

[illegible]

MF
Ta

```

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
LLLLLLLLLLLL IIIIII          SSSSSSSS
LLLLLLLLLLLL IIIIII          SSSSSSSS

```


MPERRLOG
Table of contents

- MULTI-PROCESSING ERROR LOG SUPPORT^{C 2} ROU 16-SEP-1984 02:00:51 VAX/VMS Macro V04-00

Page 0

(1) 70
(1) 115
(1) 138

MPSS\$ALLOCEMB - Allocate secondary error message buffer
MPSS\$RELEASEMB - Release secondary error message buffer
MPSS\$COLDSTART/MPSS\$WARMSTART - Build secondary error log messages

```
0000 1 :
0000 2 : Version: 'V04-000'
0000 3 :
0000 4 :
0000 5 : .MCALL MFPR
0000 6 : .TITLE MPERRLOG - MULTI-PROCESSING ERROR LOG SUPPORT ROUTINES
0000 7 : .IDENT 'V04-000'
0000 8 :
0000 9 : *****
0000 10 :
0000 11 : *
0000 12 : * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 13 : * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 14 : * ALL RIGHTS RESERVED.
0000 15 : *
0000 16 : *
0000 17 : * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 18 : * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 19 : * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 20 : * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 21 : * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 22 : * TRANSFERRED.
0000 23 : *
0000 24 : *
0000 25 : * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 26 : * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 27 : * CORPORATION.
0000 28 : *
0000 29 : *
0000 30 : * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 31 : * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 32 : *
0000 33 : *
0000 34 : *****
0000 35 :
0000 36 : ++
0000 37 : Facility: Executive , Hardware fault handling
0000 38 :
0000 39 : Abstract: Error logging routines used by secondary processor.
0000 40 :
0000 41 : Environment: MODE=Kernel
0000 42 :
0000 43 : Author: Kathleen D. Morse, Creation date: 07-Jul-1981
0000 44 :
0000 45 : Modified by:
0000 46 :
0000 47 : V03-003 KDM0066 Kathleen D. Morse 3-Aug-1983
0000 48 : Change PR$_TODR to PR780$_TODR.
0000 49 :
0000 50 : V03-002 KDM0012 Kathleen D. Morse 20-Sep-1982
0000 51 : Add second error log buffer.
0000 52 :
0000 53 : Version ,
0000 54 : of -
0000 55 : --
```


0000	51	:		
0000	52	:	Macro Library Calls	
0000	53	:		
0000	54	:		
0000	55	:	\$EMBDEF <SU>	;Error log message offsets
0000	56	:	\$MPSDEF	;Define secondary request bits
0000	57	:	\$PRDEF	;Processor register definitions
0000	58	:	\$PR780DEF	;11/780-specific IPR definitions
0000	59	:		
0000	60	:		
0000	61	:	Equated Symbols	
0000	62	:		
0000	63	:		
0000	64	:		
0000	65	:	Local Data	
0000	66	:		
0000	67	:		
00000000	68	:	.PSECT WIONONPAGED	

```
0000 70 .SBTTL MPSS$ALLOCEMB - Allocate secondary error message buffer
0000 71 :+
0000 72 : MPSS$ALLOCEMB - Allocate secondary error message buffer
0000 73 :
0000 74 : This routine is called by the secondary processor to allocate
0000 75 : the error log buffer. There is currently only one buffer, which
0000 76 : must be emptied by the primary before it can be re-used.
0000 77 :
0000 78 :
0000 79 : INPUTS:
0000 80 :
0000 81 : R1 - Size of message buffer required
0000 82 :
0000 83 : OUTPUTS:
0000 84 :
0000 85 : R0 - Low bit clear if allocation failure
0000 86 : - Low bit set if successful allocation
0000 87 : R1 - Secondary error log sequence number
0000 88 : R2 - Address of error log buffer, if successful allocation
0000 89 :
0000 90 :-
0000 91 :
0000 92 MPSS$ALLOCEMB::
0000 93 INCL W^MPSS$GL_ERLSEQNUM ;Allocate secondary error message buffer
0000 94 CMPL R1,#MPSS$R_ERLBUFSIZ ;Record another error log attempt
0000 95 BGTRU 40$ ;Is secondary buffer large enough?
0000 96 BBSSI #MPSS$V_ERLBUF1,W^MPSS$GL_ERLBUFIND,30$ ;Br if buffer busy
0000 97 MOVAB W^MPSS$AL_ERLBUF1,R2 ;Br if buffer busy
0000 98 10$: MOVW R1,EMB$W_SIZE(R2) ;Get address of error log buffer
0000 99 MOVL W^MPSS$GL_ERLSEQNUM,R1 ;Set size of this error message
0000 100 MOVW R1,EMB$W_HD_ERRSEQ(R2) ;Secondary error log sequence # for msg
0000 101 20$: MOVQ G^EXESGQ_SYSTIME,EMB$Q_HD_TIME(R2) ;Set secondary error log sequence #
0000 102 CMPL G^EXESGQ_SYSTIME,EMB$Q_HD_TIME(R2) ;Get system time
0000 103 BNEQ 20$ ;Verify that the time
0000 104 CMPL G^EXESGQ_SYSTIME+4,EMB$Q_HD_TIME+4(R2) ;acquired is valid and
0000 105 BNEQ 20$ ; is not being modified
0000 106 MFPR #PR$ SID,EMB$L_HD_SID(R2) ;Set system ID for this processor
0000 107 MOVZBL #1,R0 ;Indicate successful allocation
0000 108 RSB ;Return
0000 109 30$: BBSSI #MPSS$V_ERLBUF2,W^MPSS$GL_ERLBUFIND,40$ ;Br if buffer busy
0000 110 MOVAB W^MPSS$AL_ERLBUF2,R2 ;Br if buffer busy
0000 111 BRB 10$ ;Get address of error log buffer
0000 112 40$: CLRL R0 ;Join common code
0000 113 RSB ;Indicate failure to allocate buffer
;Return
```

0000'CF D6 0000 93
00000200 8F 51 D1 0004 94
48 1A 000B 95
35 0000'CF 00 E6 000D 96
52 0000'CF 9E 0013 97
FC A2 51 B0 0018 98
51 0000'CF D0 001C 99
OE A2 51 B0 0021 100
06 A2 00000000'GF 7D 0025 101
06 A2 00000000'GF D1 002D 102
EE 12 0035 103
0A A2 00000004'GF D1 0037 104
E4 12 003F 105
50 01 9A 0041 106
05 0044 107
05 0047 108
07 0000'CF 01 E6 0048 109
52 0000'CF 9E 004E 110
C3 11 0053 111
50 D4 0055 112
05 0057 113


```

0058 115 .SBTTL MPSS$RELEASEMB - Release secondary error message buffer
0058 116 :+
0058 117 : MPSS$RELEASEMB - Release secondary error message buffer
0058 118 :
0058 119 : This routine is called by the secondary when the error log buffer
0058 120 : contains a completed error message. The primary processor is interrupted,
0058 121 : with a request to enter the error log message for the secondary.
0058 122 :
0058 123 : INPUTS:
0058 124 :
0058 125 :     None
0058 126 :
0058 127 : OUTPUTS:
0058 128 :
0058 129 :     The primary processor is interrupted to log the secondary's message.
0058 130 :
0058 131 :-
0058 132 :
0058 133 MPSS$RELEASEMB::
0058 134 BBSSI #MPSS$V SECERRLOG,W*MPSS$GL SECREQFLG,10$ ;Release error message buffer
0058 135 10$: BSBW W*MPSS$INTPRIM ;Error log message ready
0061 136 RSB ;Interrupt primary to log message
;

```

00 0000'CF 01 E6
FF9F' 30 005E
05 0061

```
0062 138 .SBTTL MPSSCOLDSTART/MPSSWARMSTART - Build secondary error log messages
0062 139 :+
0062 140 : MPSSCOLDSTART - log coldstart (system boot)
0062 141 :
0062 142 : This routine is called by SYSINIT after correctly setting the system
0062 143 : time to log the booting of the system.
0062 144 :
0062 145 : MPSSWARMSTART - Log warmstart (power recovery)
0062 146 :
0062 147 : This routine is called by powerfail after correcting the system time
0062 148 : to log the power failure and recovery.
0062 149 :
0062 150 : INPUTS:
0062 151 :
0062 152 : None
0062 153 :
0062 154 : OUTPUTS:
0062 155 :
0062 156 : The error log buffer is allocated and filled with the appropriate
0062 157 : message if possible.
0062 158 :-
0062 159 .ENABL LSB
0062 160 MPSSCOLDSTART::
0062 161 MOVZWL #EMBSC_CS,R3 ;Set type of message to coldstart
0065 162 BRB 10$ ;
0067 163 MPSSWARMSTART::
0067 164 MOVZWL #EMBSC_WS,R3 ;Set type of message to warmstart
006A 165 10$: MOVZWL #EMBSC_SU_LENGTH,R1 ;Set size of message to allocate
006D 166 BSBB B*MPSS$ALLOCCEMB ;Allocate secondary error log buffer
006F 167 BLBC R0,20$ ;Br if not available
0072 168 MFPR #PR780$ TODR,EMB$SU_DAYTIM(R2) ;Log time of day clock
0086 169 MOVW R3,EMB$SU_ENTRY(R2) ;Set message type
008A 170 BSBB B*MPSS$RELEASEMB ;Release buffer
008C 171 20$: RSB ;
008D 172 .DSABL LSB
008D 173 .END
```

53 20 3C
03 11
53 24 3C
51 14 3C
91 10
4A 50 E9
04 A2 53 B0
9C 10
05

MPERRLOG
Symbol table

- MULTI-PROCESSING ERROR LOG SUPPORT¹ ROU 16-SEP-1984 02:00:51 VAX/VMS Macro V04-00
5-SEP-1984 02:06:17 [MP.SRC]MPERRLOG.MAR;1

Page 6
(1)

```

EMB$C_CS = 00000020
EMB$C_SU_LENGTH = 00000014
EMB$C_WS = 00000024
EMB$H_HD_SID = 00000000
EMB$H_SU_DAYTIM = 00000010
EMB$H_HD_TIME = 00000006
EMB$H_HD_ERRSEQ = 0000000E
EMB$H_SIZE = FFFFFFFC
EMB$H_SU_ENTRY = 00000004
EX$G$C_TODR ***** X 02
EX$G$Q_SYSTIME ***** X 02
EX$G$Q_TODCBASE ***** X 02
MP$S$ALCOCEMB 00000000 RG 02
MP$S$AL_ERLBUF1 ***** X 02
MP$S$AL_ERLBUF2 ***** X 02
MP$S$COCDSTART 00000062 RG 02
MP$S$GL_ERLBUFIND ***** X 02
MP$S$GL_ERLSEQNUM ***** X 02
MP$S$GL_SECREQFLG ***** X 02
MP$S$INTPRIM ***** X 02
MP$S$K_ERLBUFSIZ = 00000200
MP$S$RELEASEMB 00000058 RG 02
MP$S$V_ERLBUF1 = 00000000
MP$S$V_ERLBUF2 = 00000001
MP$S$V_SECERRLOG = 00000001
MP$S$WARMSTART 00000067 RG 02
PR$SID = 0000003E
PR780$_TODR = 0000001B

```

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
WIONONPAGED	000000BD (189.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	32	00:00:00.09	00:00:00.59
Command processing	128	00:00:00.82	00:00:05.12
Pass 1	172	00:00:03.28	00:00:14.96
Symbol table sort	0	00:00:00.22	00:00:00.63
Pass 2	54	00:00:00.75	00:00:03.04
Symbol table output	5	00:00:00.04	00:00:00.04
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	395	00:00:05.22	00:00:24.40

The working set limit was 1200 pages.
14377 bytes (29 pages) of virtual memory were used to buffer the intermediate code.

There were 20 pages of symbol table space allocated to hold 220 non-local and 8 local symbols.
178 source lines were read in Pass 1, producing 13 object records in Pass 2.
15 pages of virtual memory were used to define 14 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name	Macros defined
-----	-----
\$255\$DUA28:[MP.OBJ]MP.MLB;1	4
\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	4
\$255\$DUA28:[SYSLIB]STARLET.MLB;2	5
TOTALS (all libraries)	13

367 GETS were required to define 13 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LISS:MPERRLOG/OBJ=OBJ\$:MPERRLOG MSRC\$:MPPREFIX/UPDATE=(ENH\$:MPPREFIX)+MSRC\$:MPERRLOG/UPDATE=(ENH\$:MPERRLOG)+EXECML\$/LIB+LI

0248 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

