


```

EEEEEEEEEE RRRRRRRR ' LL 000000 GGGGGGGG SSSSSSSS TTTTTTTTTT SSSSSSSS
EEEEEEEEEE RRRRRRRR LL 000000 GGGGGGGG SSSSSSSS TTTTTTTTTT SSSSSSSS
EE RR RR LL LL 00 00 GG SS TT SS
EE RR RR LL LL 00 00 GG SS TT SS
EE RR RR LL LL 00 00 GG SS TT SS
EE RR RR LL LL 00 00 GG SS TT SS
EEEEEEEEEE RRRRRRRR LL LL 00 00 GG SSSSSS TT SSSSSS
EEEEEEEEEE RRRRRRRR LL LL 00 00 GG SSSSSS TT SSSSSS
EE RR RR LL LL 00 00 GG GGGGGG SS TT SS
EE RR RR LL LL 00 00 GG GGGGGG SS TT SS
EE RR RR LL LL 00 00 GG GG SS TT SS
EEEEEEEEEE RR RR LLLLLLLLLL LLLLLLLLLL 000000 000000 GGGGGG SSSSSSSS T SSSSSSSS
EEEEEEEEEE RR RR LLLLLLLLLL LLLLLLLLLL 000000 000000 GGGGGG SSSSSSSS TT SSSSSSSS

```

```

LL I11111 SSSSSSSS
LL I11111 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 I11111 SSSSSSSS
LLLLLLLLLL I11111 SSSSSSSS

```

```

....
....
....
....

```

```
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 C
0028 c Author Brian Porter Creation date 07-FEB-1982
0029 c
0030 c++
0031 c Functional description
0032 c
0033 c This module provides dispatching for entries logged by erl$logstatus.
0034 c
0035 c Modified by:
0036 c
0037 c V03-010 EAD184 Elliott A. Drayton 6-Jul-1984
0038 c Add page break to begin intervening entry reports.
0039 c
0040 c V03-009 SAR270 Sharon A. Reynolds 18-Jun-1984
0041 c - Added TMSCP support.
0042 c
0043 c V03-008 SAR0259 Sharon A. Reynolds 26-Apr-1984
0044 c - TUB1 partial entry fix.
0045 c
0046 c V03-007 SAR0222 Sharon A. Reynolds, 28-Mar-1984
0047 c Changed the call to UCBSL_OWNUIC to ORBSL_OWNER.
0048 c
0049 c V03-006 SAR0195 Sharon A. Reynolds, 20-Feb-1984
0050 c Added an SYE update that:
0051 c - Added code to interrogate the 'mscp command reference'
0052 c numbers. If zero output the entry immediately.
0053 c - Fixed a bug in the output of the mscp entries when the
0054 c error log mailbox is selected as for output.
0055 c
0056 c V03-005 SAR0133 Sharon A.Reynolds, 9-Sep-1983
0057 c Added fixes that were made to SYE (erllogsts) that
```

```
0058 C removed the mscp 'first part' info message and fixed
0059 C bug relating to summary reports.
0060 C
0061 C V03-004 SAR0074 Sharon A. Reynolds, 20-Jun-1983
0062 C Changed the carriage control in the 'format' statements
0063 C for use with ERF.
0064 C
0065 C V03-003 SAR0026 Sharon A. Reynolds, 16-May-1983
0066 C Made SYECOM available and added code to check for eof
0067 C flag and call 'dudriver_mscp_dq'. Also changed name of
0068 C 'recnt' input parameter due to conflict with SYECOM
0069 C recnt.
0070 C
0071 C v03-002 BP0001 Brian Porter, 21-OCT-1982
0072 C Added ra60.
0073 C
0074 C v03-001 BP0001 Brian Porter, 18-APR-1982
0075 C Added brief.
0076 C--
0077 C**
0078
0079 Subroutine ERL$LOGSTATUS_DISPATCHER (lun,record_length,record_number,
0080 1 option)
0081
0082
0083 include 'src$:msghdr.for /nolist'
0142 include 'src$:embspdef.for /nolist'
0255 Include 'src$:syecom.for /nolist'
0383
0384
0385 byte lun
0386 integer*4 record_length
0387 integer*4 record_number
0388 character*1 option
0389
0390 byte mount_flag_and_label_array(16)
0391
0392 integer*4 volume_mount_flag
0393 integer*4 current_volume_label
0394
0395 equivalence (mount_flag_and_label_array(1),volume_mount_flag)
0396 equivalence (mount_flag_and_label_array(5),current_volume_label)
0397
0398
0399 C
0400 C Determine if the entire file has been processed, if so, call
0401 C a routine that will de-queue and output the remainder of the
0402 C disk and tape MSCP messages and return to the calling routine.
0403 C
0404 C If (EOF FLAG) then
0405 C Call DISK_TAPE_DRIVERS_MSCP_DQ (lstlun,options) ! DU and TU drivers
0406 C Return
0407 C Endif
0408
0409 C if (options .eq. 'B') then
0410 C Call HEADER (lstlun)
0411
```

```
0412 Call LOGGER (lstlun,'ERL$LOGSTATUS ENTRY')
0413 Call DHEAD2 (lstlun,'I/O'.emb$b_sp_namlng,emb$t_sp_name,emb$w_sp_unit)
0414 endif
0415
0416 if (emb$b_sp_class .eq. 128) then
0417
0418 if (options .eq. 'S') then
0419
0420 Call PUDRIVER_MSCP_DISPATCHER (lstlun,options,record_number)
0421 endif
0422
0423 else if (
0424 1 emb$b_sp_class .eq. 1 ! Disk messages
0425 1 .OR.
0426 1 emb$b_sp_class .EQ. 2 ! Tape messages
0427 1 ) then
0428
0429 if (options .eq. 'S') then
0430
0431 C
0432 C Determine if output is directed to the report generator mailbox or if the
0433 C command reference number is equal to 0 (invalid cmd?). If so, output the
0434 C entry immediately.
0435 C
0436 If (
0437 1 mailbox_channel .NE. 0
0438 1 .OR.
0439 1 emb$l_sp_cmdref .EQ. 0
0440 1 ) then
0441
0442 Call INTERVENE_DECREMENT (lstlun)
0443
0444 Volume_mount_flag = -1
0445 Call GET_CURRENT_LABEL (3,emb$l_hd_sid,emb$b_sp_namlng,emb$t_sp_name,
0446 1 emb$w_sp_unit,%REF(current_volume_label),*5)
0447
0448 Volume_mount_flag = %LOC(current_volume_label)
0449 S Continue
0450
0451 Call DISK_TAPE_DRVR_MSCP_DISPATCHER (lstlun,options,record_number,
0452 1 mount_flag_and_label_array,record_length,(1))
0453
0454 Else
0455 C
0456 C The command is valid and output is not going to the mailbox, save the
0457 C entry so that it can be output together with the device hardware status
0458 C packet that is logged seperately.
0459 C
0460 Call DISK_TAPE_DRIVERS_MSCP_Q (record_length,record_number,
0461 1 emb$l_sp_cmdref)
0462
0463 Endif
0464 Endif
0465
0466 C
0467 C Unknown device type, call a routine that will call applicable
0468 C routines that will decode/output the entry. As new device types
```

```

0469 C are defined the IF-THEN-ELSE should be expanded at this point to
0470 C support them.
0471 C
0472     else
0473     call ERLLOGSTS (lstlun)
0474     endif
0475
0476     return
0477     end
    
```

PROGRAM SECTIONS

Name	Bytes	Attributes
0 \$CODE	248	PIC CON REL LCL SHR EXE RD NOWRT LONG
1 \$PDATA	28	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2 \$LOCAL	200	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		1032

ENTRY POINTS

Address	Type	Name
0-00000300		ERL\$LOGSTATUS_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_117ZZ	4-00000014	L*4	CRYPTK_FLAG
2-00000004	I*4	CURRENT_VOLUME_LABEL	4-0000000D	I*4	DEV_CHAR
3-00000010	L*1	EMBSB_SP_CLASS	3-00000040	L*1	EMBSB_SP_NAMLNG
3-00000011	L*1	EMBSB_SP_TYPE	3-00000000	I*4	EMBSL_HD_SID
3-00000014	I*4	EMBSL_SP_BCNT	3-00000038	I*4	EMBSL_SP_CHAR
3-0000003C	I*4	EMBSL_SP_CMDREF	3-00000020	I*4	EMBSL_SP_IOSB1
3-00000024	I*4	EMBSL_SP_IOSB2	3-00000018	I*4	EMBSL_SP_MEDIA
3-0000002C	I*4	EMBSL_SP_OPCNT	3-00000034	I*4	EMBSL_SP_OWNJIC
3-0000001C	I*4	EMBSL_SP_RQPID	3-00000041	CHAR	EMBST_SP_NAME
3-00000004	I*2	EMBSW_HD_ENTRY	3-0000000E	I*2	EMBSW_HD_ERRSEQ
3-00000012	I*2	EMBSW_SP_BOFF	3-00000030	I*2	EMBSW_SP_ERRCNT
3-00000028	I*2	EMBSW_SP_FUNC	3-00000032	I*2	EMBSW_SP_STS
3-0000002A	I*2	EMBSW_SP_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-00000008a	I*4	RECORD_LENGTH	AP-0000000Ca	I*4	RECORD_NUMBER

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	2-00000000	I*4	VOLUME_MOUNT_FLAG
4-00000018	L*1	VOLUME_OUTPUT			

ARRAYS

Address	Type	Name	Bytes	Dimensions
3-00000000	L*1	EMB	512	(0:511)
3-00000006	I*4	EMB\$Q_HD_TIME	8	(2)
2-00000000	L*1	MOUNT_FLAG_AND_LABEL_ARRAY	16	(16)

LABELS

Address	Label
0-000000B7	5

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name	Type	Name
	DHEAD2		DISK_TAPE_DRIVERS_MSCP_DQ		DISK_TAPE_DRIVERS_MSCP_Q
	DISK_TAPE_DRVR_MSCP_DISPATCHER		ERLLOGSTS		GET_CURRENT_LABEL
	HEADER		INTERVENE_DECREMENT		LOGGER
	PUDRIVER_MSCP_DISPATCHER				

```
0001
0002
0003
0004      Subroutine ERLLOGSTS (lun)
0005
0006      include 'src$:msghdr.for /nolist'
0065      include 'src$:embspdef.for /nolist'
0178      include 'src$:syecom.for /nolist'
0306
0307
0308      byte          lun
0309
0310      integer*4     compress4
0311
0312      C
0313      C Decode/output the entry header.
0314      C
0315      Call FRCTOF (lstlun)
0316      Call HEADER (lstlun)
0317      Call LOGGER (lstlun,'ERL$LOGSTATUS ENTRY')
0318      Call DHEAD2 (lstlun,'I/O',emb$b_sp_namng,emb$t_sp_name,emb$w_sp_unit)
0319
0320
0321      Entry ERLLOGSTS2 (lun)
0322
0323      C
0324      C Call the applicable routines to decode/output the software status
0325      C entry for an mscp disk/tape device.
0326      C
0327      C
0328      Call LINCHK (lstlun,1)
0329      write(lstlun,10)
0330      10 format(' ',:)
0331
0332      Call MSLG$$L_CMD_REF (lstlun,emb$l_sp_cmdref)
0333      Call ORB$L_OWNER (lstlun,emb$l_sp_ownuic)
0334      Call UCBSL_CHAR (lstlun,emb$l_sp_char)
0335
0336      Call UCBSL_OPCNT (lstlun,emb$l_sp_opcnt)
0337      Call UCBSW_ERRCNT (lstlun,emb$w_sp_errcnt)
0338      Call UCBSW_STS (lstlun,emb$w_sp_sts)
0339
0340      Call LINCHK (lstlun,1)
0341      write(lstlun,10)
0342
0343      Call CDRP$L_MEDIA (lstlun,emb$l_sp_media)
0344
0345      if (emb$b_sp_class .eq. 1) then          ! Disk qio func decode
0346      Call DUDRIVER_QIO (lstlun,emb$w_sp_func)
0347
0348      Else if (emb$b_sp_class .EQ. 2) then    ! Tape qio func decode
0349      Call TUDRIVER_QIO (lstlun,emb$w_sp_func)
0350
0351      else
0352      Call CDRP$W_FUNC (lstlun,emb$w_sp_func,'QIO FUNCTION')
0353      endif
0354
```



```

0355      Call CDRPSL_BCNT (lstlun,emb$l_sp_bcnc)
0356      Call CDRPSW_BOFF (lstlun,emb$w_sp_boff)
0357      Call CDRPSL_PID (lstlun,emb$l_sp_rgpdc)
0358      Call CDRPSQ_IOSB (lstlun,emb$l_sp_iosb1)
0359
0360      return
0361
0362      end
    
```

PROGRAM SECTIONS

Name	Bytes	Attributes
0 \$CODE	265	PIC CON REL LCL SHR EXE RD NOWRT LONG
1 \$PDATA	46	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2 \$LOCAL	252	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		1119

ENTRY POINTS

Address	Type	Name	Address	Type	Name
0-00000000		ERLLOGSTS	0-0000002B		ERLLOGSTS2

VARIABLES

Address	Type	Name	Address	Type	Name
2-00000000	I*4	COMPRESS4	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	EMB\$B_SP_CLASS	3-00000040	L*1	EMB\$B_SP_NAMLANG
3-00000011	L*1	EMB\$B_SP_TYPE	3-00000000	I*4	EMB\$l_HD_SID
3-00000014	I*4	EMB\$l_SP_BCNT	3-00000038	I*4	EMB\$l_SP_CHAR
3-0000003C	I*4	EMB\$l_SP_CMDREF	3-00000020	I*4	EMB\$l_SP_IOSB1
3-00000024	I*4	EMB\$l_SP_IOSB2	3-00000018	I*4	EMB\$l_SP_MEDIA
3-0000002C	I*4	EMB\$l_SP OPCNT	3-00000034	I*4	EMB\$l_SP_OWNUIC
3-0000001C	I*4	EMB\$l_SP_RQPID	3-00000041	CHAR	EMB\$T_SP_NAME
3-00000004	I*2	EMB\$W_HD_ENTRY	3-0000000E	I*2	EMB\$W_HD_ERRSEQ
3-00000012	I*2	EMB\$W_SP_BOFF	3-00000030	I*2	EMB\$W_SP_ERRCNT
3-00000028	I*2	EMB\$W_SP_FUNC	3-00000032	I*2	EMB\$W_SP_STS
3-0000002A	I*2	EMB\$W_SP_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
4P-00000004	L*1	LUN	4-0000001F	I*4	MAILBOX_CHANNEL
4-0000002B	CHAR	OPTIONS	4-00000008	L*4	PRINTER
4-00000000	I*4	RECCNT	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)
3-00000006	I*4	EMBSQ_HD_TIME	8	(2)

LABELS

Address	Label
1-00000029	10'

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name	Type	Name
	CDRPSL_BCNT		CDRPSL_MEDIA		CDRPSL_PID
	CDRPSQ_IOSB		CDRPSW_BOFF		CDRPSW_FUNC
	DHEAD2		DUDRIVER_QIO		FRCTOF
	HEADER		LINCHK		LOGGER
	MSLGSSL_CMD_REF		ORBSL_OWNER		TUDRIVER_QIO
	UCBSL_CHAR		UCBSL_OPCNT		UCBSW_ERRCNT
	UCBSW_STS				

COMMAND QUALIFIERS

FORTRAN /LIS=LISS:ERLLOGSTS/OBJ=OBJ\$:ERLLOGSTS MSRC\$:ERLLOGSTS

/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:	4.44 seconds
Elapsed Time:	12.43 seconds
Page Faults:	167
Dynamic Memory:	175 pages

This image displays a 13x10 grid of technical diagrams and listings, primarily consisting of line art and text-based representations of hardware and software components. Each cell in the grid contains a unique diagram, often including a title and associated data.

Key diagrams and titles include:

- GETCODE LIS
- EXECIMAGE LIS
- ERFSUMM LIS
- ERFSTAPEVE LIS
- FILES LIS
- ERLOGSTS LIS
- ERLOGMSG LIS
- IMAGeload LIS
- INITPROC1 LIS
- INITREAL LIS
- INITBUS LIS
- INITPROC4 LIS
- RM53271 LIS
- INITPROC2 LIS
- INITTAPE LIS
- INITDISK LIS
- INITPROC5 LIS
- INITPROC3 LIS
- INTERVENE LIS
- ERFRTVEC LIS
- ERFSUMVEC LIS

The diagrams use various symbols, including rectangular boxes, lines, and alphanumeric characters, to represent complex systems and their interactions. Some diagrams also feature small text labels and numerical data points.