

```

EEEEEEEEEEEEEEEE RRRRRRRRRRR RRRRRRRRRRR FFFFFFFF FFFFFFFF MMM MMM TTTTTTTTTTTTTT
EEEEEEEEEEEEEEEE RRRRRRRRRRR RRRRRRRRRRR FFFFFFFF FFFFFFFF MMM MMM TTTTTTTTTTTTTT
EEEEEEEEEEEEEEEE RRRRRRRRRRR RRRRRRRRRRR FFFFFFFF FFFFFFFF MMM MMM TTTTTTTTTTTTTT
EEE RRR RRR RRR RRR FFF MMMMMM MMMMMM TTT
EEE RRR RRR RRR RRR FFF MMMMMM MMMMMM TTT
EEE RRR RRR RRR RRR FFF MMMMMM MMMMMM TTT
EEE RRR RRR RRR RRR FFF MMM MMM MMM TTT
EEE RRR RRR RRR RRR FFF MMM MMM MMM TTT
EEEEEEEEEEEEEEEE RRRRRRRRRRR RRRRRRRRRRR FFFFFFFF FFFFFFFF MMM MMM TTT
EEEEEEEEEEEEEEEE RRRRRRRRRRR RRRRRRRRRRR FFFFFFFF FFFFFFFF MMM MMM TTT
EEEEEEEEEEEEEEEE RRRRRRRRRRR RRRRRRRRRRR FFFFFFFF FFFFFFFF MMM MMM TTT
EEE RRR RRR RRR RRR FFF MMMMMM MMMMMM TTT
EEE RRR RRR RRR RRR FFF MMMMMM MMMMMM TTT
EEE RRR RRR RRR RRR FFF MMMMMM MMMMMM TTT
EEE RRR RRR RRR RRR FFF MMMMMM MMMMMM TTT
EEE RRR RRR RRR RRR FFF MMMMMM MMMMMM TTT
EEEEEEEEEEEEEEEE RRR RRR RRR RRR FFF MMMMMM MMMMMM TTT
EEEEEEEEEEEEEEEE RRR RRR RRR RRR FFF MMMMMM MMMMMM TTT
EEEEEEEEEEEEEEEE RRR RRR RRR RRR FFF MMMMMM MMMMMM TTT

```

: : : D

```

EEEEEEEEEE  RRRRRRRR  RRRRRRRR  FFFFFFFFFF  MM      MM  TTTTTTTTTT
EEEEEEEEEE  RRRRRRRR  RRRRRRRR  FFFFFFFFFF  MM      MM  TTTTTTTTTT
EE          RR      RR  RR      RR  FF      MM  MM  MTT
EE          RR      RR  RR      RR  FF      MM  MM  MTT
EE          RR      RR  RR      RR  FF      MM  MM  MTT
EE          RR      RR  RR      RR  FF      MM  MM  MTT
EEEEEEEEEE  RRRRRRRR  RRRRRRRR  FFFFFFFFFF  MM      MM  TTT
EEEEEEEEEE  RRRRRRRR  RRRRRRRR  FFFFFFFFFF  MM      MM  TTT
EE          RR  RR    RR  RR    FF      MM      MM  TTT
EE          RR  RR    RR  RR    FF      MM      MM  TTT
EE          RR  RR    RR  RR    FF      MM      MM  TTT
EE          RR  RR    RR  RR    FF      MM      MM  TTT
EEEEEEEEEE  RR      RR  RR      RR  FF      MM      MM  TTT
EEEEEEEEEE  RR      RR  RR      RR  FF      MM      MM  TTT

```

```

LL          IIIIII  SSSSSSSS
LL          IIIIII  SSSSSSSS
LL          II      SS
LL          II      SS
LL          II      SS
LL          II      SS
LL          II      SSSSSS
LL          II      SSSSSS
LL          II      SS
LL          II      SS
LL          II      SS
LL          II      SS
LLLLLLLLLL IIIIII  SSSSSSSS
LLLLLLLLLL IIIIII  SSSSSSSS

```

(2)	123	DECLARATIONS
(3)	332	ERRFMT
(3)	590	ERRFMT KERNAL MODE INIT
(3)	617	TIME STAMP ROUTINE
(3)	646	VOLUME MOUNT/DISMOUNT MESSAGE ROUTINE
(3)	720	GET ERROR LOG BUFFER
(4)	769	ERF\$ERRSNAP
(5)	860	ERF\$SNAPSHOT_PRESENT
(6)	897	ERF\$SNAPSHOT_COPIED

```

0000 1      .TITLE  ERRFMT
0000 2      .IDENT  'V04-000'
0000 3
0000 4
0000 5      *****
0000 6      *
0000 7      *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8      *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9      *  ALL RIGHTS RESERVED.
0000 10     *
0000 11     *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12     *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13     *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14     *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15     *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16     *  TRANSFERRED.
0000 17     *
0000 18     *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19     *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20     *  CORPORATION.
0000 21     *
0000 22     *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23     *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24     *
0000 25     *
0000 26     *****
0000 27
0000 28
0000 29     ++
0000 30     FACILITY:  ERROR LOG FORMAT PROGRAM
0000 31
0000 32     ABSTRACT:  THIS PROGRAM EMPTIES THE ERROR LOG BUFFERS AND CREATES
0000 33     A FILE, ERRLOG.SYS, IN A FORMAT ACCEPTABLE TO ERF.
0000 34
0000 35
0000 36     ENVIRONMENT:
0000 37
0000 38     AUTHOR:  KATHLEEN D. MORSE,          CREATION DATE:  29-JUN-1977
0000 39
0000 40     MODIFIED BY:
0000 41
0000 42     V03-011 TCM0008      Trudy C. Matthews      20-Aug-1984
0000 43     Increase the size of the buffer that we use to receive
0000 44     messages from the venus console during creation of the
0000 45     cpu-specific error log.  Change filename of error snapshot
0000 46     file on the console device from SNAPx.LOG to SNAPx.DAT.
0000 47
0000 48     V03-010 TCM0007      Trudy C. Matthews      19-Jul-1984
0000 49     Add ability to create a CPU-specific errorlog once per
0000 50     initialization of the ERRFMT process.
0000 51
0000 52     V03-009 EAD0171      Elliott A. Drayton    7-May-1984
0000 53     Replace $UPDATE to support time stamps.
0000 54
0000 55     V03-008 EAD0139      Elliott A. Drayton    11-Apr-1984
0000 56     Changed output FAB to allow shared read access.
0000 57

```

```
0000 58 : V03-007 CWH1002 CW Hobbs 1-Mar-1983
0000 59 : Convert the errlog pid to an extended pid for the $delprc.
0000 60 :
0000 61 : V03-006 TCM0006 Trudy C. Matthews 30-Dec-1982
0000 62 : Fix bug in ERF$TIMSTMP; it modifies R2 but doesn't save the
0000 63 : old value.
0000 64 :
0000 65 : V03-005 TCM0005 Trudy C. Matthews 16-Jul-1982
0000 66 : Fix problem in V03-004 that caused a new ERRLOG.SYS to be
0000 67 : created each time the system was re-booted.
0000 68 :
0000 69 : V03-004 TCM0004 Trudy C. Matthews 24-Jun-1982
0000 70 : When opening ERRLOG.SYS, check that its the same file we
0000 71 : accessed the last time. If not, create a new version.
0000 72 :
0000 73 : V03-003 ROW0080 Ralph O. Weber 08-APR-1982
0000 74 : Move DEVFAO control string so that it is not in the middle
0000 75 : of the "ERROR ACCESSING ERROR LOG FILE" message text.
0000 76 :
0000 77 : V03-002 STJ0251 Steven T. Jeffreys 01-Apr-1982
0000 78 : Do not send mount/dismount notification messages to
0000 79 : OPCOM depending on the appropriate sysgen parameter.
0000 80 :
0000 81 : V03-001 STJ0228 Steven T. Jeffreys 19-Mar-1982
0000 82 : Use full device name when calling $GETDVI.
0000 83 :
0000 84 : V02-012 LMP0014 L. Mark Pilant, 16-Mar-1982 11:15
0000 85 : Fix a problem with the setting of the desired operator
0000 86 : bits. Also, fix a problem with using EFN 0 with GETDVI.
0000 87 :
0000 88 : V02-011 LMP0007 L. Mark Pilant 13-Jan-1982 9:55
0000 89 : Notify the appropriate operators when volume mount and
0000 90 : dismount messages area seen.
0000 91 :
0000 92 : V02-010 SPF0045 Steve Forgey 28-Dec-1981
0000 93 : Synchronize buffer copy with allocation interlock flag.
0000 94 :
0000 95 : V02-009 PHL0013 Peter H. Lipman 21-Aug-1981
0000 96 : Change the output file specification for the error log file
0000 97 : to use the new system wide logical name SYSS$ERRORLOG whic
0000 98 : is the [SYSERR] directory on the system disk.
0000 99 :
0000 100 : V02-008 TCM0003 Trudy C. Matthews 6-Aug-1981
0000 101 : Change message sent to oerator's terminal when ERRFMT
0000 102 : deletes itself.
0000 103 :
0000 104 : V02-007 KDM0059 Kathleen D. Morse 22-Jul-1981
0000 105 : Fix new file error log message.
0000 106 :
0000 107 : V02-006 KDM0057 Kathleen D. Morse 15-Jul-1981
0000 108 : Add SID to error log buffer message format and make the
0000 109 : header fields be negative offsets from the message text.
0000 110 :
0000 111 : V02-005 TCM0002 Trudy C. Matthews 13-Jul-1981
0000 112 : Document use of @SYSS$SYSTEM:STARTUP ERRFMT instead of
0000 113 : @SYSS$MANAGER:ERF$START to re-start ERRFMT process after it
0000 114 : deletes itself.
```

0000 115 :
0000 116 :
0000 117 :
0000 118 :
0000 119 :
0000 120 :
0000 121 :--

V02-004 STJ0024 Steven T. Jeffreys 01-Feb-1981
Fixed bugs in mailbox write logic.

DECLARATIONS

```

0000 123      .SBTTL  DECLARATIONS
0000 124      :
0000 125      : INCLUDE FILES:
0000 126      :
0000 127      :
0000 128      :
0000 129      : MACROS:
0000 130      :
0000 131      :
0000 132      : EQUATED SYMBOLS:
0000 133      :
0000 134      $PRDEF      ; DEFINE PROCESSOR REGISTERS
0000 135      $DCDEF      ; DEFICE DEVICE CLASS TYPES
0000 136      $DIBDEF     ; DEVICE INFORMATION BUFFER
0000 137      $DVIDEF     ; $GETDVI MESSAGE CODES
0000 138      $SEMBETDEF  ; ERROR MESSAGE ENTRY TYPES
0000 139      $SEMBDEF   ; DEFINE ERROR MESSAGE BUFFER HEADER
0000 140      $SEMBTSDEF  ; DEFINE TIME STAMP DEFINITIONS
0000 141      $ERFHDDEF  ; ERROR FORMAT HEADER DEFINITIONS
0000 142      $ERFSTSDEF  ; ERROR FORMAT TIME STAMP DEFINITIONS
0000 143      $ERFVMDEF  ; ERROR FORMAT VOLUME MOUNT DEFINITIONS
0000 144      $ERLDEF     ; SYSTEM ERROR LOGING DEFINITIONS
0000 145      $OPCDEF     ; OPERATOR MESSAGE DEFINITIONS
0000 146      $PCBDEF     ; PROCESS CONTROL BLOCK DEFINITIONS
0000 147      $SSDEF     ; DEFINE STATUS CODES
0000 148      :
00000002 0000 149  ERM$C_FORMAT = 2      ; FORMAT NUMBER FOR VAX
000000FF 0000 150  ERF$C_LOOP_CNT = 255  ; TIMES TO WAIT FOR BUFFER
00000258 0000 151  ERF$K_DLTA_STMP = <60*10> ; TIME STAMP DELTA IN SECS
FF676980 0000 152  ERF$K_CLK_TICK = -<10*1000*1000> ; CONVERSION TO CLOCK TICKS/SEC
0000 153      :
0000 154      :
0000 155      : OWN STORAGE:
0000 156      :
0000 157      :
00000000 0000 158      .PSECT  DATA,RD,WRT,NOEXE,PAGE
0000 159      :
00000200 0000 160  INBUF:  .BLKB  512      ; INPUT BUFFER
0200 161  OUTFAB: $FAB      -            ; RECORD ACCESS BLOCK
0200 162      FAC=<PUT,UPD>,-          ; PUT AND UPDATE FILE ACCESS
0200 163      FNA=OUTNAM,-           ; FILE NAME ADDRESS
0200 164      FNS=OUTNAMSZ,-         ; LENGTH OF FILE NAME
0200 165      NAM=NAMEBLOCK,-        ; ASSOCIATED NAME BLOCK
0200 166      RFM=VAR,-              ;
0200 167      FOP=CIF,-              ;
0200 168      SHR=<GET,UPI>,-        ;
0200 169      ORG=SEQ,-              ; SEQUENTIAL ORGANIZATION
0200 170      MRS=0                  ; MAX RECORD SIZE UNSPECIFIED
0250 171      :
0250 172  OUTRAB: $RAB      -            ; RECORD ACCESS BLOCK
0250 173      ROP=<EOF,WBH>,-         ; OPEN TO END OF FILE
0250 174      MBC=1,-                ;
0250 175      MBF=2,-                ;
0250 176      RAC=SEQ,-              ;
0250 177      FAB=OUTFAB             ; FILE ACCESS BLOCK ADDR
0294 178      :
0294 179  NAMEBLOCK:        ; NAME BLOCK ASSOCIATED WITH OUTFAB

```

DECLARATIONS

```

0000'0000'0000' 0294 180 $NAM
00 02F4 181 OUTFID: .WORD 0[3] ; SAVED FILE ID
00000000 02FA 182 LASTENTRY: .BYTE 0 ; ENTRY TYPE OF LAST RECORD WRITTEN
0000 02FB 183 SID: .LONG 0 ; SYSTEM ID #
0000 02FF 184 ERF$W_MBXCHN: .WORD 0 ; DIAGNOSTIC MAILBOX CHANNEL
0000 0301 185 ERF$W_MBXSIZ: .WORD 0 ; DIAGNOSTIC MAILBOX SIZE
0000 0303 186 ERF$W_MBXUNT: .WORD 0 ; PREVIOUS DIAG MBX UNIT #
0305 187
55 21 43 41 21 5F 0000030D'010E0000' 0305 188 DEVFAO: .ASCID /_!AC!UW:/ ; $FAO control string to format device
3A 57 0313
0315 189
0315 190
0315 191 ; MESSAGE SENT TO OPERATOR UPON FAILURE TO WRITE TO ERROR LOG FILE.
0315 192
0315 193 OPRMSG_DSC:
0315 194 CPRMSG_LEN:
00000031' 0315 195 .LONG OPRMSG_END-OPRMSG ; SIZE OF OPERATOR MESSAGE BUFFER
00000325' 0319 196 .LONG OPRMSG ; ADDRESS OF OPERATOR MESSAGE BUFFER
031D 197 ROMSG_DSC:
00000100' 031D 198 .LONG ROMSG_END-ROMSG
00000356' 0321 199 .LONG ROMSG
0325 200 OPRMSG:
00000103 0325 201 .LONG OPC$ RQ_RQST!- ; TYPE OF MESSAGE
0329 202 <<OPC$M_NM_CENTRL@B>> ; OPERATOR TO INFORM
00000000 0329 203 .LONG 0 ; NOBODY TO RESPOND TO
52 52 45 20 2D 20 54 4D 46 52 52 45 032D 204 .ASCII /ERRFMT - ERROR ACCESSING ERROR LOG FILE/<13><10>
47 4E 49 53 53 45 43 43 41 20 52 4F 0339
46 20 47 4F 4C 20 52 4F 52 52 45 20 0345
OA OD 45 4C 49 0351
0356 205 OPRMSG_END:
0356 206
0356 207 ROMSG:
00000456 0356 208 .BLKB 256 ; HOLDS TRANSLATED STATUS MESSAGE.
0456 209 ROMSG_END:
0456 210 ROMSG_LEN: ; HOLDS TRANSLATED MESSAGE LENGTH.
00000000 0456 211 .LONG 0
045A 212
045A 213 ; MESSAGE SENT TO OPERATOR WHEN WE'VE FAILED TOO MANY TIMES TO WRITE
045A 214 ; TO ERROR LOG FILE.
045A 215
045A 216 BYEMSG_DSC: ; MESSAGE DESCRIPTOR
045A 217 BYEMSG_LEN:
00000080' 045A 218 .LONG BYEMSG_END-BYEMSG ; LENGTH
00000462' 045E 219 .LONG BYEMSG ; ADDRESS
0462 220
0462 221 BYEMSG: ; MESSAGE
00000103 0462 222 .LONG OPC$ RQ_RQST! - ; TYPE OF MESSAGE
0466 223 <<OPC$M_NM_CENTRL@B>> ; OPERATOR TO INFORM
4C 45 44 20 2D 20 54 4D 46 52 52 45 046A 224 .LONG 0 ; NOBODY TO RESPOND TO
54 4D 46 52 52 45 20 47 4E 49 54 45 0476 225 .ASCII /ERRFMT - DELETING ERRFMT PROCESS/<13><10>
OA OD 53 53 45 43 4F 52 50 20 0482
49 46 20 47 4F 4C 20 52 4F 52 52 45 048C 226 .ASCII /ERROR LOG FILE UNWRITABLE/<13><10>
4C 42 41 54 49 52 57 4E 55 20 45 4C 0498
OA OD 45 04A4
45 20 54 52 41 54 53 45 52 20 4F 54 04A7 227 .ASCII /TO RESTART ERRFMT PROCESS, USE '@SYS$SYSTEM:STARTUP ERRFMT'/
53 45 43 4F 52 50 20 54 4D 46 52 52 04B3

```


DECLARATIONS

```

53 59 53 40 22 20 45 53 55 20 2C 53 04BF
52 41 54 53 3A 4D 45 54 53 59 53 24 04CB
   22 54 4D 46 52 52 45 20 50 55 54 04D7
   04E2 228 BYEMSG_END:
   04E2 229 :
   04E2 230 : MOUNT AND DISMOUNT MESSAGE STRINGS
   04E2 231 :
   04E2 232 MOUNT_FA0:
00000036' 04E2 233 .LONG MOUNT_END-MOUNT_MSG ; LENGTH OF CONTROL STRING
000004EA' 04E6 234 .ADDRESS MOUNT_MSG ; ADDRESS OF CONTROL STRING
   04EA 235 MOUNT_MSG:
00000003 04EA 236 .LONG OPC$_RQ_RQST ; TYPE OF MESSAGE (OPERATOR I.B.S.)
00000000 04EE 237 .LONG 0 ; NOBODY TO REPLY TO
22 44 41 21 22 20 65 6D 75 6C 6F 56 04F2 238 .ASCII \Volume '!AD'!ASmounted, on physical device !AS\
20 2C 64 65 74 6E 75 6F 6D 53 41 21 04FE
20 6C 61 63 69 73 79 68 70 20 6E 6F 050A
   53 41 21 20 65 63 69 76 65 64 0516
   0520 239 MOUNT_END:
   0520 240
   0520 241 MOUNT_DSC:
00000080 0520 242 .LONG 128 ; MAX SIZE OF THE MESSAGE
00000528' 0524 243 .ADDRESS MOUNT_BUF ; ADDRESS OF THE MESSAGE BUFFER
   0528 244 MOUNT_BUF:
000005A8 0528 245 .BLKB 128 ; STORAGE FOR FORMATTED MESSAGE
   05A8 246 MOUNT_MNT:
20 000005B0'010E0000' 05A8 247 .ASCID \ \ ; FOR VOLUME MOUNTED MESSAGE
   05B1 248 MOUNT_DMT:
73 69 64 20 000005B9'010E0000' 05B1 249 .ASCID \ dis\ ; FOR VOLUME DISMOUNTED MESSAGE
   05BD 250 :
   05BD 251 : ERROR COUNTERS
   05BD 252 :
   05BD 253 ERFSB_ERRCNT: ; COUNT ERRORS IN WRITING TO
00 05BD 254 .BYTE 0 ; ERRORLOG FILE
   05BE 255 ERFSB_MAXERRCNT: ; MAXIMUM # ERRORS BEFORE DELETING
14 05BE 256 .BYTE 20 ; THIS PROCESS
   05BF 257 :
   05BF 258 :
   05BF 259 : Data structures needed to get the version number and expanded file name of
   05BF 260 : a newly created SYS$ERRORLOG:ERRSNAP.LOG (Venus-specific).
   05BF 261 :
   05BF 262 .ALIGN PAGE
0600 263 ERRSNAP_FAB:
0600 264 .SFAB - ; File Access Block.
0600 265 FNM=<SYS$ERRORLOG:ERRSNAP.LOG>, - ; File name.
0600 266 NAM=ERRSNAP_NAM, - ; Associated NAM block.
0600 267 XAB=ERRSNAP_XAB ; Associated XAB block.
0650 268
0650 269 ERRSNAP_XAB: ; Declare date/time XAB.
0650 270 .SXABDAT
067C 271
067C 272 ERRSNAP_NAM:
067C 273 .SNAM - ; Name block.
067C 274 RSA=ERRSNAP_RSA, - ; Resultant string area address.
067C 275 RSS=NAM$C_MAXRSS ; Use maximum length of resultant string.
06DC 276
06DC 277 ERRSNAP_RSA: ; Resultant string will be returned here.
000007DB 06DC 278 .BLKB NAM$C_MAXRSS

```

DECLARATIONS

```

07DB 279
07DB 280 :
07DB 281 : Data structures used when SPAWNING a sub-process to execute ERRSNAP.COM.
07DB 282 :
07DB 283 ERRSNAP_COM: ; Descriptor for command procedure.
52 45 24 53 59 53 000007E3'01CE0000' 07DB 284 .ASCID /SYS$ERRORLOG:ERRSNAP.COM/
4E 53 52 52 45 3A 47 4F 4C 52 4F 52 07E9
4D 4F 43 2E 50 41 07F5
07FB 285 ERRSNAP_LOG1: ; Initial DCL command if copying SNAP1.
45 4C 49 46 20 24 00000803'010E0000' 07FB 286 .ASCID /% FILENAME := SNAP1.DAT/
50 41 4E 53 20 3D 3A 20 45 4D 41 4E 0809
54 41 44 2E 31 0815
081A 287 ERRSNAP_LOG2: ; Initial DCL command if copying SNAP2.
45 4C 49 46 20 24 00000822'010E0000' 081A 288 .ASCID /% FILENAME := SNAP2.DAT/
50 41 4E 53 20 3D 3A 20 45 4D 41 4E 0828
54 41 44 2E 32 0834
0839 289 ERRSNAP_FLAGS: ; Set NOCLISYM and NOWAIT flags.
00000006 0839 290 .LONG 6
083D 291 ERRSNAP_STATUS: ; Store the exit status of the SPAWned
00000000 083D 292 .LONG 0 ; command procedure here.
0841 293 :
0841 294 : Definitions needed to communicate with 11/790 logical console interface.
0841 295 :
00000030 0841 296 CON$C_REQERL = ^X30 ; Console command to request error
0841 297 ; snapshot file status.
00000031 0841 298 CON$C_INV$SNP1 = ^X31 ; Console command to invalidate SNAP1.DAT
00000032 0841 299 CON$C_INV$SNP2 = ^X32 ; Console command to invalidate SNAP2.DAT
0841 300 ERRSNAP_CONCMD: ; Store command to be sent to console.
00 .BYTE 0
0842 302 ERRSNAP_DATA: ; Store returned data from logical
00000000 0842 303 .LONG 0 ; console interface here.
0846 304 :
0846 305 :
0846 306 : PURE DATA - KEPT IN CODE PSECT FOR LOCALITY
0846 307 :
00000000 0846 308 :
00000000 309 .PSECT CODE, RD, NOWRT, EXE
0000 310 :
0000 311 :
0000 312 : ARGUMENT LIST FOR FILE CREATE TIME STAMP ENTRY
0000 313 :
00000001 0000 314 FILCRE: .LONG 1 ; ONE ARGUMENT
00000023 0004 315 .LONG EMB$K_NF ; NEW FILE TYPE MESSAGE
0008 316 :
0008 317 ERF$Q_DELTA: ; TIME BETWEEN TIME MARKS
0008 318 :
0008 319 : *** .LONG ERF$K_CLK_TICK*ERF$K_Delta_STMP&^X0FFFFFFF
0008 320 :
9A5F4400 0008 321 .LONG ^X09A5F4400 ; LOW 1/2 OF DELTA TIME
000C 322 :
000C 323 : *** .LONG ERF$K_CLK_TICK*ERF$K_Delta_STMP&-32
000C 324 :
FFFFFFFE 000C 325 .LONG ^X0FFFFFFFE ; HIGH 1/2 OF DELTA TIME
0010 326 :
FFB3B4C0 0010 327 ERF$Q_WAIT: .LONG <-10*1000*500> ; # OF 10 MILLISEC INTERVALS
FFFFFFF 0014 328 .LONG -1 ; TO WAIT FOR BUFFER COMPLETION
47 4F 4C 52 4F 52 52 45 24 53 59 53 0018 329 OUTNAM: .ASCII \SYS$ERRORLOG:ERRLOG.SYS\ ; OUTPUT FILE NAME

```

ERRFMT
V04-000

DECLARATIONS

H 7

16-SEP-1984 01:29:26 VAX/VMS Macro V04-00
5-SEP-1984 01:01:54 [ERRFMT.SRC]ERRFMT.MAR;1

Page 8
(2)

ERR
V04

53 59 53 2E 47 4F 4C 52 52 45 3A 0024
00000017 002F 330 OUTNAMSZ = . - OUTNAM

; LENGTH OF OUTPUT NAME

```

002F 332 .SBTTL ERRFMT
002F 333 :
002F 334 :+
002F 335 :+ FUNCTIONAL DESCRIPTION:
002F 336 :
002F 337 : THIS PROGRAM IS AWAKENED FROM HIBERNATION BY THE ERROR LOGG
002F 338 : WHENEVER AN ERROR LOG BUFFER BECOMES FULL. THE ERROR FORM
002F 339 : PROGRAM READS THE FULL BUFFER AND THEN RELEASES IT FOR RE-USE BY
002F 340 : THE ERROR LOGG PROGRAM. THE DATA JUST READ IS RE-ORGANIZED
002F 341 : AND WRITTEN TO A FILE CALLED 'ERRLOG.SYS' IN A FORMAT ACCEPTABLE
002F 342 : TO SYE.
002F 343 :
002F 344 : THE ERROR FORMAT PROGRAM ALSO PLACES TIME STAMP ENTRIES INTO THE
002F 345 : ERROR LOG BUFFER. THESE TIME STAMPS ARE PLACED INTO THE BUFFER
002F 346 : AT REGULAR INTERVALS. HOWEVER, SEQUENTIAL TIME STAMPS ARE NOT
002F 347 : WRITTEN INTO THE FILE, 'ERRLOG.SYS'.
002F 348 :
002F 349 : THE FILE, 'ERRLOG.SYS', IS UPDATED, OR A NEW VERSION CREATED IF
002F 350 : THE MOST RECENT VERSION IS BEING ACCESSED OR DOES NOT EXIST.
002F 351 :--
002F 352 :
0000002F 353 .PSECT CODE, RD, NOWRT, EXE
002F 354 .ENABL LSB
0000 002F 355 .ENTRY ERF$START, 0
00000000'GF 04 91 0031 356 $CMKRNL_S W^ERF$INIT : INITIALIZE THE ERR FORMATER
003E 357 CMPB #PRS SID_TYP790, - : ARE WE EXECUTING ON A VENUS CPU?
0045 358 G^EXE$GB_CPUTYPE
054A'CF 05 12 0045 359 BNEQ PRCBUF : BRANCH IF NO
0047 360 CALLS #0, W^ERF$ERRSNAP : CALL VENUS-SPECIFIC ERROR ROUTINE
004C 361 PRCBUF: $CMKRNL_S W^ERF$GETBUF : GET THE FULL ERROR LOG BUFFER
14 50 E8 0059 362 BLBS RO, PRCNXT : BR IF MESSAGE(S) TO PROCESS
005C 363 $CLOSE FAB=W^OUTFAB : CLOSE THE OUTPUT
0067 364 $HIBER_S : WAIT FOR SOMETHING TO DO
DC 11 006E 365 BRB PRCBUF
0070 366 :
0070 367 : PROCESS NEXT MESSAGE - COME HERE WHEN A BUFFER HAS BEEN COPIED FROM
0070 368 : THE SYSTEM INTO THE LOCAL BUFFER. IF THE FILE IS NOT OPEN,
0070 369 : OPEN THE OUTPUT FILE OR CREATE ONE IF MOST RECENT IS BEING ACCESSED.
0070 370 :
53 D4 0070 371 PRCNXT: CLRL R3 : R3=0 => OPEN EXISTING FILE
0072 372 : R3=0 => CREATE NEW ERRLOG FILE
0072 373 PRCNXT1:
58 0000'CF 9E 0072 374 MOVAB W^INBUF, R8 : GET ADDR OF FIRST MSG
56 01 A8 68 81 0077 375 ADDB3 ERL$B_BUSY(R8), ERL$B_MSGCNT(R8), R6 : GET COUNT OF MESSAGES
CE 13 007C 376 BEQL PRCBUF : BR IF NO MESSAGES TO PROCESS
58 0C C0 007E 377 ADDL #ERL$C_LENGTH, R8 : POINT TO START OF MESSAGES
52 0200'CF 9E 0081 378 MOVAB W^OUTFAB, R2 : SET ADDRESS OF FAB
02 A2 B5 0086 379 TSTW FAB$W_IFI(R2) : IS THE FILE OPEN?
00AB 31 008B 380 BEQL 2$ : BRANCH TO OPEN OR CREATE FILE
008E 381 BRW NXTMSG : FILE ALREADY OPEN; CONTINUE
10 A2 D4 008E 382 2$:
53 D5 0091 383 CLRL FAB$L_ALQ(R2) : CLEAR ALLOCATION
32 12 0093 384 TSTL R3 : OPEN OR CREATE ERRLOG.SYS?
0095 385 BNEQ 5$ : BR TO CREATE NEW FILE
24 50 E9 009E 386 $OPEN FAB=(R2) : OPEN MOST RECENT VERSION
00A1 387 BLBC R0, 4$ : OPEN FAILED; GO CREATE A NEW VERSION
388 :

```



```

          53  D6 0214 503 10$:
52 0200'CF 9E 0214 504
    02 A2  B4 0216 505
          021B 506
          021E 507
          021E 508
          021E 509
          021E 510
          021E 511
          021E 512
          021E 513
          FE39 31 0236 514
          0239 515 MBX:
          5B 5E  D0 0239 516
50 02FF'CF 3C 023C 517
    19 13 0241 518
0303'CF 00000000'GF B1 0243 519
    6A 13 024C 520
          024E 521
          02FF'CF B4 0258 522
          025C 523 30$:
50 00000000'GF 3C 025C 524
    0303'CF 50 B0 0263 525
    2A 13 0268 526
    5E 1C C2 026A 527
    52 5E D0 026D 528
    41424D5F 8F DD 0270 529
    5E DD 0276 530
    00C3 30 0278 531
7E 52 6E C3 027B 532
    52 5E D0 027F 533
          0282 534
          0282 535
          03 50 E8 0291 536
          008D 31 0294 537 40$:
          0297 538 45$:
          6E 20 D0 0297 539
          029A 540
          029A 541
          52 04 A2 D0 02AE 542
0301'CF 06 A2 B0 02B2 543
    50 22 A9 3C 02B8 544 50$:
0301'CF 50 B1 02BC 545
    05 1B 02C1 546
    50 0301'CF B0 02C3 547
          02C8 548 55$:
          02C8 549
          02C8 550
          02C8 551
05BE'CF 05BD'CF 91 02E7 552
    02EE 553
    02EE 554
    50 045A'CF 3C C2F0 555
00000301'EF 50 B1 02F5 556
    05 1B 02FC 557
    50 0301'CF B0 02FE 558
          0303 559 60$:

```

```

INCL R3 ; ERROR COUNT <= MAX ERROR COUNT
MOVAB W^OUTFAB,R2 ; SIGNAL ACCESS FAILURE
CLR W^FABSW_IFI(R2) ; MUST CREATE NEW FILE
$FAB_STORE - ; CLEAR INDICATOR TO OPEN NEW FILE
          ; REINITIALIZE FAB
          FAB=(R2), -
          ORG=SEQ, - ; SEQUENTIAL ORGANIZATION
          MRS=#0, - ; NO MAX ON RECORD SIZE
          FOP=CIF, -
          SHR=<GET,UPI>, -
          RFM=VAR ; VARIABLE LENGTH RECORDS
BRW PRCNXT1 ; GO TRY TO OPEN A NEW FILE
          ; MAILBOX MESSAGES
MOV SP,R11 ; MARK THE STACK
MOVZWL W^ERFSW_MBXCHN,RO ; MBX CHANNEL ALREADY?
BEQL 30$ ; BRANCH ON NONE
CMPW G^EXE$GQ_ERLMBX,W^ERFSW_MBXUNT ; SAME AS LAST TIME?
BEQL 50$ ; YES, GO MAIL THE MSG
$DASSGN_S CHAN=RO ; NO, DEASSIGN OLD CHANNEL
CLR W^ERFSW_MBXCHN ; CLEAR OLD CHANNEL

MOVZWL G^EXE$GQ_ERLMBX,RO ; GET NEW MAIL BOX UNIT
MOVW RO,W^ERFSW_MBXUNT ; SET NEW UNIT TO USE
BEQL 40$ ; BRANCH IF NONE
SUBL #32-4,SP ; ALLOCATE BUFFER IN THE STACK
MOV SP,R2 ; MARK START OF MAIL BOX UNIT
PUSHL #^A/_MBA/ ; SET PROTOTYPE NAME
PUSHL SP ; SET START OF BUFFER
BSBW 100$ ; SET UNIT OF MAILBOX
SUBL3 (SP),R2,-(SP) ; FIND LENGTH OF NAME
MOV SP,R2 ; SAVE POINTER TO NAME
$ASSIGN_S DEVNAM=(R2),- ; ASSIGN A CHANNEL TO
          CHAN=W^ERFSW_MBXCHN; THE DIAGNOSTIC MAILBOX
BLBS RO,45$ ; BRANCH ON SUCCESS
BRW 65$ ; SKIP THE QIO IF FAILED

MOV #32,(SP) ; RESET LENGTH OF BUFFER
$GETCHN_S CHAN=W^ERFSW_MBXCHN,- ; GET SIZE OF MAILBOX
          PRIBUF=(R2) ; I.E., THE MAXIMUM MSG SIZE
MOV 4(R2),R2 ; GET ADDRESS OF DEV CHAR BUFFER
MOVW DIBSW_DEVBUFSIZ(R2),W^ERFSW_MBXSIZE ; GET MAILBOX SIZE
MOVZWL RABSW_RSZ(R9),RO ; GET SIZE OF MESSAGE
CMPW RO,W^ERFSW_MBXSIZE ; MSG TOO LARGE?
BLEQU 55$ ; BRANCH ON OK
MOVW W^ERFSW_MBXSIZE,RO ; TRUNCATE MSG
SQIO_S CHAN=W^ERFSW_MBXCHN,- ; CHANNEL FOR DIAG MBX
          FUNC=#<IOS_WRITEVBLK!IOSM_NOW>,- ; DONT WAIT FOR SUCCESS
          P1=(R7),- ; ADDR OF ERROR MSG
          P2=RO ; SIZE OF MSG
CMPB W^ERFSB_ERRCNT, - ; HAVE WE EXCEEDED THE ERROR
          W^ERFSB_MAXERRCNT ; THRESHOLD?
BLEQ 65$ ; BRANCH IF NO
MOVZWL W^BYEMSG_LEN,RO ; GET LENGTH OF GOODBYE MESSAGE
CMPW RO,ERFSW_MBXSIZE ; MESSAGE TOO LARGE?
BLEQU 60$ ; BRANCH ON OK
MOVW W^ERFSW_MBXSIZE,RO ; TRUNCATE MESSAGE

```

```

ERRFMT
0303 560 $QIO_S - ; NOTIFY MAILBOX THAT PROCESS IS
0303 561 CHAN=W^ERF$W MBXCHN, - ; BEING DELETED.
0303 562 FUNC=#<IOS WRITEVBLK!IOSM_NOW>, -
0303 563 P1=W^BYEMSG, -
0303 564 P2=R0
0324 565 65$: MOVL R11,SP ; RESET THE STACK POINTER
05BE'CF 5E 5B D0 0327 566 CMPB W^ERF$B_ERRCNT, - ; HAVE WE EXCEEDED THE ERROR
05BD'CF 91 032E 567 W^ERF$B_MAXERRCNT ; THRESHOLD?
03 14 032E 568 BGTR 70$ ; BRANCH IF YES
FE06 31 0330 569 BRW NXTMSG ; ELSE GO PROCESS NEXT MESSAGE
0333 570 ;
0333 571 ; IF ERRCNT > MAXERRCNT, DELETE THIS PROCESS TO PREVENT INFINITE LOOPING.
0333 572 ; THE ERRFMT PROCESS CAN BE RESTARTED VIA AN OPERATOR COMMAND FILE.
0333 573 ;
0333 574 70$: $DELPRC_S ; DELETE THIS PROCESS
033E 575 ;
033E 576 ;
033E 577 ; LOCAL SUBROUTINE TO CONVERT BINARY TO ASCII AND STORE RESULT
033E 578 ; IN BUFFER POINTED TO BY R2
033E 579 ;
033E 580 ;
7E 50 50 51 D4 033E 581 100$: CLRL R1 ; ZERO HI 1/2 OF QUAD WORD
6E 30 7B 0A 7B 0340 582 110$: EDIV #10,R0,R0,-(SP) ; GET NEXT DIGIT
50 D5 0345 583 ADDL #^A/0/, (SP) ; FIND THE DIGIT IN ASCII
02 13 0348 584 TSTL R0 ; ANY THING LEFT
F2 10 034A 585 BEQL 120$ ; BR IF NO MORE TO CONVERT
82 8E F6 034C 586 BSBB 110$ ; GET NEXT DIGIT
05 034E 587 120$: CVTLB (SP)+,(R2)+ ; STORE A BYTE
0351 588 RSB ;

```



```

0352 590 .SBTTL ERRFMT KERNAL MODE INIT
0352 591 :++
0352 592 :
0352 593 : ERFS$INIT - INITIALIZE THE ERROR FORMAT PROGRAM
0352 594 :
0352 595 : THIS ROUTINE IS ENTERED AT KERNAL MODE TO DELETE A PRVIOUS COPY
0352 596 : OF THIS PROCESS IF ONE EXISTS, TEHREBY PERMITTING ONLINE REPLACEMENT
0352 597 : OF THE ERROR FORMAT PROGRAM, AS L L AS EASE OF TESTING. ALSO, THE
0352 598 : KERNAL MODE TIMER AST FOR TIME STAMPING THE ERROR LOG IS STARTED.
0352 599 :--
0352 600 :
0352 601 : .ENABL LSB
0352 602 : .ENTRY ERFS$INIT,^M<R2,R3>
52 00000000'EF DE 0354 603 MOVAL ERL$GL_ERLPID,R2 : GET ADDRESS OF CURRENT PID
53 00000000'GF D0 035B 604 MOVL G^SCH$GL_CURPCB,R3 : GET CURRENT PCB
62 D5 0362 605 TSTL (R2) : PID EQUAL ZERO?
1D 13 0364 606 BEQL 10$ : BR IF YES - NO PREVIOUS ERRFMT
62 60 A3 D1 0366 607 CMPL PCB$$_PID(R3),(R2) : MAKE SURE IT IS THIS PROCESS
17 13 036A 608 BEQL 10$ : BR IF SAME PROCESS
50 62 D0 036C 609 MOVL (R2),R0 : GET INTERNAL PID TO R0
00000000'GF 16 036F 610 JSB G^EXE$IPID_TO_EPID : CONVERT TO EXTENDED PID
62 50 D0 0375 611 MOVL R0,(R2) : SAVE IT IN THE SAME PLACE FOR DELPRC
0378 612 $DELPRC S PIDADR=(R2) : DELETE OLD ERRFMT
62 60 A3 D0 0383 613 10$: MOVL PCB$$_PID(R3),(R2) : SET THE PID FOR THIS PROCESS
02FB'CF 3E DB 0387 614 MFPR #PR$_SID,W^SID : GET SYS ID REGISTER
18 11 038C 615 BRB 30$

```

TIME STAMP ROUTINE

```

038E 617          .SBTTL  TIME STAMP ROUTINE
038E 618
038E 619      :++
038E 620
038E 621      : ERF$TIMSTMP - TIME STAMP
038E 622
038E 623      : THIS ROUTINE IS ENTERED PERIODICALLY TO ENTER A TIME STAMP INTO
038E 624      : THE ERROR MESSAGE BUFFER.  THEY ARE REMOVED ALONG WITH ANY OTHER
038E 625      : ENTRIES MADE BY THE MAIN LINE OF THIS PROGRAM.
038E 626
038E 627      : THIS ROUTINE HAS NO INPUTS AND ONLY OUTPUT IS THE ENTRY OF THE
038E 628      : TIME STAMP IN THE ERROR LOG BUFFER.  IF A BUFFER CAN NOT BE
038E 629      : ALLOCATED, THAT TIME STAMP IS LOST.
038E 630      :--
038E 631
0004 038E 632      .ENTRY  ERF$TIMSTMP,^M<R2>          ; TIME STAMP ROUTINE
      51 10 9A 0390 633      MOVZBL #EMB$C_TS_LENGTH,R1      ; GET LENGTH OF MSG
00000000'EF 16 0393 634      JSB     ERL$ALCOCEMB      ; GO GET A MSG BLOCK
      OA 50 E9 0399 635      BLBC    R0,30$          ; BRANCH ON NO BLOCK
      04 A2 26 B0 039C 636      MOVW   #EMB$K_TS,EMB$W_HD_ENTRY(R2) ; SET ENTRY TYPE
00000000'EF 16 03A0 637      JSB     ERL$RELEASEMB      ; RELEASE BLOCK
      30$: 03A6 638      $SETIMR_S      ; SET A TIMER FOR TIME STAMPS
      03A6 639      DAYTIM=ERF$Q_DELTA,-      ; TIMER DELTA TIME
      03A6 640      ASTADR=ERF$TIMSTMP,-      ; THE TIME STAMP ENTRY
      03A6 641      REQIDT=#EMB$K_TS      ; AST PARAMETER IS MESSAGE TYPE
04 0388 642      RET
0389 643
0389 644      .DSABL  LSB

```

MOC

EVL
REC
EVL
EVL
EVL
ERF
COM
WOF
SYS
LIE

```

03B9 646 .SBTTL VOLUME MOUNT/DISMOUNT MESSAGE ROUTINE
03B9 647
03B9 648 :++
03B9 649 :
03B9 650 : ERF$MOUNT - MOUNT STATUS MESSAGE
03B9 651 :
03B9 652 : THIS ROUTINE IS PASSED THE ADDRESS OF THE MESSAGE. FROM THE MESSAGE, IT
03B9 653 : BUILDS A MESSAGE FOR THE OPERATOR INDICATING THE MOUNT STATUS (MOUNTED OR
03B9 654 : DISMOUNTED) AND SENDS IT TO THE APPROPRIATE OPERATOR. TAPE MOUNTS GO TO
03B9 655 : THE 'TAPES' OPERATOR, DISK MOUNTS GO TO THE 'DISKS' OPERATOR, AND ANY OTHER
03B9 656 : MESSAGES GO TO THE 'DEVICES' OPERATOR FOR HANDLING.
03B9 657 :
03B9 658 :--
03B9 659
077C 03B9 660 .ENTRY ERF$MOUNT,^M<R2,R3,R4,R5,R6,R8,R9,R10>
03BB 661 :
03BB 662 : DETERMINE IF A MESSAGE SHOULD BE SENT.
51 00000000'GF D0 03BB 663 : MOVL G^EXE$GL_MSGFLAGS,R1 ; GET SYSTEM MESSAGE FLAGS
40 8F 04 A7 91 03C2 664 : CMPB EMB$W_HD_ENTRY(R7),#EMB$C_VM ; IS THIS A MOUNT NOTIFICATION?
09 51 00000000'8F E0 03C7 665 : BNEQ 20$ ; BRANCH IF NOT
09 51 00000000'8F E0 03C9 666 : BBS #EXE$V_MOUNTMSG,R1,SNDRMSG ; BR IF MOUNT NOTIFICATION DESIRED
F7 51 00000000'8F E1 03D1 667 10$: RET ; OTHERWISE RETURN (NO STATUS)
03D2 668 20$: BBC #EXE$V_DISMOUMSG,R1,10$ ; BR IF DISMOUNT NOTIFICATION NOT DE
03DA 669
03DA 670 :
03DA 671 : BUILD A BUFFER DESCRIPTOR AND USE IT TO HOLD THE FORMATTED DEVICE NAME.
03DA 672 :
SE 1C C2 03DA 673 SNDRMSG: SUBL2 #28,SP ; MAKE ROOM FOR DESCRIPTOR AND BUFFER
56 SE D0 03DD 674 : MOVL SP,R6 ; COPY DESCRIPTOR ADDRESS
86 14 D0 03E0 675 : MOVL #20,(R6)+ ; SET DEVICE NAME BUFFER LENGTH
86 04 A6 DE 03E3 676 : MOVAL 4(R6),(R6)+ ; SET DEVICE NAME BUFFER ADDRESS
56 5E D0 03E7 677 : MOVL SP,R6 ; RESET DESCRIPTOR ADDRESS
58 1E A7 9E 03EA 678 : MOVAB ERF$B_VM_NAMLNG(R7),R8 ; GET ADDRESS OF DEVICE ASCII STRING
59 1C A7 3C 03EE 679 : MOVZWL ERF$W_VM_UNIT(R7),R9 ; GET DEVICE UNIT NUMBER
03F2 680 : $FAO_S DEVFAO,(R6),(R6),R8,R9 ; FORMAT THE DEVICE NAME
0407 681 :
0407 682 : BUILD A $GETDVI ITEM LIST ON THE STACK AND
0407 683 : CALL $GETDVI TO DETERMINE THE DEVICE CLASS.
0407 684 :
7E 7C 0407 685 : CLRQ -(SP) ; MAKE ROOM ON THE STACK FOR THE
7E 7C 0409 686 : CLRQ -(SP) ; $GETDVI ITEM LIST
68 00040004 8F D0 040B 687 : MOVL SP,R8 ; SAVE ADDRESS FOR LATER
04  A8 5E D0 040E 688 : MOVL #<DVI$_DEVCLASS@16>!4,(R8) ; SET ITEM CODE AND BUFFER SIZE
7E D4 0415 689 : CLRL -(SP) ; MAKE ROOM FOR THE DEVICE CLASS
0417 690 : MOVL SP,4(R8) ; NOTE THE STORAGE ADDRESS
041B 691 : ; NO RETURN LENGTH NEEDED
041B 692 : $GETDVI_S EFN=#6,- ; GET THE NEEDED DEVICE INFO
041B 693 : DEVNAM=(R6),-
041B 694 : ITMLST=(R8)
0431 695 : $WAITFR_S EFN=#6 ; WAIT UNTIL COMPLETE
50 0803 8F 3C 043A 696 : POPL R1 ; GET THE DEVICE CLASS
01 51 91 0442 697 : MOVZWL #<OPCSM_NM_DISKS@8>!OPCS RQ RQST,R0 ; SET FOR DISK OPERATOR
0F 13 0445 698 : CMPB R1,#DCS_DISK ; WAS IT A DISK DEVICE?
50 0403 8F 3C 0447 699 : BEQL 10$ ; XFER IF SO
02 51 91 044C 700 : MOVZWL #<OPCSM_NM_TAPES@8>!OPCS RQ RQST,R0 ; ELSE SET FOR TAPE
05 05 13 044F 701 : CMPB R1,#DCS_TAPE ; WAS IT A TAPE DEVICE?
05 05 13 044F 702 : BEQL 10$ ; XFER IF SO

```

VOLUME MOUNT/DISMOUNT MESSAGE ROUTINE

```

      50 1003 8F 3C 0451 703      MOVZWL #<OPCSM NM DEVICE@8>!OPCS_RQ RQST,R0 ; ELSE UNKNOWN.
000004EA'EF 50  DO 0456 704 10$:  MOVL   RO,MOUNT_MSG ; SET OPERATOR NAME
      045D 705 :
      045D 706 : FORMAT THE OPERATOR MESSAGE AND SEND IT TO OPCOM.
      045D 707 :
00000520'EF 0080 8F 80 045D 708      MOVW   #128,MOUNT_DSC ; RESET DESCRIPTOR SIZE
      56  DD 0466 709      PUSHL  R6 ; SET DEVICE NAME DESCRIPTOR
      000005AB'EF 9F 0468 710      PUSHAB MOUNT_MNT ; SET FOR MOUNT MESSAGE
40 8F 04 A7 91 046E 711      CMPB  EMB$W_HD_ENTRY(R7),#EMB$C-VM ; RIGHT?
      07 13 0473 712      BEQL  40$ ; XFER IF SO
6E 000005B1'EF 9E 0475 713      MOVAB  MOUNT_DMT,(SP) ; ELSE SET FOR DISMOUNT MESSAGE
      32 A7 9F 047C 714 40$:  PUSHAB ERFST_VM_LABEL(R7) ; ADDRESS OF VOLUME LABEL
      OC DD 047F 715      PUSHL  #12 ; SIZE OF THE LABEL
      0481 716      $FAO S MOUNT_FAO,MOUNT_DSC,MOUNT_DSC ; FORMAT THE MESSAGE
      049A 717      $SNDOPR_S MSGBUF=MOUNT_DSC ; SEND THE MESSAGE
      04 04AA 718      RET ; RETURN WHEN DONE

```

-\$2
Pse
SPL
SGL
SOW
SCO
MSG
MSC
MSC
MSC

\$\$TAB	= 0000067C	R	02	ERFSK_HD_LENGTH	00000010		
\$\$TABEND	= 000006DC	R	02	ERFSK_TS_LENGTH	00000010		
\$\$TMP	= 02000000			ERFSL_HD_SID	00000000		
\$\$TMP1	= 00000001			ERFSL_VM_ERRCNT	00000014		
\$\$TMP2	= 000000CF			ERFSL_VM_OPRCNT	00000018		
\$\$TMPX	= 00000000	R	03	ERFSL_VM_OWNUIC	00000010		
\$\$TMPX1	= 00000018			ERFSMOUNT	000003B9	RG	04
\$\$T1	= 00000000			ERFSM_HD_INVALID	= 0000C 0		
\$\$T2	= 00000003			ERFSQ_DELTA	00000008	R	04
..AFLG	= 00000000			ERFSQ_HD_TIME	00000006		
..FLG	= 00000000			ERFSQ_WAIT	00000010	R	04
..MOD	= 00000001			ERFSSNAPSHOT_COPIED	000005E8	RG	04
..N	= 00000001			ERFSSNAPSHOT_PRESENT	000005D4	RG	04
..TYP	= 00000003			ERF\$START	0000002F	RG	04
.LEN	= 00000001			ERF\$TIMSTMP	0000038E	RG	04
BYEMSG	00000462	R	02	ERF\$T_VM_LABEL	00000032		
BYEMSG_DSC	0000045A	R	02	ERF\$T_VM_NAMTX	0000001F		
BYEMSG_END	000004E2	R	02	ERFSW_HD_ENTRY	00000004		
BYEMSG_LEN	0000045A	R	02	ERFSW_HD_ERRSEQ	0000000E		
CONSC_INVSMP1	= 00000031			ERFSW_MBXCHN	000002FF	R	02
CONSC_INVSMP2	= 00000032			ERFSW_MBXSIZ	00000301	R	02
CONSC_REQERL	= 00000030			ERFSW_MBXUNT	00000303	R	02
CON\$SENDCONSCMD	*****	X	04	ERFSW_VM_NUMSET	0000C030		
DCS_DISK	= 00000001			ERFSW_VM_UNIT	0000001C		
DCS_TAPE	= 00000002			ERFSW_VM_VOLNUM	0000002E		
DEVFAO	00000305	R	02	ERL\$A\$CLOC\$EMB	*****	X	04
DIBSW_DEVBUSIZ	= 00000006			ERL\$A\$B\$UF\$ADDR	*****	X	04
DVIS_DEVCLASS	= 00000004			ERL\$B\$BUSY	= 00000000		
EMBSB_VALID	= FFFFFFFF			ERL\$B\$FLAGS	= 00000003		
EMBSK_HD_LENGTH	= 00000010			ERL\$B\$MSGCNT	= 00000001		
EMBSK_HLT	= 00000010			ERL\$C\$LENGTH	= 0000000C		
EMBSK_TS	= 00000026			ERL\$GB\$BUF\$PTR	*****	X	04
EMBSK_TS_LENGTH	= 00000010			ERL\$GL\$ERL\$PID	*****	X	04
EMBSK_VD	= 00000041			ERL\$END	= 00000008		
EMBSK_VM	= 00000040			ERL\$NEXT	= 00000004		
EMBSK_HD_LENGTH	= 00000010			ERL\$RELEASE\$EMB	*****	X	04
EMBSK_LENGTH	= 00000004			ERL\$V\$LOCK	= 00000000		
EMBSK_NF	= 00000023			ERM\$C\$FORMAT	= 00000002		
EMBSK_TS	= 00000026			ERRSNAP_COM	000007DB	R	02
EMBSL_HD_SID	= 00000000			ERRSNAP_CONCMD	00000841	R	02
EMBSQ_HD_TIME	= 00000006			ERRSNAP_DATA	00000842	R	02
EMBSW_HD_ENTRY	= 00000004			ERRSNAP_FAB	00000600	R	02
EMBSW_HD_ERRSEQ	= 0000000E			ERRSNAP_FLAGS	00000839	R	02
EMBSW_SIZE	= FFFFFFFC			ERRSNAP_LOG1	000007FB	R	02
ERFSB_ERRCNT	000005BD	R	02	ERRSNAP_LOG2	0000081A	R	02
ERFSB_HD_DCLASS	00000004			ERRSNAP_NAM	0000067C	R	02
ERFSB_HD_DTYPE	00000005			ERRSNAP_RSA	000006DC	R	02
ERFSB_MAXERRCNT	000005BE	R	02	ERRSNAP_STATUS	0000083D	R	02
ERFSB_VM_NAM\$NG	0000001E			ERRSNAP_XAB	00000650	R	02
ERFSC_HD_LENGTH	00000010			EXE\$GB\$CPUTYPE	*****	X	04
ERFSC_LOOP CNT	= 000000FF			EXE\$GL\$MSG\$FLAGS	*****	X	04
ERFSC_TS_LENGTH	00000010			EXE\$GQ\$ERL\$MBX	*****	X	04
ERF\$ERRSNAP	0000054A	RG	04	EXE\$IPID TO EPID	*****	X	04
ERF\$GETBUF	000004AB	RG	04	EXE\$V\$DISMOUNT\$MSG	*****	X	04
ERF\$INIT	00000352	RG	04	EXE\$V\$MOUNT\$MSG	*****	X	04
ERFSK_CLK TICK	= FF676980			FAB\$B\$FNS	= 00000034		
ERFSK_DLT\$A_STMP	= 00000258			FAB\$B\$ORG	= 0000001D		

Symt

SYS
SYS
WKQ
WKQ
WKQ
WKQ

ERRFMT
Symbol table

FABSB_RFM	=	0000001F			OUTFID	=	000002F4	R	02
FABSB_SHR	=	00000017			OUTNAM	=	00000018	R	04
FABSC_BID	=	00000003			OUTNAMSZ	=	00000017		
FABSC_BLN	=	00000050			OUTRAB	=	00000250	R	02
FABSC_SEQ	=	00000000			PCBSL_PID	=	00000060		
FABSC_VAR	=	00000002			PR\$ IPL	=	00000012		
FABSL_ALQ	=	00000010			PR\$ SID	=	0000003E		
FABSL_FNA	=	0000002C			PR\$ SID_TYP 790	=	00000004		
FABSL_FOP	=	00000004			PRCBUF	=	0000004C	R	04
FABSV_CHAN_MODE	=	00000002			PRCNXT	=	00000070	R	04
FABSV_CIF	=	00000019			PRCNXT1	=	00000072	R	04
FABSV_FILE_MODE	=	00000004			ROMSG	=	00000356	R	02
FABSV_GET	=	00000001			ROMSG_DSC	=	0000031D	R	02
FABSV_LNM_MODE	=	00000000			ROMSG_END	=	00000456	R	02
FABSV_PUT	=	00000000			ROMSG_LEN	=	00000456	R	02
FABSV_UPD	=	00000003			RAB\$B_RAC	=	0000001E		
FABSV_UPI	=	00000006			RAB\$C_BID	=	00000001		
FABSW_GBC	=	00000048			RAB\$C_BLN	=	00000044		
FABSW_IFI	=	00000002			RAB\$C_RFA	=	00000002		
FABSW_MRS	=	00000036			RAB\$C_SEQ	=	00000000		
FILCRE	=	00000000	R	04	RAB\$C_CTX	=	00000018		
INBUF	=	00000000	R	02	RAB\$C_RBF	=	00000028		
IOSM_NOW	*****		X	04	RAB\$C_RQP	=	00000004		
IOS WRITEVBLK	*****		X	04	RAB\$V_EOF	=	00000008		
LASTENTRY	=	000002FA	R	02	RAB\$W_BWH	=	0000000A		
LIB\$SPAWN	*****		X	04	RAB\$W_ISI	=	00000002		
MBX	=	00000239	R	04	RAB\$W_RSZ	=	00000022		
MOUNT_BUF	=	00000528	R	02	RETURN STATUS	=	000005D3	R	04
MOUNT_DMT	=	000005B1	R	02	SCH\$GL_CURPCB	*****		X	04
MOUNT_DSC	=	00000520	R	02	SID	=	000002FB	R	02
MOUNT_END	=	00000520	R	02	SNDMSG	=	000003DA	R	04
MOUNT_FAO	=	000004E2	R	02	SS\$ NORMAL	=	00000001		
MOUNT_MNT	=	000005A8	R	02	SY\$S ASSIGN	*****		GX	04
MOUNT_MSG	=	000004EA	R	02	SY\$S CLOSE	*****		GX	04
MSGOUT	=	00000162	R	04	SY\$S CMKRN	*****		GX	04
NAM\$B_ESS	=	0000000A			SY\$S CONNECT	*****		GX	04
NAM\$B_NOP	=	00000008			SY\$S CREATE	*****		GX	04
NAM\$B_RSL	=	00000003			SY\$S DASSGN	*****		GX	04
NAM\$B_RSS	=	00000002			SY\$S DELPRC	*****		GX	04
NAM\$C_BID	=	00000002			SY\$S FAO	*****		X	04
NAM\$C_BLN	=	00000060			SY\$S FIND	*****		GX	04
NAM\$C_MAXRSS	=	000000FF			SY\$S GETCHN	*****		GX	04
NAM\$S_ESA	=	0000000C			SY\$S GETDVI	*****		GX	04
NAM\$S_RSA	=	00000004			SY\$S GETMSG	*****		GX	04
NAM\$W_FID	=	00000024			SY\$S HIBER	*****		GX	04
NAMELOCK	=	00000294	R	02	SY\$S OPEN	*****		GX	04
NXTMSG	=	00000139	R	04	SY\$S PUT	*****		GX	04
OPCSM_NM_CENTRL	=	00000001			SY\$S QIO	*****		GX	04
OPCSM_NM_DEVICE	=	00000010			SY\$S SETIMR	*****		GX	04
OPCSM_NM_DISKS	=	00000008			SY\$S SNDOPR	*****		GX	04
OPCSM_NM_TAPES	=	00000004			SY\$S UPDATE	*****		GX	04
OPCS RQ_RQST	=	00000003			SY\$S WAITFR	*****		GX	04
OPRMSG	=	00000325	R	02	WRITE_FAILURE	=	000001BB	R	04
OPRMSG_DSC	=	00000315	R	02	XAB\$C_DAT	=	00000012		
OPRMSG_END	=	00000356	R	02	XAB\$C_DATLEN	=	0000002C		
OPRMSG_LEN	=	00000315	R	02	XAB\$C_NXT	=	00000004		
OUTFAB	=	00000200	R	02	XAB\$C_CDT	=	00000014		

Valu

0000

0000

0000

0000

0000

0000

0000

0000

0000

0000

0000

0000

0000

0000

0000

0000

0000

0000

0000

0000

0000

0000

0000

0000

0200

0200

0200

0200

0200

0200

0200

0200

0200

0200

