







```
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$LOGMESSAGE_DISPATCHER (lun,record_length,reccnts,option)
0076
0077
0078 include 'src$:msghdr.for /nolist'
0137 include 'src$:emblmdef.for /nolist'
0206 include 'src$:syecom.for /nolist'
0334
0335 byte          lun
J336 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 C
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_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     Endif
0424
```



```

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_namng,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$b_lm_namng,emb$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_ACPTH (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
0548     Call ERLLOGMSG (lstlun,record_size)
0549     endif
0550     endif
0551
0552     return
0553
0554     end
    
```

PROGRAM SECTIONS

Name	Bytes	Attributes
0 \$CODE	820	PIC CON REL LCL SHR EXE RD NOWRT LCNG
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_1172Z	4-00000014	L*4	CRYPTK_FLAG
2-00000004	I*4	CURRENT_VOLUME_LABEL	4-00000000	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
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_DRVR_MSCP_DISPATCHER
	ERLLOGMSG		FRCTDF		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		PADRIVER_MSCP_DISPATCHER		

```
0001
0002
0003
0004 Subroutine ERLLOGMSG (lun,record_length)
0005
0006
0007 include 'src$:msghdr.for /nolist'
0066 include 'src$:emblmdef.for /nolist'
0135 include 'src$:syecom.for /nolist'
0263
0264 byte lun
0265
0266 integer*4 record_length
0267 integer*4 compress4
0268
0269
0270 Call FRCTOF (lstlun)
0271 Call HEADER (lstlun)
0272 Call LOGGER (lstlun,'ERL$LOGMESSAGE ENTRY')
0273 Call DHEAD2 (lstlun,'I/O',emb$b_lm_namlng,emb$t_lm_name,emb$w_lm_unit)
0274
0275
0276
0277
0278 Entry ERLLOGMSG2 (lun,record_length)
0279
0280 Call LINCHK (lstlun,3)
0281
0282 write(lstlun,90) 'EMBSW_LM_MSGTYP',(emb$w_lm_msgtyp,i = 1,2)
0283 90 format(/' ',t8,a,t28,z4:4,7,
0284 1 t40,a,i<compress4 (lib$extzv(0,16,emb$w_lm_msgtyp))>>)
0285
0286 Call DUMPREG (lstlun,((record_size + 3)/4) - 9,emb(38))
0287
0288 return
0289
0290 end
```

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	4-00000018	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



