press CONT to retry or restart the BUILD procedure from the beginning. Do not assume that a valid OS/8 system remains in core.
SYS NOT FOUND	BUILD	No active handler with the name SYS was present when a BOOTSTRAP command

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
		was issued.
SYSTEM DEVICE ERROR	FRTS	I/O failure on the system device.
SYSTEM ERR	Keyboard Monitor	An error occurred while doing I/O to the system device. The system should be restarted at 7600 or 7605. Do not press CONTINUE because this will cause further errors.
SYSTEM ERROR	Loader	Fatal error during a USR function.
SYSTEM ERROR--CLOSING FILE	FOTP	Self-explanatory.
TB	BCOMP	Program too big. Condense or CHAIN.
TD	BCOMP	Too much data in program.
TD	F4	Bad syntax in type declaration statement.
THERE IS NO HOPE--THERE IS NO TTY HANDLER IN YOUR SYSTEM!	DIRECT	A command was issued to print a directory on the terminal when no TTY handler is present on the OS/8 system. Use BUILD to insert a TTY handler into the system.
TOO FEW ARGS	Keyboard Monitor	An important argument has been omitted from a command.
TOO MANY FILES	CCL	Too many files were included in a CCL command.
TOO MANY FILES	Command Decoder	More than three output files or nine input files were specified. Some programs may restrict the user to fewer files.
TOO MANY FILES	MCPIP	More than one output or

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
		one input device was specified.
TOO MANY HANDLERS	FRTS	Too many I/O device handlers are resident in memory, or files have been defined on too many devices.
TOO MANY LEVELS	Loader	The Ø option was specified more than seven times.
TOO MANY OVERLAYS	Loader	More than 16 overlays were defined in the current level.
TOO MANY RALF FILES	Loader	More than 128 input files were specified.
TS	BCOMP	Too many total characters in the string literals.
?TTY DOES NOT EXIST	RESORC	An output device was not specified in the RESORC command line and the TTY handler does not exist on the OS/8 system. Use BUILD to insert TTY handlers into the system.
U	SABR	No symbol table is being produced, but there is at least one undefined symbol in the program.
UD	BCCOMP	Error in UDEF statement.
UF	BCOMP	FOR loop without corresponding NEXT statement.
UNDF	SABR	Undefined symbol; printed in the symbol table listing.
UNIT ERROR	FRTS	I/O unit not assigned, or incapable of executing the requested operation.

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
?UNKNOWN ATTRIBUTE FOR DEVICE dev	SET	An illegal attribute was specified for the given device.
UO	PAL8	Undefined origin--an undefined symbol has occurred in an origin statement.
US	BCOMP, F4	Undefined statement number.
US	FLAP/RALF, PAL8	Undefined symbol in an expression.
USE PIP FOR NON-FILE STRUCTURED DEVICE	FOTP	An input device specified is not a file structured device; e.g., PTR.
USER ERROR	FRTS	Illegal subroutine call, or call to undefined subroutine. Execution continues only if the E option was requested.
USER ERROR 0 AT xxxx	Keyboard Monitor	An input error was detected while loading the program. xxxx refers to the Monitor location where the error was generated.
USER ERROR 1 AT xxxx	FORT	Library The user tried to reference an entry point of a program that was not loaded, or failed to define a subscripted variable in a DIMENSION statement. xxxx has no meaning.
USR n dev:name	EPIC	the USR encountered an error while attempting to perform a fetch, lookup, enter, or close on the file name on device dev. n=1 is a fetch, n=2 is lookup, n=3 is enter, n=4 is

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
		close. EPIC aborts the command and returns to the Command Decoder.
UU	BCOMP	Incorrect or missing array designator in USE statement.
VE	F4	Version error. One of the compiler programs is absent from SYS: or is present in the wrong version.
VR	BRTS	Attempt to read variable-length file.
WE	BRTS	Attempt to write past end of file (NF).
WRONG OS/8 MONITOR	BATCH	OS/8 BATCH requires an OS/8 Monitor no older than version 3.
XC	BCOMP	Extra characters after the logical end of line.
XS	RALF	External symbol table overflow. Control returns to the OS/8 Keyboard Monitor.
ZE	PAL8	Page 0 exceeded--same as PE except with reference to page 0.
ZERO SYS?	PIP	Attempt made to zero the system device directory. Responding with Y causes the directory to be zeroed; any other character aborts the operation.



APPENDIX D  
COS ERROR MESSAGES

This appendix is provided for your convenience. Error messages generated by COS programs appear in alphabetical order and are identified by the system program by which they are generated. This appendix is only a summary. Refer to the appropriate manuals for more detailed information about error conditions.

TABLE D-1  
COS ERROR MESSAGES

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
ALL UNITS ASSIGNED	SYSGEN	Attempt to assign more than 15 units.
BAD ALPHA FIELD	BUILD	Illegal entry in alpha field.
BAD ALPHA VALUE	BUILD COMP	Initial value in an alpha data specification did not begin or end with a single quotation.
BAD CHAR	CONVEX	Attempted to convert an OS/8 character for which there is no COS code.
BAD CHECKDIGIT	BUILD	Checkdigit calculated by BUILD does not match the one entered.
BAD CHECKSUM	PATCH	An attempt was made to write a block which was incorrectly patched.
BAD COMPILATION	Monitor	User tried to SAVE a compiled binary that had errors.
BAD DECIMAL FIELD	BUILD	Illegal entry in decimal field.

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
BAD DECIMAL VALUE	COMP	The initial value for a decimal data specification was incorrectly formed.
BAD DEFAULT FIELD	BUILD	Default field does not agree with size or type of object field.
BAD DELIM	BUILD	Fields are separated by an illegal delimiter.
BAD DEPICTOR	BUILD	OUTPUT format in error.
BAD DEV	CONVEX	Output device not same type physical device as system device.
BAD DEVICE	BUILD	Not KBD, SYS, RDR or CDR.
BAD DEVICE	UPDATE	Illegal device specified on INPUT statement.
BAD DIGIT AT LINE nnnn	Run Time	A character other than +, -, space, or the digits 0-9 was encountered in an alpha to decimal conversion; nnnn is the line number in which the error occurred.
BAD DIGIT IN DECIMAL INITIAL VALUE	BUILD, SORT, UPDATE	Alpha character in a decimal initial value.
BAD DIRECTORY	PIP, Monitor, PATCH CONVEX	Attempt to reference or store a file on a device with no directory (or a directory that has been destroyed). Also an attempt to get a directory of a data file.
BAD END STATEMENT	BUILD	Missing or misspelled END statement.
BAD FIELD NUMBER	BUILD	Field number specified is greater than 2047.
BAD INPUT STATEMENT	BUILD, UPDATE	Statement missing, misspelled, or out of order.

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
BAD KEY STMT	UPDATE	Key in KEY statement has bad syntax.
BAD KEYWORD	BUILD	Keyword entered was not described in the control program.
BAD LABEL	Monitor	Tape has no label, or its form is incorrect.
BAD NUMBER	PATCH	A number with more than 4 digits, a non-digit, or 8 or 9 was typed.
BAD OPTION	BUILD	Something other than C or + is specified in OUTPUT section.
BAD OUTPUT FORMAT #	BUILD	Format number is wrong or out of range (<0 or >7).
BAD PROC #	COMP	The number in a PROC statement was not a digit from 0 to 7.
BAD PROGRAM AT LINE nnnn	Run time	Attempting to run a binary program which contains a compilation error. Check compilation listing for error flags. Correct flagged errors, and recompile.
BAD RANGE CONSTRUCTION	BUILD	In RANGE portion of statement an open or close parenthesis or comma is missing.
BAD RECORD SIZE	SORT	File contains records of variable length.
BAD RELATIONAL	COMP	An illegal relational occurs in an IF statement. For example, a .GX. instead of a .GT.
BAD SWITCH	SYSGEN	Not /T, /C, or /L.
BAD UPDATE COMMAND	UPDATE	Bad syntax, probably missing F, =, or comma.

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
BAD UPDATE STMT	UPDATE	UPDATE statement is incorrect.
BAD VALUE	UPDATE	Value in update command too long or incorrect.
BAD WORK UNIT COUNT	SORT	# of work units not in range 3-7.
BLOCK TOO BIG	PATCH	An incorrect block number was typed. It cannot be longer than the length of the file being patched.
BLOCK NUMBER ERROR PHASE n	TDMARK	Refer to DEC-8E-EUZC-D.
C-FIELD NOT PERMITTED	BUILD	C-type fields cannot be specified in the INPUT section.
CANT BACKSPACE PAST BEGIN OR FILE	DAFT	Attempt was made to backspace past beginning of file.
CANT BACKSPACE WITH SEQUENTIAL INPUT	DAFT	Attempted to backspace with sequential input.
CCP ERROR	COMP	Matching angled bracket (< or >) missing.
CHECKSUM ERROR PHASE n	TDMARK	Refer to DEC-8E-EUZC-D.
COMMA MISSING	COMP	No comma appeared where one was expected.
CONTROL BUSY ERROR	RK8MRK	Disk IOT issued when control was busy.
DATA INITIALIZATION MISSING	COMP	No data initialization followed a comma in a data specification statement.
DATA ERROR PHASE n	TDMARK	Refer to DEC-8E-EUZC-D.
DATA RATE ERROR	RK8MRK	The processor was busy and did not respond to a data break request within the 13 us required. The transfer

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
		is terminated immediately.
DATA TABLE OVERFLOW!	BUILD,UPDATE,SORT	Too much data. Maximum = 960 characters.
DESCRIPTOR TABLE OVERFLOW!	BUILD,UPDATE,SORT	Too many F, T and C fields defined. Maximum = 160.
DESTINATION FIELD NOT DECIMAL	BUILD	The destination field specified is not defined as decimal.
DIBOL FILE NUMBER IN USE AT LINE nnnn	Run Time	In INIT, the channel number is already INITed to a mass storage device.
DIBOL FILE NUMBER NOT INITED AT LINE nnnn	Run Time	An attempt was made to XMIT, READ, or WRITE with a channel number that was not INITed.
DISK DATA ERROR	RKEMRK	Refer to DEC-08-DHRKD-A-D.
DT n?	Monitor	The expected tape is not available on a DECTape drive.  For example, if following a MOUNT message the operator has specified that a file should be written on unit 3, and unit 3 is not selected, or is WRITE LOCKed, this message is output.  On LINctapes there is no message.  To recover, set switch to ENABLE the unit or set the dial selector to the proper number. Then type any key to continue.
END OF FILE	Run Time	The last record of an

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
		input file has been read and the end of file mark encountered but no EOF label was specified in the XMIT statement.
END OF INPUT FILE AT RECORD nnnn	DAFT	Attempted to read past end of file mark on input. Not necessarily an error.
END TAPE ERROR PHASE n	DTMARK	Refer to DEC-08-EUFB-D.
ENTER "DATE MM/DD/YY"	Monitor	Typed an unrecognizable date.
ERROR IN COMMAND	Monitor	Miscellaneous.
EXEC WORD ERROR	RK8MRK	An error occurred while reading the EXEC. word of the sector.
EXCESSIVE GRID SIZE	DAFT	The grid printer width may not be greater than 130 characters.
EXPECTED LABEL IS MISSING	COMP	A required label is missing.
EXPRESSION NOT ALLOWED	COMP	An expression or bad character occurs to the left of an =.
EXTRA CHARS AT END OF DATA	BUILD	More data was entered than was defined in the control program.
EXTRA CHARS AT STMT END	BUILD, SORT, UPDATE COMP	Characters not relating to statement appear on line.
FIELD NOT DECIMAL	BUILD	Field not defined as decimal.
FIELD NUMBER MISSING OR 0	BUILD, SORT, UPDATE	Field number or default unit number is missing or is 0 or greater than 511.
FIELD OUT OF RANGE	BUILD	Data entered is not within the range

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
		specified in the control program.
FIELD TOO LARGE OR 0	COMP	In a data description statement, the dimension was 0 or more than 3 digits long, or the field size was 0 or larger than 511.
FILE NOT FOUND	Monitor,PATCH	The program with the name specified was not found. After, for example, FETCH FILEX.
FORMAT ERROR	RK8MRK	An error occurred while reading the header word.
FORMAT PARITY ERROR	RK8MRK	A parity error occurred in either the Header word or Exec word while reading the sector.
FULL	CONVEX	Specified an output device with -S option and source file being created was filled.
HASH FIELD MISSING	BUILD	No field specified for storage of hash total.
I/O ERROR ON xx, RETRY? (COS-310 only)	Monitor	System failed in three attempts to read created was filled.
I/O ERROR ON xxn, RETRY? (COS-310 only)	Monitor	Where xx is the device type and n is the unit number. Type N for no retry. Any other input causes the device handler to retry.
I RECORD ALREADY EXISTS	UPDATE	Tried to insert a record already present.
ILLEGAL CHAIN AT LINE 0000		CHAIN argument does not match .RUN command.
ILLEGAL DEVICE	SYSGEN	Device other than DT0-DT7 or DK0-DK3.

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
ILLEGAL DEVICE AT LINE nnnn	Run Time	Attempt to WRITE on a file that was not INITIALIZED for UPDATE or attempt to READ from a file that was not INITIALIZED or INITIALIZED for INPUT or UPDATE.
ILLEGAL DEVICE SWITCH	PIP	A switch was specified that was not /R, /C, or /K for input or /P, /L or /T for output.
ILLEGAL OPERATOR	COMP	A bad character was encountered in an expression where an operator would be expected.
ILLEGAL RECORD # AT LINE nnnn	Run Time	Record number is 0, past the end of the physical unit or the length specified in the record header word does not match the length of the XMIT block (records in data file are all not the same length).
ILLEGAL SORT KEY	SORT	Bad syntax on KEY statement, KEY too complex or KEY statement missing.
ILLEGAL STMT	COMP	The statement was not a data manipulation statement with a recognizable keyword.
ILLEGAL SUBSTRING AT LINE nnnn	Run Time	A DIBOL PROCedure section statement has attempted to access a data field, FI(m,n), but m=0 or m>n.
?ILLEGAL UNIT	Monitor	The unit specified is not DK0-DK3 or DT0-DT7; for example,  WRITE FILE,XY3 WRITE FILE, RK9



<u>Message</u>	<u>Program</u>	<u>Explanation</u>
ILLEGAL UNIT	SORT	Default unit is 0 or >15.
IN USE	Monitor	The unit specified in answer to the MOUNT message is already being used.
INCREMENT TOO BIG	BUILD	Length of increment field larger than length of output field.
INSUFFICIENT SPACE ON DEVICE	SYSGEN	Attempt to allocate more segments than are available on a device.
INITIAL ALPHA VALUE DOESN'T BEGIN WITH QUOTE	BUILD,SORT,UPDATE	Beginning quotation mark missing for initial alpha value.
INITIAL VALUE TOO BIG	BUILD,SORT,UPDATE	The initial value in a data specification statement had a length different from the field size specified.
INITIAL VALUE TOO SMALL	BUILD,SORT,UPDATE	The initial value specified is smaller than the field size.
KEYWORD MISSING OR TOO BIG	BUILD	
LABEL NOT ALLOWED	COMP	A symbol in an arithmetic expression was not of type alpha or decimal, or a symbol which had been redefined was used.
LAST INT NOT END ZONE	DTMARK	Refer to DEC-08-EUFB-D.
LINE # TOO LARGE	Monitor	Greater than 4095.
LINE TOO LONG	Monitor	Greater than 120 characters.
LINE TOO LONG AT LINE nnnn	Run Time	An input line (record) overflowed the block into which it was read.

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
LOCATION TOO BIG	PATCH	A location $\geq 400$ was typed.
MARK TRACK ERROR PHASE n	TDMARK, DTMARK	The DECTape being used is bad. Try reformatting the tape.
!*!MEMORY FAIL!*!	Monitor	A hardware error has occurred. Restart the program.
MISSING CLOSE PAREN	COMP	No close parenthesis occurred where one was expected.
MISSING CLOSE QUOTE ON ALPHA INITIAL	SORT,UPDATE,BUILD	Quotation mark misplaced or missing.
MISSING FIELD NAME	BUILD	Field name is missing.
MISSING INITIAL VALUE	BUILD,SORT,UPDATE	Comma was inserted after type and size but initial value was not specified.
MISSING OPEN PAREN	COMP	No open parenthesis occurred where one was expected.
MISSING OPERAND	COMP	A binary operator occurs in an expression with no operand following it; or no expression at all occurs where one is expected.
MISSING OR BAD DEVICE	COMP	The device in an INIT statement was missing or started with an illegal character.
MISSING OR BAD OPTION	COMP	In a record statement, no option appeared after a comma or the option which appeared was not X or C.
MISSING OR OUT OF ORDER	UPDATE	On C or D, record was not there. Probably because scratch file was

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
		not sorted.
MISSING QUOTE	COMP	The statement contained an odd number of quotes (').
MISSING RELATIONAL	COMP	No relational appeared in an IF statement.
MISSING UNIT	UPDATE	Unit number specified is a multiple of 16.
NAME PREVIOUSLY DEFINED	COMP	The name used was previously defined and this statement tries to redefine it.
NECESSARY FIELD MISSING	BUILD	Field specified as necessary in control program was not entered.
NO		BOOT Device switch illegal or missing.
NO BUFFERS LEFT AT LINE nnnn	Run Time	Not enough core available for I/O buffers. An I/O buffer of some multiple of 512 characters is set up for each active mass storage file. Another possibility: too few files were specified in the PROC statement.
NO CHANGE IN BLOCK	PATCH	An attempt was made to write a block but no changes were made to it. If this was on purpose, fine; otherwise, make changes to the block again.
NO COMMA AFTER FIELD NAME	BUILD, SORT, UPDATE	No comma or a character other than comma was specified after the field name.
NO DEFINE!	UPDATE	Control file did not start with a DEFINE. Returns to Monitor immediately.

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
NO END	CONVEX	Entire OS/8 ASCII input file was read but no end of file (CTRL/Z) was found.
NO END STATEMENT	UPDATE	END statement missing after OUTPUT specification.
NO FILE AT LINE nnnn	Run Time	No file specified in RUN statement to satisfy INIT (SYS) command.
?NO FILE TO SAVE	Monitor	Nothing in the edit work area when WRITE command is issued.
NO INIT DONE	Monitor	Program attempted a read or write on a device not opened by the system program.
NO INPUT	SORT, MERGE	Input file is null or not enough input files specified for a MERGE.
NO ROOM	PIP, CONVEX	Attempt to store a file on a full device.
NONCE ERROR	DAFT	Tried to use a feature not currently supported or documented.
NONEXISTENT FIELD	UPDATE	Tried to update Fn where no Fn appeared in file description section.
NOT A OR D	BUILD,COMP, SORT, UPDATE	A character other than A or D occurred in a data specification statement where A or D was expected.
NOT BUILD FILE!	BUILD	File did not start with DEFINE statements.
NOT ENOUGH ROOM FOR SYSTEM AND FILES	SYSGEN	Device specified was too small to accommodate system program and files.

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
NOT FOUND	CONVEX	File with specified name not found.
NOT I,D, OR C	UPDATE	Bad first character on update command.
NOT LABEL	COMP	A symbol which was not a 'label' occurred where a label was required.
NOTHING AFTER FIELD NAME	BUILD, SORT, UPDATE	Field type and size are not specified after field name and comma.
NUMBER REPEATED OR OUT OF ORDER	BUILD, SORT, UPDATE	A field sequence number is used more than once or is out of ascending order sequence.
# TOO LARGE	SYSGEN	Number entered was greater than 4095,
NUMBER TOO LONG	Run Time	A decimal field longer than 15 digits was used in a calculation.
OUT OF ORDER	UPDATE	SORT statement missing and update commands are not in order.
OUTPUT OF ROOM!	BUILD	Too many descriptors in INPUT section. Maximum keywords = approx. 121. Aprox. maximum fields for all keywords = 192. Too many fields in OUTPUT section. Aprox. maximum = 128.
OUTPUT ERROR	SORT	Indicates a system malfunction.
OUTPUT RECORD TOO BIG	BUILD	Record is larger than 510 characters.
PARITY ERROR	RK8MRK	A bit in data, parity or timing has been picked up or dropped on read. Transfer will continue to the end of the sector where the error occurred. Word count,

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
		current address information can be used to identify the error.
PARTIY ERROR PHASE n	DTMARK	Refer to DEC-08-EUFB-D.
PIP INITIALIZATION ERROR	PIP	PIP unble to identify the type of devices resident in the system. Following this message, PIP attempts to return control to Monitor.
PROGRAM TOO BIG	COMP	Binary output too big for the binary scratch area. Remedy: Run PIP/n with Option E to enlarge scratch area.
PROGRAM TOO BIG AT LINE nnnn	Run Time	Binary program does not fit in available core. Reduce program size.
PUSHDOWN OVERFLOW AT LINE nnnn	Run Time	Either (1) a statement is too complex or (2) subroutines are nested to a depth greater than 50, or a combination of the two.
READ STATUS ERROR	RKEMRK	Refer to DEC-08-DHRKD-A-D.
RECALIBRATE STATUS ERROR	RKEMRK	Refer to DEC-08-DHRKD-A-D.
RECORD TOO BIG	COMP	A named record exceeded 510 words in size.
REPLACE?	Monitor	Duplicate file names. Type Y to replace.
RETURN WITHOUT CALL AT LINE nnnn	Run Time	The program tried to execute a RETURN, but there was no place to go; there was no corresponding CALL statement.
SECTOR NO GOOD	RK8MRK	The program attempted to read or write data on a sector whose header

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
		words indicated a bad sector. The transfer is terminated immediately.
SELECT ERROR PHASE n	DTMARK	
SETUP?	TDMARK, DTMARK	Indicates an error in the DECTape setup. One of the units specified is in WRITE LOCK position, not selected, or the write flip-flop is unable to be set, or there may be a timing error.
STMNT TOO COMPLEX	COMP	The statement generated too much code and overflowed the Compiler's code buffer, or it had too much nesting and overflowed the Compiler's pushdown stack. Remedy: break up the statement into smaller parts.
SUBSCRIPT ERROR	COMP	No comma or close parenthesis occurs after a subscript.
SUBSCRIPT NOT DECIMAL	COMP	The type of a subscript was not decimal.
SUBSCRIPT TOO BIG AT LINE nnnn	Run Time	Program attempted to destroy the run-time system or itself by using a large subscript; larger than that defined in the Data Definition section. Note that the run-time system does not detect all illegal subscripts; only those which would cause the user's program or the system to be destroyed.
SWITCH NOT SET TO WIM OR SINGLE LINE FLAG FAILED TO SET SWITCH TO WIM.	TDMARK	Switch on the M868 module is not set to the WIM position or the

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
		timing generator for writing the mark and timing tracks is not setting the single line flag.
SYMBOLS DEFINED BUT NOT REFERENCED; xx	CREF	Symbols were defined but not referenced. Not necessarily an error.
SYNTAX ERROR	SYSGEN	Missing comma, extra characters, etc.
TXT AREA FULL	Monitor	Greater than 8,150 characters.
TIME OUT ERROR	RK8MRK	The control did not complete an operation after 32 revolutions.
TIMING ERROR PHASE n	TDMARK	Refer to DEC-8E-EUZC-D DTMARK Refer to DEC-08-EUFB-D.
TOO BIG	BUILD	Data entered is greater than the size of the field as defined in the control program.
TOO MANY FILES	MERGE	More than 6 input files specified.
TOO MANY ITEMS	COMP	More items appeared in a data initialization than were specified by the dimension.
TOO MANY SYMBOLS!	COMP	A fatal error message 365 symbols allowed in symbol table in 8K system, and 511 symbols allowed in larger systems. The compiler stops compiling; no storage map can be produced.
TOO MUCH DATA	COMP	Program's data division exceeds 32K bytes.
TRACK ADDRESS ERROR	RK8MRK	Track, surface or sector



<u>Message</u>	<u>Program</u>	<u>Explanation</u>
		address read from the disk did not agree with the address count registers or the disk drive electronics indicated track position 000 and the track counter did not agree. The transfer is terminated immediately.
TRACK CAPACITY EXCEED ERROR	RK8MRK	The program attempted to read or write beyond sector 17.
UNDEFINED FIELD	BUILD	Field referenced was not defined in field descriptor section of control program or a decimal field was being equated to an alpha field, or an alpha field was specified for a flag.
UNDEFINED FORMAT	BUILD	Format number referenced but not defined (errors given on END statement).
UNDEFINED NAME	COMP	A name is used which was never defined in the data section.
UNEXPECTED END OF FILE	BUILD, UPDATE	Missing END statement.
UNIT xx IS FREE	SORT	NOT AN ERROR - merely an informative message to aid the operator. xx is a COS unit number.
UNRECOGNIZABLE LINE	SORT	Parameter line did not start with a good keyword.
WRITE ERROR PHASE n	TDMARK	Refer to DEC-8E-EUZC-D.
WRITE LOCK ERROR	RK8MRK	The program attempted to write a section that was write protected. The write operation is terminated immediately.
WRITE STATUS ERROR	RKEMRK	Refer to

<u>Message</u>	<u>Program</u>	<u>Explanation</u>
		DEC-08-DHRKD-A-D.
WRONG DATA TYPE	COMP	Mixed modes occurred in an expression; or an argument which was supposed to be decimal was not or one of the three arguments in a data manipulation statement was of the wrong type.
ZERO DIVISOR AT LINE nnnn	Run Time	The program attempted to divide by zero.
xxxx IS BEING IGNORED	CREF	The line specified cannot be interpreted.
xxxx SHOULD BE yyyy BLK ERROR PHASE n	DTMARK	Refer to DEC-08-EUFB-D.
xxxx SHOULD B yyyy DATA ERROR PHASE n	DTMARK	Refer to DEC-08-EUFB-D.

APPENDIX E  
OS/8 CHARACTER CODES

TABLE E-1  
OS/8 Character Codes

Character	8-Bit Octal	6-Bit Octal	Decimal Equivalent (Al Format)	Character	8-Bit Octal	6-Bit Octal	Decimal Equivalent (Al Format)
A	301	01	96	!	241	41	-1952
B	302	02	160	"	242	42	-1888
C	303	03	224	#	243	43	-1824
D	304	04	288	\$	244	44	-1760
E	305	05	352	%	245	45	-1696
F	306	06	416	&	246	46	-1632
G	307	07	480	'	247	47	-1568
H	310	10	544	(	250	50	-1504
I	311	11	608	)	251	51	-1440
J	312	12	672	*	252	52	-1376
K	313	13	736	+	253	53	-1312
L	314	14	800	,	254	54	-1248
M	315	15	864	-	255	55	-1184
N	316	16	928	.	256	56	-1120
O	317	17	992	/	257	57	-1056
P	320	20	1056	:	272	72	-352
Q	321	21	1120	;	273	73	-288
R	322	22	1184	<	274	74	-224
S	323	23	1248	=	275	75	-160
T	324	24	1312	>	276	76	-96
U	325	25	1376	?	277	77	-32
V	326	26	1440	@	300		32
W	327	27	1504	[	333	33	1760
X	330	30	1568	\	334	34	1824
Y	331	31	1632	]	335	35	1888
Z	332	32	1696	^(^)**	336	36	1952
0	260	60	-992	<-(-)**	337	37	2016
1	261	61	-928	Leader/Trailer	200		
2	262	62	-864	LINE FEED	212		
3	263	63	-800	Carriage RETURN	215		
4	264	64	-736	SPACE	240	40	-2016
5	265	65	-672	RUBOUT	377		
6	266	66	-608	Blank	0000		
7	267	67	-544	BELL	207		
8	270	70	-480	TAB	211		
9	271	71	-416	FORM	214		

\*\* The character in parentheses is printed on some Teletypes.



APPENDIX F  
COS CHARACTER CODES

TABLE F-1  
COS Character Codes

Character	Equivalent Negative Numbers	COS Code	Character	COS Code
Ø		21	!	Ø2
1		22	"	Ø3
3		24	#	Ø4
4		25	%	Ø6
5		26	&	Ø7
6		27	'	1Ø
7		3Ø	(	11
8		31	)	12
9		32	*	13
A		42	+	14
B		43	,	15
C		44	-	16
D		45	.	17
E		46	/	2Ø
F		47	:	33
G		5Ø	;	34
H		51	<	35
I		52	=	36
J		53	>	37
K		54		
L		55	?	4Ø
M		56	@	41
N		57	[	74
O		6Ø	\	--
P	-Ø	61		76
Q	-1	62	^	77
R	-2	63		--
S	-3	64	Leader/Trailer	--
T	-4	65	LINE FEED	--
U	-5	66	Carriage Return	--
V	-6	67	Space	Ø1
W	-7	7Ø	RUB OUT	--
X	-8	71	Blank	--
Y	-9	72	Bell	--
Z		73	TAB	75
null		ØØ	FORM	--



APPENDIX G  
DEMONSTRATION RUNS

The following write-ups are explanations of the subsequent ETOS examples illustrated on the terminal.

Figure G-1

1. The user logs into console 2.
2. BASIC is called in and the disk area directory is listed.
3. The directory of the public library (SYS) is listed.
4. The user attempts to call FILE1.BA from his disk area, but it is not saved there. It must be called from SYS.
5. File FILE1.BA is saved in the disk area.
6. The directory of the disk area shows that FILE1.BA has been saved.
7. The user tries to list FILE1.BA on the line printer, which has not been assigned to his job.
8. The line printer is owned by another job.
9. After JOB 3 deassigns the line printer, the user assigns it to his job, decrements the program counter by 1 and continues, so that the listing is printed.

Figure G-2

1. The user deposits IAC;TLS;JMP .-2 starting at location 200.
2. He then restarts his virtual machine at location 200.
3. This prints a string of ASCII characters until the virtual machine is halted. Note that TSF, JMP .-1 is not necessary since this is done automatically in the software buffering routines.

Figure G-3 (see section 7.2.1)

1. Starting at location 7777, the CHANIO IOT is deposited followed by the high order sector, low order sector, minus word count, and buffer address.
2. The MQ contains function 0 (read, wait) and channel 0.
3. The AC contains the address of the argument list-1.
4. The VM is started at 7777.

This sequence performs the bootstrap of channel 0 that the BOOT command in SCALE would execute.

-----  
 Figure G-1  
Demonstration Run #1

1. !LOGIN;10,10 ACCT  
 JOB 5 LOGGED IN ON CONSOLE 2  
 WELCOME TO QUODATA'S EXTENDED TIMESHARED OPERATING SYSTEM
2. .R BASIC  
 READY  
  
CAT  
 1/15/79 EDUCOMP BASIC  
 TEST1.BA 7 12/15/78  
 READY
3. CAT SYS:  
 1/15/79 EDUCOMP BASIC  
 FILE1.BA 5 12/10/78  
 READY
4. OLD FILE1.BA  
 FILE1.BA NOT FOUND  
 READY  
  
OLD SYS:FILE1.BA  
 READY
5. SAVE



READY

6. CAT  
1/15/79 EDUCOMP BASIC  
TEST1.BA 7 12/14/78  
FILE1.BA 5 1/15/79  
READY
7. LIST LPT:  
?ILEGAL IOT  
PC 7330 AC 177 FL 5100 MQ 30 SC 1146 SR 0
8. !ASSIGN LPT:  
DEVICE OWNED BY JOB 3 CONSOLE 1
9. !ASSIGN LPT:  
!PC 7327  
!CONT  
  
READY  
^C

Figure G-2  
Demonstration Run #2

1. .^VS  
DEP 200 7001 6046 5200
2. !CONT 200
3. !"#\$%&'()\*+,-./0123456789:;<=>?@ABCDEF^VS  
!

Figure G-3  
Demonstration Run #3

1. !DEP 7777 6100 0 0 7400 0
  2. !MQ 0
  3. !AC 7777
  4. !PC 7777  
!CONT
- .



## APPENDIX H

### GLOSSARY

Account	Entry in the master file directory, consisting of an account number, password and protection code. This entry controls access to ETOS.
Account Number	Identification number made up of two octal numbers (project number and programmer number), which uniquely define a user's storage area.
Accumulator (AC)	A 12-bit register in which arithmetic and logic operations are performed; also an input/output register. Information transfers between core memory and a slow input/output device pass through the AC.
Bit	An abbreviation of binary digit; a unit of data; a unit of data in binary notation.
Bootstrap	A technique for loading the first few instructions of a routine into storage, then using these instructions to bring in the rest of the routine. This usually involves either the entering of a few instructions manually or the use of a special switch on the computer console. The SCALE command BOOT bootstraps an ETOS virtual machine.
Break character	A character which activates the user's virtual machine.
Break mask	A 12-bit word which defines the characters which are break characters.
Buffer	An area used to temporarily hold information being transmitted between external and internal core storage

devices or input/output devices and internal high-speed storage. A buffer is often a special register or registers, or a designated area of internal storage.

Central processing unit

The hardware unit of a computing system that contains the circuits that control and perform the execution of instructions.

Channel

A path along which data can be sent; e.g., data channel, output channel. Each job under ETOS has eight channels associated with it, which are used to access ETOS files.

Compute-bound

A program using all the computational resources of the central processor. The program is running and not waiting for I/O.

Console

A section of the computer which may be used to control the machine, correct errors, determine the status of registers and counters, determine the contents of storage, revise the contents of storage, and set jump and stop switches. Also called the front panel or operator's console. Under ETOS, each terminal acts as a virtual console. See also "Dummy console".

Core memory or Core

A storage device utilizing matrix arrays of ferrite cores which are most often used as the computer's working memory.

Core resident

Used to describe a part of the monitor which resides in core memory at all times.

Device

A mechanical contrivance or appliance, such as the PT8E paper tape reader/punch, RK05 disk, DK8E-A clock, etc.

Device handler	A subroutine for controlling input/output on channels and peripheral equipment.
Directory	An area containing the names and location of user files.
DSK	Each virtual OS/8 has its own DSK.
Dummy console	A console controlling a job that does not have a physical terminal associated with it. The job can be accessed via SCALE's ATTACH, ECROF, FORCE and TALK commands.
ETOS	Extended Timeshared Operating System.
Executive	See Operating system.
Extended Arithmetic Element (EAE)	An extended hardware option which performs addition, subtraction, multiplication, and division.
External storage	The storage of information on a device which is not an integral part of a computer, but is a form prescribed for use by the computer.
Field	A 4096-word block. PDP-8 memory is divided into fields numbered from 0 to 7 depending on the amount of memory installed.
Flag word	A 12-bit word which is the processor flag word containing the status of the virtual machine.
Internal storage	The storage of information on a device which is an integral part of a computer. Most internal storage devices are magnetic cores or magnetic drums.

Interrupt	A signal which when activated causes the hardware to transfer program control to some specific location in internal storage thus breaking the normal flow of the program being executed. After the interrupt has been processed, program control is again returned to the interrupted program.
I/O wait	The state of a user program while it is waiting for input/output to be completed before resuming execution of the program.
Job	A virtual machine from the monitor's standpoint. Allowed to make requests on the system for computation, input/output, and memory. Under ETOS, jobs are assigned sequentially.
Keyboard echo	The ETOS monitor echoes the characters entered on the keyboard.
Keyboard monitor	Provides communication between the user and the OS/8 executive routines by accepting commands from the console terminal. The commands enable the user to create logical names for devices, run system and user programs, save programs, and call ODT.
Location	An address in memory. On the PDP8, each location is a twelve-bit word.
Login	Assign and initialize a virtual machine and bootstrap OS/8.
Logout	Release the virtual machine from the system.
Mapping	Conversion of a virtual request to a physical request. All I/O is mapped under ETOS.

Multiplier Quotient (MQ)	A 12-bit bi-directional shift register that acts as an extension of the AC during EAE operations.
Normal mode	When the terminal is communicating with the user program, as opposed to being in SCALE mode.
Operating system	An integrated collection of programs that automatically permits continuous job processing by a digital computer. It is often called a monitor, supervisor, or executive.
Peripherals	Various units or machines that are used in combination or conjunction with the computer, but are not part of the computer itself, such as line printer, typewriters, mass storage devices, etc.
Physical	Pertaining to the actual machine. This term contrasts the term "virtual".
Program Counter (PC)	A 12-bit register that is used to control the program sequence. The PC contains the address of the core memory location from which the next instruction is taken.
Programmer Number	Two digit octal number, used to further segment a project number.
Project Number	Two digit octal number, used to divide accounts into logical groupings, such as departments.
Quantum	The smallest unit of central processor time available to a user. For ETOS this is 100 milliseconds.
Register	A device for the temporary storage of one or more words to facilitate arithmetical, logical, or transferral operations. A

term to designate a specific computer unit for storing a group of bits or characters.

Resident monitor	That part of the ETOS monitor which is always located in core memory and is core resident.
Run queue	A list, line or queue of jobs ready or submitted for processing, but which has not yet begun to run.
SCALE	An acronym for System Command and Login/Logout Executive.
SCALE mode	Keyboard input is interpreted by SCALE as system commands.
Stand-alone	Pertaining to a single user operating system. This term contrasts the term time-sharing.
Step Counter (SC)	A 5-bit register which is part of the EAE option. Used to record the number of steps performed and to stop many EAE instructions after the correct number of operations.
Swap Region	The region on the system high-speed mass storage device where virtual memory fields reside when they are not in physical core memory.
Swapping	Transferring copies of user core memory between main memory and some high-speed mass storage device. Without swapping, all user jobs would have been resident in main memory at once.
Switch Register (SR)	The 12 switches which provide a means of communication between operator and machine. ETOS simulates the switches with a 12-bit SR register.



SYS	Read-only storage device containing all OS/8 public files and programs.
System	An integrated structure of hardware, software, data structures, and documentation which implements a certain function or set of functions. For example, ETOS, OS/8 and COS are all systems.
System device	A device which stores the user accounts and system files.
Terminal	An input/output device designed to receive data in an environment associated with the job to be performed and capable of transmitting entries to and obtaining output from the system of which it is a part.
Time-sharing system	A computer system in which the available computer time is shared among several users.
User mode	Hardware mode which disables I/O instructions, HLT and OSR from having their usual effect of stopping the physical machine. This protects user programs from each other.
Virtual	Generally pertains to an entity that ETOS simulates for the user, so that many jobs may share the corresponding physical resource.
Virtual Core, or Virtual Memory	Simulated core memory used by the virtual machine. The use of swapping makes it possible to have much more virtual memory on the system than physical memory. Each virtual memory field must have a corresponding swap region.
Virtual machine (VM)	The PDP-8 simulated by ETOS.

Word

A set of characters that occupies one storage location (12 bits for the PDP-8) and is treated by the computer circuits as a unit and transported as such.

## INDEX

Account Numbers	2-1	LOGIN	3-2
AC	3-10	LOGOUT	3-2
ASBLDR Program, OS/8	4-6	QUEUE	3-20
ASSIGN, SCALE Command	3-18	STOP	3-3
ASSIGN, OS/8 Command	4-15	SYSTAT	3-23
ATTACH SCALE Command	3-21	TALK	3-17
		VERSION	3-21
BASIC	4-6	Console number	2-3
EDUCOMP (EB BASIC)	4-6	CONTINUE	3-3
CBASIC	4-6	CORE	3-6
OBASIC	4-6	COS	5-1 to 18
BATCH, OS/8 Program	4-7	Bootstraps	2-6, 3-8, 5-5
BITMAP OS/8 Program	4-7	Character Codes	F-1
BOOTSTRAPS	3-7	Converting from OS/8 to	
COS	2-6, 3-8	COS	2-6, 3-8
OS/8	2-6, 3-7	Date Word	5-4
SCALE Command	3-7	Differences under ETOS	5-5
BREAK, COS Program	5-17	Error Messages	5-13
BREAK, SCALE Command	3-12	Monitor Changes	5-2
Buffering	3-28	System Programs	2-9, 5-5
BUILD, OS/8 Program	4-6	COSBO	2-6, 3-2
		CREF, OS/8	4-8
Card Reader, Assigning	3-19	CTRL/C (^C)	3-23
Card Reader, IOTs	6-7, 8, 9	CTRL/V (^V)	2-11
CBASIC Program	4-6	CTRL/P (^P)	3-23
CCL Program	4-8	CUSPs, OS/8	4-5 to 4-13
Channels, ETOS	7-1 to 15	ASBLDR	4-6
Channels, IOTs	7-15 to 34	BASIC	4-6
CLOSE SCALE Command	7-11	BATCH	4-7
COBOL Program	4-8	BITMAP	4-7
Commands, ETOS Channel	7-3 to 14	BUILD	4-3
CLOSE	7-13	CBASIC	4-6
DELETE	7-12	CCL	4-8
ENTER	7-5	COBOL	4-8
EXTEND	7-9	CREF	4-8
INQUIRE	7-14	DIRECT	4-8
LOOKUP	7-8	EDIT	4-9
READ	7-4	EPIC	4-9
REDUCE	7-11	FOCAL	4-10
RENAME	7-13	FORTRAN II	4-10
WRITE	7-5	FORTRAN IV	4-10
Commands, SCALE	3-1 to 3-28, A-1	FOTP	4-10
ASSIGN	3-18	FUTIL	4-10
ATTACH	3-2	HELP	4-11
BOOTSTRAP	3-7	KSORT	4-11
CONTINUE	3-3	OBASIC	4-6
CORE	3-6	PAL 8	4-11
CTRL/V	2-11	PIP	4-12
DEASSIGN	3-18	QBOL	4-12
DETACH	3-21	RESORC	4-12
EXAMINE	3-15	SET	4-12
HELLO	3-2	SQUASH	4-13

## INDEX

SRCCOM	4-13	Memory, IOTs	6-5
6SORT	4-13	Miscellaneous IOTs	6-25
TECO	4-13	Monitor	1-3, 4, 5
3040	4-13	MQ Command	3-10
DELETE Command	7-12	OBASIC	4-6
DEPOSIT Command	3-15	OS/8	
DETACH Command	3-21	Bootstrap	2-6, 3-7
DIRECT Program	4-8	Character Codes	E-1
EAE	3-16	Converting from OS/8 to ETOS	4-14
EB BASIC Program	4-6	Date Word	4-18
EDIT Program	4-9	Differences	4-1
ENTER Command	7-5	Error Messages	C-1
EPIC Program	4-9	Monitor Changes	4-2
ERROR MESSAGE, COS	D-1 to D-18	System Programs	2-8, 4-5 to 13
ERROR MESSAGES, OS/8	C-1 to C-44	PAL8 Program	4-11
ERROR MESSAGE, SCALE	3-25 to 27, B-1 to 4	Passwords	2-2
ETOS		PC Command	3-10
Architecture	1-3	PIP Program	4-12
Entering	2-2 to 5, 3-2	Private Packs	3-18
Exiting	2-12, 3-2	Programmer Number	2-1
EXAMINE Command	3-15	Project Number	2-1
Extended Arithmetic Element	3-10	QBOL Program	4-12
FLAGS Command	3-10	QLP Handler	4-23
FOCAL Program	4-10	QUEUE	3-20
FORTTRAN II Programs	4-10	R Command	3-8
FORTTRAN IV Programs	4-10	READ Command	7-4
FOTP Program	4-10	REDUCE Command	7-11
FUTIL Program	4-10	RENAME Command	7-13
HELLO Command	3-2	RESORC Program	4-12
HELP Program	4-11	SC Command	3-11
INQUIR Program, COS	4-23	SCALE Mode	2-1, 2-11, 3-1
INQUIRE	7-14	SET Program	4-12
Installation Name	4-18	SKETOS IOT	6-6
I/O Processing	1-3	Spooling	3-20, 4-23
KSORT, OS/8	4-11	SQUASH Program	4-13
KSTAT	3-13	SR Command	3-11
Line Printer, ASSIGNing	3-19	SRCCOM Program	4-13
Line Printer, IOTs	6-10	STOP Command	3-43
LOGIN Command	2-2, 3-2	Switch Register (SR)	3-11
LOGOUT Command	3-2	SYSGEN Program	5-6
Errors	2-7 to 8	SYSGET Program	5-8
LOOKUP Command	7-8	SYSTAT	3-23
Manual Conventions	1-5	System IOTs	6-11 to 24
		6SORT Program	4-13
		TALK Command	3-17
		TECO Program	4-13

## INDEX

Terminal Communication	3-17
Terminal IOTs	6-4
TIME Program	4-20
Timesharing IOTs	6-6
TTY Reader Control	3-28
3040 Program	4-13
UNBRK Program	5-17
Unimplemented IOTs	6-25
User Mode	1-2
VERSION	3-21
Virtual Machine	1-1
Virtual Memory	3-15
Virtual Registers	3-9
WHERE Command	3-15
WHO Program	4-22
WRITE Command	7-5



USER COMMENT FORM

MANUAL:

Please use the space below for any comments you may have concerning this manual. Our goal is to produce documentation that is clear, complete and useful, and you, the reader, are the final judge.

We are especially interested in any thoughts you may have concerning this manual's clarity, completeness, accuracy, structure and overall effectiveness.

Please use this form only for comments about this manual.

NAME:

---

OCCUPATION:

---

ADDRESS:

---

---

