

[illegible][illegible]

```

EEEEEEEEEE RRRRRRRR LL LL 000000 GGGGGGGG MM MM SSSSSSSS GGGGGGGG
EEEEEEEEEE RRRRRRRR LL LL 000000 GGGGGGGG MM MM SSSSSSSS GGGGGGGG
EE          RR          RR LL LL 00 00 GG          MMMM MMMM SS          GG
EE          RR          RR LL LL 00 00 GG          MMMM MMMM SS          GG
EE          RR          RR LL LL 00 00 GG          MM MM MM SS          GG
EEEEEEEEEE RRRRRRRR LL LL 00 00 GG          MM MM MM SS          GG
EEEEEEEEEE RRRRRRRR LL LL 00 00 GG          MM MM MM SS          GG
EE          RR RR LL LL 00 00 GG          MM MM MM SS          GG
EE          RR RR LL LL 00 00 GG          MM MM MM SS          GG
EE          RR RR LL LL 00 00 GG          MM MM MM SS          GG
EEEEEEEEEE RR          RR LL LLLLLLLLLL LL LLLLLLLLLL 000000 GGGGGG MM MM SSSSSSSS GG GGGGGG
EEEEEEEEEE RR          RR LL LLLLLLLLLL LL LLLLLLLLLL 000000 GGGGGG MM MM SSSSSSSS GG GGGGGG
                                     ....
                                     ....
                                     ....
                                     ....

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

```

J 5
16-Sep-1984 00:21:37
5-Sep-1984 13:56:07

VAX-11 FORTRAN V3.4-56
DISK\$VMSMASTER:[ERF.SRC]ERLLOGMSG.FOR;1

Page 1

```
0001 C
0002 C Version: 'V04-000'
0003 C
0004 C*****
0005 C*
0006 C* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0007 C* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0008 C* ALL RIGHTS RESERVED.
0009 C*
0010 C* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0011 C* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0012 C* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0013 C* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0014 C* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0015 C* TRANSFERRED.
0016 C*
0017 C* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0018 C* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0019 C* CORPORATION.
0020 C*
0021 C* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0022 C* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0023 C*
0024 C*
0025 C*****
0026 C
0027
0028 c Author Brian Porter Creation date 07-FEB-1982
0029
0030 c++
0031 c Functional description
0032 c
0033 c This module provides dispatching for entries logged by erl$logmessage.
0034 c Dispatching is done by emb$w_lm_msgtyp.
0035 c
0036 c Modified by:
0037 c
0038 c V03-009 EAD0196 Elliott A. Drayton 6-Jul-1984
0039 c Changed LUN, OPTION and RECCNT to now come from SYECOM.
0040 c
0041 c
0042 c V03-008 SAR0271 Sharon A. Reynolds 18-Jun-1984
0043 c - Added TMSCP support.
0044 c
0045 c V03-007 SAR0196 Sharon A. Reynolds, 20-Feb-1984
0046 c Added an SYE update that:
0047 c - Added code to interrogate the 'mscp command reference'
0048 c numbers. If zero output the entry immediately.
0049 c - Fixed a bug in the output of mscp entries when
0050 c error logging mailbox selected as output.
0051 c
0052 c V03-006 SAR0134 Sharon A. Reynolds, 9-Sep-1983
0053 c Added changes that were made to SYE (erllogmsg) to
0054 c remove the mscp 'first part' info msg and fix a bug
0055 c relating to summary reports.
0056 c
0057 c V03-005 SAR0073 Sharon A. Reynolds, 20-Jun-1983
```

K 5
16-Sep-1984 00:21:37
5-Sep-1984 13:56:07

VAX-11 FORTRAN V3.4-56 Page 2
DISK\$VMSMASTER:[ERF.SRC]ERLLOGMSG.FOR;1

```
0058 C          Changed the carriage control in the 'format' statements
0059 C          for use with ERF.
0060 C
0061 C          v03-004 BP0004          Brian Porter,          08-FEB-1983
0062 C          Corrected argument list to pudriver.
0063 C
0064 C          v03-003 BP0003          Brian Porter,          18-JUN-1982
0065 C          Added ci780 support.
0066 C
0067 C          v03-002 BP0002          Brian Porter,          25-MAY-1982
0068 C          Added 'option' filtering.
0069 C
0070 C          v03-001 BP0001          Brian Porter,          14-APR-1982
0071 C          Added more routines.
0072 C--
0073 C**
0074
0075 Subroutine ERL$LOGMESSAGE_DISPATCHER (lun,record_length,reccnts,option)
0076
0077 include 'src$:msghdr.for /nolist'
0137 include 'src$:emblmdef.for /nolist'
0206 include 'src$:syecom.for /nolist'
0334
0335 byte          lun
0336 integer*4     record_length
0337 integer*4     reccnts
0338 character*1   option
0339
0340 byte          mount_flag_and_label_array(16)
0341
0342 integer*4     volume_mount_flag
0343 integer*4     current_volume_label
0344
0345 equivalence   (mount_flag_and_label_array(1),volume_mount_flag)
0346 equivalence   (mount_flag_and_label_array(5),current_volume_label)
0347
0348
0349 C
0350 C Via the message type, determine which module to call to decode/output
0351 C the MSCP entry.
0352 C
0353 C   if (
0354 C   1 emb$w_lm_msgtyp .eq. 1          ! Disk message
0355 C   1 .OR.
0356 C   1 emb$w_lm_msgtyp .EQ. 2          ! Tape message
0357 C   1 ) then
0358 C
0359 C   if (option .eq. 'B') then          ! Brief report
0360 C
0361 C Output the entry header.
0362 C
0363 C Call HEADER (lstlun)
0364 C Call LOGGER (lstlun,'ERL$LOGMESSAGE ENTRY')
0365 C Call DHEAD2 (lstlun,'I/O',emb$b_lm_namlng,emb$t_lm_name,emb$w_lm_unit)
0366
0367 C Else if (option .eq. 'S') then      ! Full report
```

```
0368
0369 C
0370 C Determine if output is going to the mailbox or if there is no message
0371 C text logged in this packet, if so output the entry immediately.
0372 C
0373     If (
0374     1 mailbox_channel .NE. 0
0375     1 .OR.
0376     1 LIB$EXTZV(0,31,emb$b_lm_msgtxt) .EQ. 0
0377     1 ) then
0378
0379     Call INTERVENE_DECREMENT (lstlun)
0380
0381     Volume_mount_flag = -1
0382     Call GET_CURRENT_LABEL (3,emb$l_hd_sid,emb$b_lm_namlng,emb$t_lm_name,
0383     1 emb$w_lm_unit,%REF(current_volume_label), #5)
0384
0385     Volume_mount_flag = %LOC(current_volume_label)
0386 S    Continue
0387
0388     Call DISK_TAPE_DRV_R_MSCP_DISPATCHER (lstlun,option,recnt,
0389     1 mount_flag_and_label_array,record_size, (1))
0390
0391     Else
0392 C
0393 C Save the entry so it can be output with the software status information
0394 C that is logged in a seperate packet.
0395 C
0396     Call DISK_TAPE_DRIVERS_MSCP_Q (record_size,recnt,emb$b_lm_msgtxt)
0397
0398     Endif
0399     Endif
0400
0401     else if (emb$w_lm_msgtyp .eq. 3) then    ! CI Port message
0402
0403     if (
0404     1 option .eq. 'S'
0405     1 .or.
0406     1 option .eq. 'B'
0407     1 ) then
0408
0409     Call INTERVENE_DECREMENT (lstlun)
0410     Call PADRIVER_LOGMESSAGE (lstlun,option)
0411     endif
0412
0413     else if (emb$w_lm_msgtyp .eq. 4) then    ! UDA Port message
0414
0415     if (
0416     1 option .eq. 'S'
0417     1 .or.
0418     1 option .eq. 'B'
0419     1 ) then
0420
0421     Call INTERVENE_DECREMENT (lstlun)
0422     Call PADRIVER_MSCP_DISPATCHER (lstlun,option,recnt,record_size)
0423
0424     Endif
```

```
0425
0426     else if (emb$w_lm_msgtyp .eq. 5) then      ! Available Attention Msg
0427
0428         if (
0429             1 option .eq. 'S'
0430             1 .or.
0431             1 option .eq. 'B'
0432             1 ) then
0433
0434             Call INTERVENE_DECREMENT (lstlun)
0435
0436             If (option .EQ. 'S') then
0437                 Call FRCTOF (lstlun)
0438             Endif
0439
0440             Call HEADER (lstlun)
0441             Call LOGGER (lstlun,'ERL$LOGMESSAGE ENTRY')
0442             Call DHEAD2 (lstlun,'I/O',emb$b_lm_naming,emb$t_lm_name,emb$w_lm_unit)
0443
0444             Call LINCHK (lstlun,2)
0445             write(lstlun,10) "'AVAILABLE" MSCP ATTENTION MESSAGE'
0446             format(/' ',a)
0447             endif
0448
0449             if (option .eq. 'S') then
0450                 Call MSCP$K_OP_AVATN (lstlun,.true.)
0451             endif
0452
0453     else if (emb$w_lm_msgtyp .eq. 6) then      ! Duplicate Unit Msg
0454
0455         if (
0456             1 option .eq. 'S'
0457             1 .or.
0458             1 option .eq. 'B'
0459             1 ) then
0460
0461             Call INTERVENE_DECREMENT (lstlun)
0462
0463             If (option .EQ. 'S') then
0464                 Call FRCTOF (lstlun)
0465             Endif
0466
0467             Call HEADER (lstlun)
0468             Call LOGGER (lstlun,'ERL$LOGMESSAGE ENTRY')
0469             Call DHEAD2 (lstlun,'I/O',emb$b_lm_naming,emb$t_lm_name,emb$w_lm_unit)
0470
0471             Call LINCHK (lstlun,2)
0472             write(lstlun,10) "'DUPLICATE UNIT NUMBER" MSCP ATTENTION MESSAGE'
0473             endif
0474
0475             if (option .eq. 'S') then
0476                 Call MSCP$K_OP_DUPUN (lstlun,.true.)
0477             endif
0478
0479     else if (emb$w_lm_msgtyp .eq. 7) then      ! Invalid Command Msg
0480
0481         if (
```

```
0482      1 option .eq. 'S'
0483      1 .or.
0484      1 option .eq. 'B'
0485      1 ) then
0486
0487      Call INTERVENE_DECREMENT (lstlun)
0488
0489      If (option .EQ. 'S') then
0490      Call FRCTOF (lstlun)
0491      Endif
0492
0493      Call HEADER (lstlun)
0494      Call LOGGER (lstlun,'ERL$LOGMESSAGE ENTRY')
0495      Call DHEAD2 (lstlun,'I/O',emb$b_lm_namlng,emb$t_lm_name,emb$w_lm_unit)
0496
0497      Call LINCHK (lstlun,2)
0498      Write (lstlun,10) "'INVALID COMMAND'" MSCP END MESSAGE'
0499      endif
0500
0501      if (option .eq. 'S') then
0502      Call MSCP_INVALID_COMMAND_END_MESSAGE (lstlun)
0503      endif
0504
0505      else if (emb$w_lm_msgtyp .eq. 8) then ! Access Path Attention Msg
0506
0507      if (
0508      1 option .eq. 'S'
0509      1 .or.
0510      1 option .eq. 'B'
0511      1 ) then
0512
0513      Call INTERVENE_DECREMENT (lstlun)
0514
0515      If (option .EQ. 'S') then
0516      Call FRCTOF (lstlun)
0517      Endif
0518
0519      Call HEADER (lstlun)
0520      Call LOGGER (lstlun,'ERL$LOGMESSAGE ENTRY')
0521      Call DHEAD2 (lstlun,'I/O',emb$b_lm_namlng,emb$t_lm_name,emb$w_lm_unit)
0522
0523      Call LINCHK (lstlun,2)
0524      write(lstlun,10) "'ACCESS PATH'" MSCP ATTENTION MESSAGE'
0525      endif
0526
0527      if (option .eq. 'S') then
0528      Call MSCP$$K_OP_ACPH (lstlun,.true.)
0529      endif
0530
0531      c
0532      c      The IF-THEN-ELSE should be expanded at this point to add
0533      c      new entry sub codes.
0534      c
0535
0536      else
0537
0538      c
```

```

0539      C Unknown message type, call the applicable routines to decode/output the
0540      C header and output the remainder of the packet in a hex longword format.
0541      C
0542          if (
0543              1 option .eq. 'S'
0544              1 .or.
0545              1 option .eq. 'B'
0546              1 ) then
0547              Call ERLLOGMSG (lstlun,record_size)
0548          endif
0549      endif
0550
0551      return
0552
0553      end
0554

```

PROGRAM SECTIONS

Name	Bytes	Attributes
0 \$CODE	820	PIC CON REL LCL SHR EXE RD NOWRT LONG
1 \$PDATA	201	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2 \$LOCAL	424	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
3 EMB	512	PIC OVR REL GBL SHR NOEXE RD WRT LONG
4 SYECOM	44	PIC OVR REL GBL SHR NOEXE RD WRT LONG
Total Space Allocated	2001	

ENTRY POINTS

Address	Type	Name
0-00000000		ERL\$LOGMESSAGE_DISPATCHER

VARIABLES

Address	Type	Name	Address	Type	Name
4-00000012	L*1	CP_11750	4-00000011	L*1	CP_11780
4-00000013	L*1	CP_11722	4-00000014	L*4	CRYPTK_FLAG
2-00000004	I*4	CURRENT_VOLUME_LABEL	4-0000000D	I*4	DEV_CHAR
3-00000010	L*1	EMB\$B_LM_CLASS	3-00000014	L*1	EMB\$B_LM_NAMLNG
3-00000011	L*1	EMB\$B_LM_TYPE	3-00000000	I*4	EMB\$B_LM_SID
3-00000015	CHAR	EMB\$T_LM_NAME	3-00000004	I*2	EMB\$W_HD_ENTRY
3-0000000E	I*2	EMB\$W_HD_ERRSEQ	3-00000024	I*2	EMB\$W_LM_MSGTYP
3-00000012	I*2	EMB\$W_LM_UNIT	4-0000001E	L*1	END_VALUE
4-0000001D	L*1	EOF_FLAG	4-00000004	L*4	FORMS
4-0000000C	L*1	LINES	4-00000027	I*4	LSTLUN
AP-00000004a	L*1	LUN	4-0000001F	I*4	MAILBOX_CHANNEL
AP-00000010a	CHAR	OPTION	4-0000002B	CHAR	OPTIONS
4-00000008	L*4	PRINTER	4-00000000	I*4	RECCNT

AP-0000000C@ I*4 RECCNTS
4-00000023 I*4 RECORD_SIZE
4-0000001A L*1 VALID_CPU
4-0000001C L*1 VALID_TYPE
4-00000018 L*1 VOLUME_OUTPUT

AP-00000008@ I*4 RECORD_LENGTH
4-00000019 L*1 VALID_CLASS
4-0000001B L*1 VALID_ENTRY
2-00000000 I*4 VOLUME_MOUNT_FLAG

ARRAYS

Address	Type	Name	Bytes	Dimensions
3-00000000	L*1	EMB	512	(0:511)
3-00000026	L*1	EMBSB_LM_MSGTXT	460	(460)
3-00000006	I*4	EMBSQ_HD_TIME	8	(2)
2-00000000	L*1	MOUNT_FLAG_AND_LABEL_ARRAY	16	(16)

LABELS

Address	Label	Address	Label
0-0000007E	5	1-000000C3	10'

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name	Type	Name
	DHEAD2		DISK_TAPE_DRIVERS_MSCP_Q		DISK_TAPE_DRV_MSCP_DISPATCHER
	ERLLOGMSG		FRCTOF		GET_CURRENT_LABEL
	HEADER		INTERVENE_DECREMENT	I*4	LIB\$EXTZV
	LINCHK		LOGGER		MSCP\$\$K_OP_ACPH
	MSCP\$\$K_OP_AVATN		MSCP\$\$K_OP_DUPUN		MSCP_INVALID_COMMAND_END_MESSAGE
	PADRIVER_LOGMESSAGE		PUDRIVER_MSCP_DISPATCHER		

Subroutine ERLLOGMSG (lun,record_length)

include 'src\$:msghdr.for /nolist'
include 'src\$:emblmdef.for /nolist'
include 'src\$:syecom.for /nolist'

byte lun

integer*4 record_length
integer*4 compress4

Call FRCTOF (lstlun)
Call HEADER (lstlun)
Call LOGGER (lstlun,'ERL\$LOGMESSAGE ENTRY')
Call DHEAD2 (lstlun,'I/O',emb\$b_lm_namlng,emb\$t_lm_name,emb\$w_lm_unit)

Entry ERLLOGMSG2 (lun,record_length)

Call LINCHK (lstlun,3)

write(lstlun,90) 'EMB\$W_LM_MSGTYP',(emb\$w_lm_msgtyp,i = 1,2)
format(/' ',t8,a,t28,z4.4,7,
1 t40,a,i<compress4 (lib\$extzv(0,16,emb\$w_lm_msgtyp)))>>

Call DUMPREG (lstlun,((record_size + 3)/4) - 9,emb(38))

return

end

0001
0002
0003
0004
0005
0006
0007
0066
0135
0263
0264
0265
0266
0267
0268
0269
0270
0271
0272
0273
0274
0275
0276
0277
0278
0279
0280
0281
0282
0283
0284
0285
0286
0287
0288
0289
0290

90

PROGRAM SECTIONS

Name	Bytes	Attributes
0 \$CODE	181	PIC CON REL LCL SHR EXE RD NOWRT LONG
1 \$PDATA	75	PIC CCN REL LCL SHR NOEXE RD NOWRT LONG
2 \$LOCAL	132	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
3 EMB	512	PIC OVR REL GBL SHR NOEXE RD WRT LONG
4 SYECOM	44	PIC OVR REL GBL SHR NOEXE RD WRT LONG
Total Space Allocated	944	

ENTRY POINTS

Address	Type	Name	Address	Type	Name
0-00000000		ERLLOGMSG	0-0000002B		ERLLOGMSG2

VARIABLES

Address	Type	Name	Address	Type	Name
4-00000012	L*1	CP_11750	4-00000011	L*1	CP_11780
4-00000013	L*1	CP_11722	4-00000014	L*4	CRYPTK_FLAG
4-0000000D	I*4	DEV_CHAR	3-00000010	L*1	EMBSB_LM_CLASS
3-00000014	L*1	EMBSB_LM_NAMLNG	3-00000011	L*1	EMBSB_LM_TYPE
3-00000000	I*4	EMBSL_HD_SID	3-00000015	CHAR	EMBST_LM_NAME
3-00000004	I*2	EMBSW_HD_ENTRY	3-0000000E	I*2	EMBSW_HD_ERRSEQ
3-00000024	I*2	EMBSW_LM_MSGTYP	3-00000012	I*2	EMBSW_LM_UNIT
4-0000001E	L*1	END_VALUE	4-0000001D	L*1	EOF_FLAG
4-00000004	L*4	FORMS	2-00000000	I*4	I
4-0000000C	L*1	LINES	4-00000027	I*4	LSTLUN
AP-00000004a	L*1	LUN	4-0000001F	I*4	MAILBOX_CHANNEL
4-0000002B	CHAR	OPTIONS	4-00000008	L*4	PRINTER
4-00000000	I*4	RECCNT	AP-00000008a	I*4	RECORD_LENGTH
4-00000023	I*4	RECORD_SIZE	4-00000019	L*1	VALID_CLASS
4-0000001A	L*1	VALID_CPU	4-0000001B	L*1	VALID_ENTRY
4-0000001C	L*1	VALID_TYPE	4-00000018	L*1	VOLUME_OUTPUT

ARRAYS

Address	Type	Name	Bytes	Dimensions
3-00000000	L*1	EMB	512	(0:511)
3-00000026	L*1	EMBSB_LM_MSGTXT	460	(460)
3-00000006	I*4	EMBSQ_HD_TIME	8	(2)

LABELS

Address	Label
1-00000034	90'

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name	Type	Name	Type	Name	Type	Name
I*4	COMPRESS4 LINCHK		DHEAD2 LOGGER		DUMPREG		FRCTOF		HEADER
								I*4	LIB\$EXTZV

COMMAND QUALIFIERS

FORTRAN /LIS=LIS\$:ERLLOGMSG/OBJ=OBJ\$:ERLLOGMSG MSRC\$:ERLLOGMSG

/CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)

/DEBUG=(NOSYMBOLS,TRACEBACK)

/STANDARD=(NOSYNTAX,NOSOURCE FORM)

/SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP)

/F77 /NOG_FLOATING /I4 /OPTIMIZE /WARNINGS /NOD_LINES /NOCROSS_REFERENCE /NOMACHINE_CODE /CONTINUATIONS=19

COMPILATION STATISTICS

Run Time:	5.49 seconds
Elapsed Time:	16.60 seconds
Page Faults:	160
Dynamic Memory:	192 pages

0149 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

