

ETOS V5B: SYSTEM MANAGER'S GUIDE

LIST OF UPDATES AND REVISIONS

Please note that the physical number of sheets removed may not equal the number of sheets to be inserted as their replacement.

<u>Date</u>	<u>Remove Page(s)</u>	<u>Replacement Page(s)</u>	<u>Remarks</u>
12/1/79			New Version of Documentation
4/28/80	6-65 thru 6-72	6-65 thru 6-72	USAGE Corrections
9/30/80	2-37	2-37	Section Header Incorrect
	3-25	3-25 thru 3-28	New CONFIG Section
	8-23 thru 8-37	8-23 thru 8-38	CDUMP Correction
	E-9	E-9 thru E-12	Optional Patch Correction

3.17 SCALE ERROR MESSAGES

A complete list of SCALE error messages is contained in Appendix B. The section on each SCALE command contains the errors pertaining to that command. In addition to these specified error messages, the following is a list of general error messages which may occur after a number of SCALE commands.

?BAD NUMBER

Too many digits were entered, no number was entered where one was required, or non-octal digits were entered in an argument which must be octal.

?FILE SPECIFICATION ERROR

An illegal syntax for an ETOS file was specified. An invalid ETOS device, account number, file name, file extension or protection code causes this error.

?UNKNOWN COMMAND

A command is entered which is not one of the valid SCALE commands listed in Appendix A.

?WHAT? WHERE?

A command is utilized in a manner which is invalid.

In addition to the error messages printed in response to SCALE commands, there are several error messages which may be printed by SCALE while a user program is running. These messages stop the virtual machine and return it to SCALE.

?ILLEGAL IOT

An input/output instruction, which ETOS does not support, was executed by the user program. This message will be printed if a non-privileged user attempts to run a privileged program or if you attempt to access a device which was not ASSIGNED.

The PC contains the (address +1) of the instruction which generated the error. The FLAGS word indicates the field of the address in bits 6-8.

?JMP/JMS TO NON EXISTENT FIELD

The user program executed a CIF to a non-existent field and attempted to execute a JMP or JMS. The PC contains the (address +1) of the JMP/JMS and bits 6-8 of the FLAG word contains the Instruction Field which the program tried to jump to. There is no indication which field the JMP/JMS came from. If you wish 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.

?% VM HLT

The user program executed a HLT (7402) instruction. The PC contains the (address +1) of the instruction which contained the HLT. The FLAGS word indicates the field of the address in bits 6-8.

To recover from a ?% VM HLT, type one of the following:

- a) !R [0,2]OS8.RTS (if you were running in OS/8)
- b) !R COSDISK.RTS (if you were running in COS)

If either of these do not work, then LOGOUT and LOGIN again to

your account.

Do not use !CONT or !BOOT.

There are several errors which indicate a system malfunction. This malfunction may be due to hardware or software. If these errors occur frequently, there is a problem with the hardware, an ETOS system disk pack or the file structure. Diagnostics can be run to determine if there is a hardware problem. A copy of the system disk may be used to determine if the problem is caused by a disk pack. The file structure may be recreated by the system manager. A list of serious errors follows.

?READ ERROR

A READ error occurred while attempting to read information from an ETOS disk.

?SYSTEM SWAP ERROR

A READ or WRITE error occurred while swapping a user into or out of memory.

?SYSTEM R/W ERROR

A READ or WRITE error occurred while attempting to read or write information from the ETOS system pack.

?WAIT FOR I/O

Disk input/output has been initiated but not completed.

3.18 EFFECTS OF NO FREE BUFFERS ON SCALE

Under extreme circumstances, the monitor's character buffering space may be entirely used. In this case, attempts at output by SCALE will not succeed or only partially succeed. For instance, if the user typed ^VW, the entire output from the W command (see section 3.7.7) might not be printed if at the moment there was not enough space for it in the output buffers. The remainder is discarded. This action can be annoying in the case of a program which displays a textual message and then halts. If it just fills the output buffers, the message "?%VM HLT" is not printed, although the HLT will, of course, have its usual effect. In this case, the terminal might appear to be hung. Pressing <RET> determines the state of the terminal.

3.19 TTY READER CONTROL

If the XON/XOFF feature on the terminal is not enabled, reading paper tapes when the system is heavily loaded may result in the reader getting so far ahead of the program executing the read, that the input buffer overflows. It may be necessary to stop the reader about every 150 characters for a moment so that no characters are lost. This should occur only under severe loading.

If the user overflows the input buffer and his program does not read the buffer, no further characters may be entered. In this case, the only character that may be entered is ^V. Entering ^V completely clears the input buffer, and puts the TTY in SCALE mode for the rest of the current line (if the VM is running) or leaves its mode unaffected if it was already in SCALE mode. To escape a full input buffer, enter ^V<RET>.

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
TAD      (TODBUF-1   /MQ = FUNCTION NUMBER
CDB FLDCHG         /AC = BUFFER POINTER
SYSCAL           /BUFFER IS IN DATA FIELD
/ AC = 0 UPON RETURN /GET TIME OF DAY
.
.
.
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
TAD      (-COUNT   /MQ = FUNCTION NUMBER
SYSCAL           /AC = - TICKS (100 M.S.) TO SLEEP FOR
/ AC = 0 UPON RETURN /GO TO SLEEP

```

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	Program not resident
1	Non-resident job wait
2	Wake up on any condition
3	Waiting for DMON
4	Waiting on 1 minute timer
5	Waiting on 100ms timer
6	Card reader wait
7	Disk file input/output
8	Line printer wait
9	Terminal output wait
10	Terminal input wait
11	Waiting in SCALE mode

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
```

3.7	AUTOMATIC START-UP	3-12
3.8	MEMORY CONFIGURATION	3-12
3.9	TERMINAL CONFIGURATION	3-13
3.10	FILLER CHARACTERISTICS	3-16
3.11	IGNORING SPECIAL CHARACTERS	3-17
3.12	UNDEFINED INTERRUPTS	3-18
3.13	TUNING THE ETOS SCHEDULER	3-19
3.14	TERMINAL BUFFERS	3-21
3.15	TERMINAL OUTPUT	3-22
3.16	TERMINAL INPUT	3-23
3.17	AUTOMATIC COMMAND LINE	3-24
3.18	ERROR MESSAGES	3-26

START-UP OPTIONS

4.1	INTRODUCTION	4- 1
4.2	SYSGENING A PACK	4- 1
4.3	TRANSFERRING VIRTUAL OS/8	4- 4
4.3.1	Transferring Virtual OS/8 System Programs	4- 4
4.4	RETURNING TO STAND-ALONE OS/8	4- 6
4.5	STARTING ETOS FROM SINGLE-USER OS/8	4- 7
4.6	INITIALIZING THE SYSTEM TIME AND DATE	4- 8
4.7	STOPPING ETOS	4- 9

ETOS ACCOUNTS

5.1	INTRODUCTION	5- 1
5.2	ACCOUNT STRUCTURE	5- 2
5.2.1	Account Attributes	5- 3
5.2.2	ETOS File Attributes	5- 5
5.2.3	Initial Account List	5- 9
5.3	ACCNT PROGRAM	5- 9
5.3.1	Initiating The ACCNT Program	5-10
5.3.2	Available ACCNT Options	5-12
5.4	LISTING ACCOUNTS AND ETOS FILES	5-14
5.5	CREATING ACCOUNTS AND ETOS FILES	5-22
5.5.1	Creating an OS/8 Storage Area	5-28



5.5.2	Creating a COS Program Storage Area	5-31
5.5.3	Creating a COS Data File Storage Area	5-35
5.6	DELETING ACCOUNTS AND ETOS FILES	5-37
5.7	CHANGING ACCOUNT AND ETOS FILE ATTRIBUTES	5-39
5.8	CHANGING ETOS FILE LENGTHS	5-42
5.9	ADDITIONAL ACCNT INFORMATION	5-45
5.9.1	Input Conventions	5-45
5.9.2	Restrictions	5-48
5.9.3	ACCNT Error Messages	5-49
5.9.4	Enabling Other Consoles to Run ACCNT	5-53
5.9.5	Enabling Other Accounts to Run ACCNT	5-55
5.9.6	Specifying ACCNT Commands From an OS/8 Input File	5-56
5.9.7	Printing Output to an OS/8 Output File	5-58
5.10	ACCESSING A PRIVATE PACK	5-60
5.10.1	Accessing a Stand-alone OS/8 Pack	5-61
5.10.2	Accessing a Stand-alone COS Pack	5-64
5.10.3	Creating a Stand-alone OS/8 Pack	5-67
5.10.4	Creating a Stand-alone COS Pack	5-68
5.11	CREATING A PUBLIC PACK	5-69
5.11.1	Creating an OS/8 Storage Area	5-73
5.11.2	Creating a COS Program Storage Area	5-75
5.11.3	Creating a COS Data File Area	5-78
5.12	MOVING FILES BETWEEN VIRTUAL OS/8 AND ETOS	5-79
5.12.1	Loading SYSTAT	5-82
5.12.2	Loading DKCOPY	5-83
5.12.3	Loading LOGIN	5-84
5.12.4	Loading an Installation File	5-85
5.12.5	Loading a LOGIN Message	5-86
5.13	MOVING FILES BETWEEN ACCOUNTS	5-87
5.13.1	Moving Files Between OS/8 Storage Areas	5-90
5.13.2	Moving Programs Between COS Storage Areas	5-91
5.13.3	Moving Files Between COS Data File Areas	5-92
5.13.4	Moving Programs Between OS/8 and COS Storage Areas	5-93

5.13.5	Moving Files Between OS/8 Storage and COS Data File Areas	5-95
5.13.6	Moving Files Between Real and Virtual OS/8	5-96
5.13.7	Moving Files Between OS/8 Virtual SYS Storage and OS/8 Storage Areas	5-97
5.14	FACILITATING ETOS FILE ACCESS	5-98
5.14.1	Setting LOOKUP Defaults	5-101
5.14.2	Inhibiting LOOKUP Questions	5-102
5.15	ETOS SYSGEN ACCOUNTS	5-104
5.15.1	Master File Directory (Account [00,01])	5-104
5.15.2	Library (Account [00,02])	5-105
5.15.3	Operator's Account (Account [00,03])	5-109
5.16	DISTRIBUTION ACCOUNTS	5-112
5.16.1	Documentation (Account [00,04])	5-113
5.16.2	Differences (Account [00,05])	5-113
5.16.3	Device Handlers (Account [00,06])	5-114
5.16.4	ETOS Cusps (Account [00,07])	5-116
5.16.5	Assembly Language (Account [00,10])	5-117
5.16.6	Stand-Alone Software (Account [00,11])	5-119
PRIVILEGED OPERATIONS		
6.1	INTRODUCTION	6- 1
6.2	PRIVILEGED SCALE COMMANDS	6- 1
6.2.1	PRIVilege Command	6- 2
6.2.2	PEEK and POKE Commands	6- 4
6.2.3	BROADCAST Command	6- 5
6.2.4	FORCE and ECROF Commands	6- 6
6.2.5	Write Lock and Write Enable Commands	6- 8
6.2.6	MOUNT and DISMOUNT Commands	6- 8
6.2.7	SHUTUP Command	6-11
6.3	PRIVILEGED SYSCAL FUNCTIONS	6-11
6.3.1	LOCK	6-11
6.3.2	UNLOCK	6-12
6.3.3	HOOK	6-12
6.3.4	UNHOOK	6-13

6.3.5	MOUNT	6-14
6.3.6	XTRAP	6-15
6.3.7	DISMNT	6-15
6.4	PRIVILEGED OS/8 FUNCTIONS	6-16
6.4.1	Changing Device Handlers in OS/8	6-17
6.4.2	TD8E DEctape	6-26
6.4.3	TC08 DEctape	6-27
6.4.4	RX01 Floppy Disk	6-28
6.4.5	PT8E High Speed Reader/Punch	6-28
6.4.6	User-written Device Handlers	6-29
6.4.7	Reading Binary Paper Tapes	6-37
6.4.8	Enabling CCL	6-37
6.4.9	Setting Terminal Characteristics	6-38
6.4.10	Setting Line Printer Characteristics	6-43
6.4.11	Setting General Device Characteristics	6-44
6.4.12	Setting Automatic LOGIN Batches	6-47
6.4.13	Line Printer and Terminal Spooling	6-55
6.4.14	Printing Cumulative Job Statistics	6-64
6.5	PRIVILEGED COS FUNCTIONS	6-73
6.5.1	RX01 Floppy Disk	6-73
6.5.2	VT05 and VT52 Scopes	6-74
6.6	ON-LINE BACKUP	6-75
6.7	TEMPORARY PRIVILEGE	6-82
ETOS REAL-TIME PROGRAMMING		
7.1	INTRODUCTION	7- 1
7.2	SAMPLE PROGRAM	7- 3
7.3	SYSTEM STRUCTURE	7- 6
7.4	SCHEDULING	7- 6
7.4.1	Levels	7- 6
7.4.2	Scheduling Algorithms	7- 7
7.5	MONITOR AND MEMORY ARCHITECTURE	7- 9
7.6	REAL-TIME TASKS	7-11
7.7	PRIVILEGED LEVEL 3 PROGRAMMING	7-12
7.8	REAL-TIME DEMONSTRATION	7-13

ERROR DIAGNOSIS	
8.1 INTRODUCTION	8- 1
8.2 ADDITION AND REMOVAL OF TERMINALS	8- 2
8.3 RUNNING FROM NON-SYSTEM DRIVES	8- 5
8.4 DIAGNOSING DISK ERRORS	8- 7
8.4.1 RK05 Disk Errors	8- 7
8.4.2 System Industries Disk Errors	8- 8
8.4.3 Verifying Bad Disk Blocks	8-10
8.5 MFD AND UFD STRUCTURE	8-11
8.6 PRINTING FREE BLOCKS FILE	8-21
8.7 RECOVERY AFTER A CRASH	8-23
8.8 DEBUGGING THE MONITOR	8-28
8.9 EXAMINING PHYSICAL MEMORY	8-29
8.10 SOFTWARE PERFORMANCE REPORTS	8-30
8.11 TECHNICAL SUMMARIES	8-30
8.12 MEDIA MANAGEMENT	8-33
8.12.1 Entering SMRs	8-34
8.12.2 Maintaining the Production Pack	8-37

Figure 6-55	Sample Run of USAGE	6-65
Figure 6-56	Initialization of USAGE Statistics File	6-66
Figure 6-57	Sample Deletion of USAGE Statistics File	6-66
Figure 6-58	Sample Run of USAGE with Options	6-69
Figure 6-59	Sample Access of the RX01 Under COS	6-74
Figure 6-60	Disabling the Answer Back Feature Under COS	6-75
Figure 6-61	Sample Initiation of DKCOPY	6-77
Figure 6-62	Sample Run of DKCOPY	6-78
Figure 6-63	Setting of Temporary Privilege	6-82
Figure 7-1	Sample Real-Time Program	7- 3
Figure 7-2	Compiling and Saving RTDEMO	7-15
Figure 8-1	Sample Run of TTYSET	8- 3
Figure 8-2	Running ETOS from Non-system Drives	8- 5
Figure 8-3	Sample Run of ETOS from a Non-system Drive	8- 6
Figure 8-4	Sample SHUTUP of ETOS from a Non-system drive	8- 6
Figure 8-5	Sample Operational RK05 Disk Error	8- 8
Figure 8-6	Sample Run of FRRCPY	8- 8
Figure 8-7	Sample Run of the 3040 Program	8-10
Figure 8-8	Sample Verification of Bad Blocks	8-11
Figure 8-9	Sample Run of FREE	8-22
Figure 8-10	Sample Crash Dump Under Stand-alone OS/8	8-24
Figure 8-11	Sample Crash Dump Under ETOS	8-25
Figure 8-12	Sample Crash Dump Under ETOS	8-25
Figure 8-13	Sample Crash Dump Under ETOS	8-26
Figure 8-14	Creating a Debugging Version of ETOS	8-28
Figure 8-15	Sample Run of Debugging Version of ETOS	8-28
Figure 8-16	Output of FUTIL Dialogue to Line Printer	8-36
Figure 8-17	Sample FUTIL Input Batch	8-37
Figure E-1	Modifying LOGIN to set PRIVilege	E- 3
Figure E-2	Modifying Sysgen to set LOGIN PRIVilege	E- 4
Figure E-3	Disabling Cumulative Statistics Files	E- 6
Figure E-4	Modifying Sysgen to Disable Cumulative Statistics Files	E- 7
Figure E-5	Making the TALK Command PRIVileged	E- 8
Figure E-6	Disabling CHANIO Commands	E-10
Figure E-7	Changing the Comma Construction of Commercial BASIC	E-11
Figure E-8	Setting Line Printer Width in Commercial BASIC	E-12

2.10 ENABLING PROPER TERMINAL CHARACTERISTICS

In order to perform proper terminal service, terminals and terminal interfaces must have proper settings. These settings control parity, stop bits, filtering and filler characters.

2.10.1 Parity

When a character is struck on a terminal connected to a PDP8, a bit pattern is sent to the computer via an interface. This bit pattern is sent in ASCII coding. However, ASCII standards specify only the seven-bit configuration for all characters. The high order eighth bit may be 0 or 1, depending on the settings of the terminal or the interface. Some of the software for the PDP8 ignores this high order bit. The remainder of the software assumes that this high order bit is always set ("eighth-bit always marking"). If this bit is not set, the character is ignored.

Most DEC terminals (e.g., ASR33, LA36, VT52) have an option to always set the high order bit to 1. The VT100 and many non-DEC terminals do not have this option. The latter terminals usually have options for odd or even parity. With odd parity, each character is transmitted with an odd number of bits. The high order bit is used to insure this fact. An "A" is represented as a 301, "B" is represented as 302, "C" is represented as 103, etc. With even parity, each character is transmitted with an even number of bits. "A" is represented as 101, "B" is represented as 102, "C" is represented as 303, etc. Both of these types of parity are unacceptable for software which assumes the high order bit is always on.

For terminals which do not have the capability to set the high order bit, you must set this bit on the interface board. The KL8E (M8650) and DKC8-AA (M8316) do not have this feature.

Therefore, you should not connect a VT100 or Hazeltine terminal to one of these modules.

The KL8JA (M8655) and KL8A (M8319) boards contain this option. To enable the option on a KL8JA board, insure that the

NP jumper,
NB1 jumper, and
NB2 jumper

are removed from the board. Refer to the KL8JA (M8655) prints for the location of these jumpers.

To enable the option on a KL8A board (M8319), insure that the

NP jumper,
NB1 jumper and
NB 2 jumper

are removed from the unit for each line on which you wish to generate eighth bit always mark. Refer to the KL8-A User's Manual for the location of these jumpers.

2.10.2 Stop Bits and Filter

In addition to possible parity problems, you must consider problems concerned with terminals running at 110 baud. Interfaces operating at 110 baud must generate two stop bits instead of one. A filter must also be enabled.

To enable a KL8-JA (M8655) to generate two stop bits and enable the 110 baud filter, insure that the

W4 jumper

is removed from the board and that the

*INITLN,offset=ASCII code

INITLN is a 30 (base 8) or 24 (base 10) word table. Each entry contains an ASCII code which represents one character in the command. The command is formed by appending each character starting with offset 00. The command must be terminated by a 215 (<RET>). After the last entry in the command, you must set the next entry to 0000. The initial values in the INITLN table are listed in Table 3-10.

Table 3-10
Initial Value of the INITLN Table

<u>Offset</u>	<u>Entry</u>	<u>Character Value</u>
00	0310	H
01	0273	;
02	0260	0
03	0254	,
04	0263	3
05	0240	space
06	0317	0
07	0320	P
10	0305	E
11	0322	R
12	0301	A
13	0324	T
14	0317	O
15	0322	R
16	0215	<RET>
17	0000	terminator
20	0000	
21	0000	
22	0000	
23	0000	
24	0000	
25	0000	
26	0000	
27	0000	

The initial command specification is H;0,3 OPERATOR<RET>. This command LOGs you INTO the operator's account. If you had changed the password of account [0,3] to SYSTEM, the dialogue of Figure

3-14 would be used to specify the INITLN.

Figure 3-14
Sample Use of the INITLN Option Under CONFIG

```
*INITLN,06=323
INITLN(00006) 0317 0323
*INITLN,07=331
INITLN(00007) 0320 0331
*INITLN,10=323
INITLN(00010) 0305 0323
*INITLN,11=324
INITLN(00011) 0322 0324
*INITLN,12=305
INITLN(00012) 0301 0305
*INITLN,13=315
INITLN(00013) 0324 0315
*INITLN,14=215
INITLN(00014) 0317 0215
*INITLN,15=0
INITLN(00015) 0322 0000
*
```

After you use the INITLN option to automatically LOG you IN, you can use the SET SYS INIT option of OS/8 to automatically execute a command or a batch file. See 6.4.12 for details on using SET SYS INIT.

3.18 ERROR MESSAGES

All error messages which may be output by the CONFIG program are listed below.

BAD CHARACTER IN OFFSET SPECIFICATION

One of the characters in the offset is not a number in the range 0 to 7.

CONFIG OPTION NOT FOUND

The specified option is not one of the valid options which are

listed in Table 3-2.

INCOMPATIBLE VERSION OF ETOS.SV

The version of ETOS.SV on the OS/8 system device is not V5B. You cannot configure a V5A system with V5B CONFIG.

INTERNAL FAILURE. CALL QUODATA (203) 728-6777

The file ETOS.SV is corrupted. After insuring that you have a valid copy of ETOS.SV on the OS/8 system device, call QUODATA for further instructions.

INVALID ADDRESS SPECIFICATION FOR MONITOR

The address specification is not a one to four digit octal number.

INVALID FIELD FOR MONITOR PATCH

The field specification for a monitor patch is not 0 or 1.

I/O ERROR ON SYS:ETOS.SV. RESTORE WITH BACKUP.

There is a bad block in the file ETOS.SV on the OS/8 system device. Copy a new file ETOS.SV from backup medium.

NO OPTION OTHER THAN /T ALLOWED FOR THIS REQUEST.

When specifying a binary patch, /O and /L options are not allowed. The /T option has no effect but it is allowed to retain compatibility with any batch streams created under ETOS V5A.

NO OPTIONS OTHER THAN /O AND /L ALLOWED FOR THIS REQUEST

When specifying a table or monitor component, /O and /L are the only options allowed. All other options are invalid.

NO SYSG,1

Since SYSGEN is a 4K word module, field 1 does not exist for this component. Only field 0 is allowed.

SYS:ETOS.SV NOT FOUND

The file ETOS.SV is not found on the OS/8 system device. You must copy ETOS.SV onto the current SYS or ASSIGN SYS to a device containing ETOS.SV.

TABLE OFFSET OUT OF RANGE

An offset has been specified which is greater than the maximum number of entries in the table. The length of each table is documented in the section pertaining to the table.

TOO MANY DIGITS IN THE = SPECIFICATION

More than four digits have been specified to the right of the "=" sign.

A sample crash dump performed under ETOS, with crash files saved on stand-alone OS/8 sys, is listed in Figure 8-11.

Figure 8-11
Sample Crash Dump Under ETOS

```
.SAVE SYS:CRASH2 20000-27777,30000-37777;7600  
.R ETOS  
ETOS V5B  
OPTION? T  
H;0,3 OPERATOR  
  
JOB 3 LOGGED IN ON CONSOLE 0  
.^VS  
!ASSIGN LPT:  
!LOOKUP 3=[0,2]OS8.OS8  
!CONTINUE  
RUN CHN3:CDUMP  
'CDUMP' V5.002  
*LPT:<CHN3:ETOS,CRASH,CRASH2  
.
```

A sample crash dump performed under ETOS, with crash files saved on both portions of an OS/8 pack mounted in RK05 drive 1 is given in Figure 8-12.

Figure 8-12
Sample Crash Dump Under ETOS

```
.SAVE RKAB:CRASH3 20000-27777,30000-37777;7600  
.R ETOS  
ETOS V5B  
OPTION? T  
H;0,3 OPERATOR  
  
JOB 3 LOGGED IN ON CONSOLE 0  
.^VS  
!ASSIGN LPT:  
!LOOKUP 3=[0,2]OS8.OS8  
!ASSIGN DK1:  
!LOOKUP 4=DK1:  
!LOOKUP 5=DK1:,6260  
!CONTINUE  
RUN CHN3: CDUMP  
'CDUMP' V5.002  
*LPT:<CHN3:ETOS,CHN4:CRASH1,CRASH2,CHN5:CRASH3  
.
```

Figure 8-13 contains a sample crash dump performed under ETOS, with crash files, which are transferred to an ETOS account before the dump.

Figure 8-13
Sample Crash Dump Under ETOS

```
.SAVE DTA0:CRASH4 20000-27777,30000-37777;7600
.R ETOS
ETOS V5B
OPTION? T
H;0,3 OPERATOR

JOB 3 LOGGED IN ON CONSOLE 0

.^VS
!ASSIGN LPT:
!LOOKUP 3=[0,2]OS8.OS8
!PRIV 4040
!CONTINUE
COPY CHN2:<DTA0:CRASH?.SV
FILES COPIED:
CRASH1.SV
CRASH2.SV
CRASH3.SV
CRASH4.SV
RUN CHN3:CDUMP
'CDUMP' V5.002
*LPT:<CHN3:ETOS,CHN2:CRASH1,CRASH2,CRASH3,CRASH4
.
```

The previous examples output crash dump listings to the line printer. The output specification may be any OS/8 file. If no line printer is available, TTY: or LT: may be used to obtain a listing on the terminal.

The first input specification must be the file ETOS.SV that was running when the crash occurred. One to eight crash files must also be specified for input.

Following is a list and explanation of all error messages which may be printed by the CDUMP program.

BAD INPUT DEVICE

The input device handler is not co-resident with the system device. Under single-user operation, only the system disk's handlers are co-resident. Under ETOS, all channel handlers are co-resident.

BAD OUTPUT DEVICE OR NAME

The output device handler cannot be loaded or the output file cannot be created. If the output device is file structured, there may not be enough room on the device to stop the crash dump.

ERROR IN CLOSING OUTPUT FILE

The output file cannot be closed properly. This error indicates a serious problem with the output device.

ERROR ON OUTPUT DEVICE

An error occurred while attempting to write a section of the crash dump to the output device. Unless the output device has been inadvertently turned off-lines or write-protected, this error indicates a bad block on the output device or a hardware problem.

ERROR READING INPUT FILE

An error occurred while attempting to read from one of the specified input files. Unless the input device has been inadvertently turned off-line, this error indicates a bad block on the input device or a hardware problem.

MUST HAVE AT LEAST TWO INPUT FILES

To obtain a crash dump, you must compare the file ETOS.SV with at least one crash file.

8.8 DEBUGGING THE MONITOR

If you are having serious problems with ETOS, QUODATA may request that you run a version of ETOS which has a debugging tool built in. This debugging tool (XDDT7), which is stored on stand-alone OS/8 SYS, uses the upper 4K words of physical memory. Since 8K words of user memory are required and 8K words is taken up by the ETOS resident monitor, this procedure cannot be performed on a machine which contains less than 32K words of memory. To create a debugging version of the monitor, use the procedure of Figure 8-14 under stand-alone OS/8.

Figure 8-14
Creating a Debugging Version of ETOS

```
.R CONFIG
'CONFIG' V3.012
*MEMTAB,7=1
MEMTAB(00007) 0000 0001
*EXIT
.R PIP
*SYS:XETOS.SV<SYS:XDDT.SV,ETOS.SV/I
*^C
.
```

After this debugging monitor is saved, you can run it by entering the commands of Figure 8-15.

Figure 8-15
Sample Execution of Debugging Version of ETOS

```
.R XETOS
l#
4000' ETOS V5B
OPTION? T

JOB 03 LOGGED IN ON CONSOLE 00
.
```

8.9 EXAMINING PHYSICAL MEMORY

With the PEEK command (see 6.2.2), you can examine the contents of physical memory. However, if the contents of the locations which you are examining are changing rapidly, you cannot keep up with the changes. If you have a front panel, you can watch changes being made to addresses in physical memory. The front panel is normally set to display the accumulator so you can see the null job and determine the use of ETOS. However, the examination of memory requires that you display the Multiplier Quotient (MQ). To display the MQ on a PDP8/E, turn the rotary switch on the front panel to the position labeled MQ and insure that the key switch is set to power, not panel lock. To display the MQ on a PDP8/A with a KC8-AA programmer's console, press the "1" button and then the DISP button on the front panel. To set the field, which you wish to examine, set bits 0-8 to zero. On a PDP8E, this is accomplished by setting switches 0-8 down. On a PDP8/A, this is accomplished by pressing the "0" button on the front panel three times. When bits 0-8 have been set to 0, you can set bits 9-11 to the field which you wish to examine. On a PDP8/E, this is accomplished by treating switches 9-11 as an octal number in the range 0-7. On a PDP8/A, press the number button corresponding to the field you wish to examine and press the LSR button. Now that the field is set, set the switch register to the address you wish to examine. On a PDP8/E this is accomplished by setting the switches to the twelve-bit address you wish to examine. On a PDP8/A, press four numeric buttons to represent the four octal numbers in the address and press the LSR button.

To change the address you are examining, change the switch register. To change the field you are examining, set bits 0-8 to zero and bits 9-11 to the desired field. You cannot examine locations 0-7 in any field, since this requires setting bits 0-8 to zero, which resets the field.

8.10 SOFTWARE PERFORMANCE REPORTS

If you have a problem with one of the ETOS components, fill out a Software Performance Report. These forms are contained in the release notes. When we receive the Software Performance Report, we will respond in writing within thirty days. The response could be a patch, an explanation or an acknowledgement of the problem with a time frame for its solution. Written communication is essential for proper support. We also request that you write to us about new features which you would like to see in the next version. Most of the features of a new release are decided by tabulating these customer requests.

The Software Performance Report is your best method of communicating with us. Our method of communicating with you is the Technical Summaries. These summaries are discussed in the next section.

8.11 TECHNICAL SUMMARIES

When we begin to work on a problem with one of the ETOS components, we consult any program discrepancy forms which deal with the problem. If more information is required, we contact the customers who sent in the forms. The next step in the debugging process is the recreation of the problem at QUODATA. Once we can reproduce the problem, the only step remaining is to devise the best method for the problem's solution. This solution is normally a modification to the ETOS software. However, it may be an operational solution, which requires a different method of processing. This solution is implemented in our development system. After the solution is tested, we test the solution further by having a customer implement it into his system. After we are confident that the solution is valid, we distribute it to all users via the Technical Summaries.

Technical Summaries are periodic mailings from QUODATA, which allow us to support a large number of customers in the most effective manner. These Summaries contain solutions to customer problems, which are devised by the methods discussed previously. The Summaries also contain documentation changes, notification of new products and general information about ETOS.

The information of the Technical Summaries may be communicated through technical summary letters, by documentation page replacements, by optional patches and by mandatory patches. Besides containing general information about ETOS and notification of new products, the technical summary letters (see Appendix D) can also act as a forum for user discussions. If you have a program of general interest, send us a description of the program. We will include the description and your installation name and address in the next Technical Summary. Any users who are interested in the program can contact you directly. QUODATA does not distribute user software. You should also write to us if you require a program to perform a specified task. We will include your request in the next technical summary.

Documentation are distributed in the form of replacement pages. We distribute the manuals in three-ring binders. New documentation is inserted, replacing the earlier copy of the page. Therefore, your manuals are always up-to-date. We make these changes to our master copies of the manuals, so that any new printings are up-to-date. If you have more than one copy of the manuals, you must reproduce the replacement pages for insertion in all copies.

Modification of software is documented in the Software Maintenance Reports, contained in the Technical Summaries. These maintenance reports contain solutions to problems reported in the Software Performance Reports. The maintenance reports are divided into three forms. One form of the report is a mandatory

patch. This patch contains all of the dialogue necessary to enter the change to the system. User input is underlined to distinguish it from computer output. All mandatory patches (see Appendix F) should be made, even if you are not concerned with the problem fixed by the patch. We must have all software at the same patch level to maintain proper support.

If you don't have the license for the product which is changed, ignore the modification. Entry of these mandatory patches are discussed further in the next section.

The second form of the Software Maintenance Report is the optional patch (see Appendix E). These patches are not fixes for software errors or bugs. Optional patches change the software in a manner which is not desirable to every user. It is your decision whether or not to implement these patches.

When we release a Technical Summary, we make all mandatory patches to our Master Distribution Pack. No optional patches are made. Therefore, any packs which are sent out from QUODATA are always up to the patch level of the most recent summary. When a distribution pack is sent to a new ETOS customer, all mandatory patches have been performed on this new distribution pack. The distribution pack contains the release of ETOS (V5B) followed by an asterisk (*) and a number, representing the last technical summary patches which have been implemented.

It is necessary that software be kept standard for proper support. Any user modifications to the system might cause an invalidation of the warranty. When we receive notification of changes to ETOS, we will respond as to what effect these changes will have on your system. To insure that a user requesting support has implemented all required patches, every program under ETOS has been assigned a release number. Every mandatory patch changes the release number of the particular product. The

general form of the release number is X.YYZ, where X is the version number, YY is the edit number and Z is the patch level. After a release of ETOS, Z is the only number that changes. Z is initialized at 0 and is incremented by 1 in each mandatory patch. The overall version number for ETOS is V5B. The specification release number for ETOS.SV is 05.20Z, where Z is the patch level.

8.12 MEDIA MANAGEMENT

The Technical Summaries contain mandatory patches which all users are required to enter in their system. It is critical that these patches be entered exactly as written. This poses an important problem, particularly at sites where the system manager is not an experienced computer person familiar with the general rules of properly managing computer systems.

The proper operation of an ETOS system requires at least four disk packs:

- Pack 1 - Original ETOS distribution pack from QUODATA. This pack should never be modified in any way. It is rarely used, and it should always be write-locked when it is used.
- Pack 2 - SMR pack. This pack contains all SMR (Software Maintenance Report) corrections.

When you install ETOS, create this pack with RKCOPY or SICOPY by copying an image of the ETOS distribution pack (pack 1) onto a scratch pack, which becomes the SMR pack. Whenever you receive a technical summary, enter the patches (alias SMRs) onto this pack. Thus, the SMR pack is a completely up-to-date version of the ETOS distribution pack. It includes any modifications to configure the pack for the user's installation, such as CONFIG options. Once the SMRs have been correctly entered onto the SMR pack, the modified files and system heads may be copied onto the Production pack.

- Pack 3 - Production pack. This is the pack that is used as the ETOS system device in the daily operations of the installation. It is fully configured for the specified needs of the particular site. Some sites may have more than one production pack.

Pack 4 - Production Daily Backup. This pack is an image of the Production pack. It is used to restore data lost through either human or system error. It is always good form to have a backup pack, and in some cases it is vital. Once a day, RKCOPY or SICOPY should be used to copy the Production pack to the Production Daily Backup Pack.

Pack 5 - Production Weekly Backup. Similar to the Daily Backup, but this pack provides access to older information. It is valuable because one does not always immediately realize that he has deleted an important file or that the system is corrupted, so the daily backup may be made before the data has been recovered. A Weekly Backup Pack provides additional security against data loss.

Most industrial installations should maintain both a Daily Backup and a Weekly Backup. The time it takes to do so is minimal. However, some noncritical users, particularly schools, may not wish to tie up two disk packs for backup. We strongly advise that at least one form of backup be performed. The decision of whether to perform a Daily or a Weekly backup depends on the usage of the system and the type of data that the system manager is most interested in preserving.

8.12.1 Entering SMRs

As noted before, it is critical that the system manager correctly enter his SMRs, and that he keep track of system modules to avoid patching corrupt or improperly modified files. In order to correctly enter a SMR, all of the preceding required SMRs for the particular component must have been correctly entered. For example, if you are installing the SMR for CONFIG.SV sequence 1-3, then CONFIG.SV sequence 1-2 must have already been installed. Moreover, no other modifications should have been made to the module, such as an early patch received over the telephone. If you maintain the four packs described earlier, and promptly enter SMRs, there should be no problem.

After each SMR has been installed, you should double-check your output to insure that no errors or irregularities occurred. All files (e.g., FILE BCOMP.SV), access modes (e.g., SET MODE SAVE),

addresses, old contents (typed out), and new contents (entered) should match, with the following exceptions. The text "xxxx", or generally anything in lower case text in the SMR indicates data that is not expected to match the user's actual output. The output of this command contains block numbers for the location of the program, which will probably be different at each site. Also, certain documentation conventions are used which will not themselves appear in the user's output. For example, all data which the user enters is underlined in the SMR report. When underlined text is the last item on the line, it should be terminated with a carriage return, unless otherwise indicated. The text "<CR>" and "<LF>" is used to explicitly denote what terminating character should be used. "<CR>" stands for the carriage return key, and "<LF>" stands for the linefeed key.

It is a good idea to back up your SMR pack (pack 2) onto a scratch pack before entering SMRs. This way, if you do make any mistakes, or if irregularities turn up, it is easier to recover.

All functions of the system manager should be performed on a hard copy terminal. You should save your hard-copy terminal output from every SMR entering session in a permanent file. This hard copy should be double-checked for errors at the end of the session, as described above. It can be checked later if problems arise, and it may be necessary for QUODATA personnel to examine this material.

Obviously, you should always use a hard-copy terminal rather than a scope while entering SMRs. If this is not possible, there is an alternate method of obtaining hard copy which you should use. Most patches are entered with FUTIL. FUTIL has an option to allow you to store your entire terminal dialog in an OS/8 file. This method is not as good as using a hard-copy terminal, but because the non-FUTIL parts of the dialog are not saved, but it is far better than nothing. You direct FUTIL to save its

terminal dialog on the line printer with the commands of Figure 8-16.

Figure 8-16
Output of FUTIL Dialogue to Line Printer

<u>.R FUTIL</u>	(this command is part of the SMR)
<u>!VASSIGN LPT</u>	(insert these 3 commands into SMR flow
<u>SET DDEV LPT</u>	immediately after entering FUTIL to
<u>SET DMODE ALL</u>	save the dialog on the LPT)
(continue with SMR)	
.	
.	
<u>WRITE</u>	(part of SMR)
<u>CLOSE</u>	(insert this just before EXIT - closes
	LPT file)
<u>EXIT</u>	(part of SMR)

The line printer output should be used to double-check the SMR entry, and it should also be saved in the permanent file discussed above.

There is a further technique that may be of interest to users who have the OS/8 EXTENSION KIT. Both FUTIL and CONFIG are compatible with OS/8 BATCH. Moreover, in ETOS V5B, OS/8 BATCH has been modified so that it will handle SCALE commands. Using BATCH streams has advantages in that it gives you better control over what gets done, and makes it far simpler to re-enter SMRs. If OS/8 BATCH is used, listings of the BATCH streams and the BATCH log should be kept in your permanent file of terminal output. The example of Figure 8-17 should give you some idea of how to prepare the BATCH streams.

Figure 8-17
Sample FUTIL Input Batch

```
$JOB ENTERS ETOS V5A SMR FOR OS/8 BASIC, SEQ 1
!PRIV 4040
!WE 0
!PRIV 40
.R FUTIL
FILE BASIC.FF
SET MODE SAVE
13402/7775
```

(Note that linefeeds may be entered into the BATCH stream just as you would type them on a terminal)

```
4550
4477
4477
3230
3230
1206
```

```
WRITE
:
:
1322/2201
WRITE
EXIT
!PRIV 4040
!WL 0
!PRIV 40
$END
```

8.12.2 Maintaining The Production Pack

All of the system manager's functions should be performed on a hard-copy terminal. The terminal output should be saved in a permanent file that documents all modifications related to the system management. Such modifications include entering SMRs onto the SMR pack, copying SMR mods onto the Production pack, initial building and configuration of the Production pack, and all modifications to the account structure. Such data may be crucial in the diagnosis of any problems that occur. If it is impossible to use a hard-copy terminal, you should make an effort to obtain hard-copy documentation through the line printer. The ACCNT program's LIST command, FUTIL's SET DDEV LPT and SET DMODE ALL commands (described above), and the OCOMP program (to document

changes made) can be used for this purpose.

After entering an SMR onto the SMR pack, you can copy the modified modules to the production pack. OS/8 and COS files can be copied with OS/8's COPY command. OS/8 system heads can be copied using OS/8 PIP's /Y option. COS system heads can be copied under OS/8 or COS with the program COSBLD.SV.

E.5 DISABLING CHANIO COMMANDS

There are ten non-privileged SCALE commands which are used to manipulate ETOS channels. These commands are known as CHANIO commands (see 7.2, System User's Guide). These commands are used to manipulate (e.g., create, delete, change the length of) ETOS files. On a system with hostile users, these commands can be abused. Variable length ETOS files, such as OS/8 or COS storage areas may be EXTENDED or REDUCED. All of the available space on the disk could be wasted in this manner. Another common problem is the deletion of the OS/8 scratch blocks file on channel 1.

The solution to these problems of users abusing the CHANIO command is to disable these commands. After this change is made, if a non-privileged user attempts to use a CHANIO command, the message "?UNKNOWN COMMAND" is printed. If the SCALE commands are disabled, the program CHANIO IOTs (see 7.4, System User's Guide) may still be utilized by non-privileged users. Therefore, the LOOKUP program (see 5.14) may be run by non-privileged users. This allows you to disable direct use of all CHANIO operations except LOOKUP. LOOKUP is often used to facilitate access of an auxiliary storage area or a common program library. If you wish to disable LOOKUP also, delete the LOOKUP program from the virtual OS/8 system area.

Figure E-6 contains the dialogue which is followed in order to make the CHANIO commands privileged. This dialogue must be followed under stand-alone OS/8.

Figure E-6
Disabling CHANIO Commands

.R CONFIG

```
'CONFIG' V3.015
*KMON,2265=5334<CR>
KMON(02265) 5310 5334
*KMON,2307/0
KMON(02307) / 0000 : 3532<LF>+
KMON(02310) / 7301 : 1513<LF>+
KMON(02311) / 4523 : 3535<LF>+
KMON(02312) / 0001 : 1514<LF>+
KMON(02313) / 7650 : 3541<LF>+
KMON(02314) / 5775 : 0516<LF>+
KMON(02315) / 4557 : 6233<LF>+
KMON(02316) / 3124 : 0521<LF>+
KMON(02317) / 6237 : 2747<LF>+
KMON(02320) / 0000 : 0522<LF>+
KMON(02321) / 4557 : 2753<LF>+
KMON(02322) / 3072 : 6654<LF>+
KMON(02323) / 5400 : 3321<LF>+
KMON(02324) / 3075 : 3514<LF>+
KMON(02325) / 2440 : 3325<LF>+
KMON(02326) / 3102 : 7201<LF>+
KMON(02327) / 4600 : 3330<LF>+
KMON(02330) / 3532 : 7200<LF>+
KMON(02331) / 1513 : 2357<LF>+
KMON(02332) / 3535 : 6451<LF>+
KMON(02333) / 1514 : 0000<LF>+
KMON(02334) / 3541 : 7301<LF>+
KMON(02335) / 0516 : 4523<LF>+
KMON(02336) / 6233 : 0001<LF>+
KMON(02337) / 0521 : 7650<LF>+
KMON(02340) / 2747 : 5775<LF>+
KMON(02341) / 0522 : 4557<LF>+
KMON(02342) / 2753 : 3124<LF>+
KMON(02343) / 6654 : 6237<LF>+
KMON(02344) / 3321 : 0000<LF>+
KMON(02345) / 3514 : 4557<LF>+
KMON(02346) / 3325 : 3072<LF>+
KMON(02347) / 7201 : 5400<LF>+
KMON(02350) / 3330 : 3075<LF>+
KMON(02351) / 7200 : 2440<LF>+
KMON(02352) / 2357 : 3102<LF>+
KMON(02353) / 6451 : 4600<CR>C
*EXIT<CR>
```

E.6 ALTERING THE TELEPRINTER AND LINE PRINTER CHARACTERISTICS IN COMMERCIAL BASIC

As distributed, the teleprinter width is set in the CBASIC software to 132_{10} (204_8) spaces. The cells created by the comma (,) construction of the PRINT statement are 14_{10} (16_8) spaces wide.

These settings result in a ten column wide output when one uses the comma construction.

The output construction can be changed by altering two locations in CBRTS.SV. To change the output characteristics to 80_{10} spaces with 5 columns, 16 spaces wide, follow the dialogue of Figure E-7.

Figure E-7
Changing the Comma Construction
of Commercial BASIC

```
^VS  
!PRIV 4040<CR>  
!WE 0<CR>  
!BOOT<CR>  
  
.R FUTIL<CR>  
  
SET DEVICE SYS<CR>  
FILE CBRTS<CR>  
CBRTS.SV ssss-eeee 0030 (0024) S.SSS 16-AUG-79  
SET MODE SAVE<CR>  
17/ 7574 7660<CR>  
2571/ 0016 0020<CR>  
WRITE<CR>  
EXIT<CR>
```

Location 17 in Figure E-7 is the negative of the octal value of the total number of teleprinter spaces. Eighty spaces is 120_8 and -120_8 is 7660_8 .

Location 2571 in Figure E-7 is the column width in octal.

With this information, the user may alter his teleprinter output to any desired arrangement.

The line printer width in Commercial BASIC is controlled by the OS/8 SET program.

To set the width to 70_{10} in order to properly space out 5 columns of 14 spaces each, follow the dialogue of Figure E-8, using a

PRIVileged account.

Figure E-8
Setting Line Printer Width in Commercial BASIC

```
.^VS  
!PRIV 4040<CR>  
!WE 0<CR>  
  
.SET LPT WIDTH 70  
  
.
```

If you do change the line printer width, it is recommended that the behavior of the line printer be checked for its effect on other operations.