

RRRRRRRRRRR		TTTTTTTTTTTT	PPPPPPPPPPP		AAAAAAAAA		DDDDDDDDDDD	
RRRRRRRRRRR		TTTTTTTTTTTT	PPPPPPPPPPP		AAAAAAAAA		DDDDDDDDDDD	
RRRRRRRRRRR		TTTTTTTTTTTT	PPPPPPPPPPP		AAAAAAAAA		DDDDDDDDDDD	
RRR	FRR	TTT	PPP	PPP	AAA	AAA	DDD	DDD
RRR	RRR	TTT	PPP	PPP	AAA	AAA	DDD	DDD
RRR	RRR	TTT	PPP	PPP	AAA	AAA	DDD	DDD
RRR	RRR	TTT	PPP	PPP	AAA	AAA	DDD	DDD
RRR	RRR	TTT	PPP	PPP	AAA	AAA	DDD	DDD
RRR	RRR	TTT	PPP	PPP	AAA	AAA	DDD	DDD
RRR	RRR	TTT	PPP	PPP	AAA	AAA	DDD	DDD
RRRRRRRRRRR		TTT	PPPPPPPPPPP		AAA	AAA	DDD	DDD
RRRRRRRRRRR		TTT	PPPPPPPPPPP		AAA	AAA	DDD	DDD
RRRRRRRRRRR		TTT	PPPPPPPPPPP		AAA	AAA	DDD	DDD
RRR	RRR	TTT	PPP	PPP	AAAAAAAAAAAAA	AAA	DDD	DDD
RRR	RRR	TTT	PPP	PPP	AAAAAAAAAAAAA	AAA	DDD	DDD
RRR	RRR	TTT	PPP	PPP	AAAAAAAAAAAAA	AAA	DDD	DDD
RRR	RRR	TTT	PPP	PPP	AAA	AAA	DDD	DDD
RRR	RRR	TTT	PPP	PPP	AAA	AAA	DDD	DDD
RRR	RRR	TTT	PPP	PPP	AAA	AAA	DDD	DDD
RRR	RRR	TTT	PPP	PPP	AAA	AAA	DDDDDDDDDDD	DDD
RRR	RRR	TTT	PPP	PPP	AAA	AAA	DDDDDDDDDDD	DDD
RRR	RRR	TTT	PPP	PPP	AAA	AAA	DDDDDDDDDDD	DDD

```

RRRRRRR      TTTTTTTTT  LL      000000  GGGGGGGG  IIIIII  000000
RRRRRRR      TTTTTTTTT  LL      000000  GGGGGGGG  IIIIII  000000
RR      RR    TT      LL      00      00  GG      II      00      00
RR      RR    TT      LL      00      00  GG      II      00      00
RR      RR    TT      LL      00      00  GG      II      00      00
RR      RR    TT      LL      00      00  GG      II      00      00
RRRRRRR      TT      LL      00      00  GG      II      00      00
RRRRRRR      TT      LL      00      00  GG      II      00      00
RR      RR    TT      LL      00      00  GG      II      00      00
RR      RR    TT      LL      00      00  GG      II      00      00
RR      RR    TT      LL      00      00  GG      II      00      00
RR      RR    TT      LL      00      00  GG      II      00      00
RR      RR    TT      LL      00      00  GG      II      00      00
RR      RR    TT      LL      00      00  GG      II      00      00
RR      RR    TT      LLLLLLLLLL 000000  GGGGGG  IIIIII  000000
RR      RR    TT      LLLLLLLLLL 000000  GGGGGG  IIIIII  000000

```

```

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

```

```

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)	61
(3)	107
(4)	178
(5)	247
(6)	287

DECLARATIONS
CTERMSLOG IO - LOG I/O
CTERMSOPEN LOG - OPEN LOG FILE
CTERMSCLOSE LOG - CLOSE LOG FILE IF OPEN
RTLOG\$WRITE_STRING - write string to log file

```

0000 1 .TITLE RTLOGIO - CTERM LOG IO
0000 2 .IDENT 'V04-000'
0000 3
0000 4 :*****
0000 5 :
0000 6 :* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 7 :* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 8 :* ALL RIGHTS RESERVED.
0000 9 :
0000 10 :* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 11 :* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 12 :* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 13 :* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 14 :* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 15 :* TRANSFERRED.
0000 16 :
0000 17 :* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 18 :* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 19 :* CORPORATION.
0000 20 :
0000 21 :* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 22 :* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 23 :
0000 24 :
0000 25 :*****
0000 26 :
0000 27 :
0000 28 :++
0000 29 :
0000 30 : FACILITY: RTPAD, SET HOST
0000 31 :
0000 32 : ABSTRACT:
0000 33 :
0000 34 : This module provides the support for doing logging of an interactive
0000 35 : terminal session.
0000 36 :
0000 37 : ENVIRONMENT:
0000 38 :
0000 39 : user mode.
0000 40 :
0000 41 :--
0000 42 :
0000 43 : AUTHOR: Jake VanNoy, CREATION DATE: 5-Jun-1983
0000 44 :
0000 45 : MODIFIED BY:
0000 46 :
0000 47 : V03-003 JLV0352 Jake VanNoy 10-APR-1984
0000 48 : Add CALL entry point for DTE code.
0000 49 :
0000 50 : V03-002 JLV0333 Jake VanNoy 28-FEB-1984
0000 51 : Add minor amount of code that is only used for debug
0000 52 : purposes. Outputs trace to RTPAD$TRACE.
0000 53 :
0000 54 : V03-001 JLV0323 Jake VanNoy 10-JAN-1984
0000 55 : Fix up error handling. Add write behind to options to reduce
0000 56 : overhead of /LOG.
0000 57 :

```

RTLOGIO
V04-000

- CTERM LOG .0

K 12

16-SEP-1984 02:13:04 VAX/VMS Macro V04-00
5-SEP-1984 03:15:40 [RTPAD.SRC]RTLOGIO.MAR;1

Page 2
(1)

0000 58 ;**
0000 59

```

0000 61 .SBTTL DECLARATIONS
0000 62 :
0000 63 : INCLUDE FILES:
0000 64 :
0000 65 $RTPADDEF ; Miscellaneous symbols
0000 66 $TSADEF ; CTERM packet defintion
0000 67 :
0000 68 :
0000 69 : MACROS:
0000 70 :
0000 71 :
0000 72 :
0000 73 : EQUATED SYMBOLS:
0000 74 :
0000 75 :
0000 76 :
0000 77 : OWN STORAGE:
0000 78 :
0000 79 :
00000000 80 .PSECT _RTPAD, LONG ; READ/WRITE DATA
0000 81 :
0000 82 .ALIGN LONG
0000 83 :
0000 84 cterm_fab: $FAB DNM = <SETHOST.LOG>,-
0000 85 FAC = PUT,-
0000 86 RAT = CR,-
0000 87 FOP = SQO,- ; sequential only
0000 88 ORG = SEQ,- ; sequential
0000 89 RFM = VAR,- ; variable
0000 90 NAM = cterm_nam
0050 91 :
0050 92 cterm_rab: $RAB FAB = cterm_fab,-
0050 93 RAC = SEQ,-
0050 94 ROP = WBH ; write behind for performan
0094 95 :
0094 96 cterm_nam: $NAM ESA = file_str,-
0094 97 ESS = nam$c_maxrss
00F4 98 :
00000000 00F4 99 filespec: .long 0
000000FC 00F8 100 .address file_str
000001FB 00FC 101 file_str: .blkb nam$c_maxrss
01FB 102 :
47 4F 4C 2E 01FB 103 LOG_A_EXT: .ascii /.LOG/
00000004 01FF 104 LOG_S_EXT = .-LOG_A_EXT
01FF 105 :

```

```

000001FF 107 .SBTTL CTERM$LOG_IO - LOG I/O
00000000 108 .PSECT RTPAD,nowrt
0000 109 :++
0000 110 :
0000 111 : FUNCTIONAL DESCRIPTION:
0000 112 :
0000 113 :
0000 114 : CALLING SEQUENCE:
0000 115 :
0000 116 : BSBW CTERM$LOG_IO
0000 117 :
0000 118 : INPUT PARAMETERS:
0000 119 :
0000 120 : R1 - address of string
0000 121 : R2 - length
0000 122 :
0000 123 : IMPLICIT INPUTS:
0000 124 : NONE
0000 125 :
0000 126 : OUTPUT PARAMETERS:
0000 127 : NONE
0000 128 :
0000 129 : IMPLICIT OUTPUTS:
0000 130 :
0000 131 : An error writing the output file may turn off logging.
0000 132 :
0000 133 : COMPLETION CODES:
0000 134 : NONE
0000 135 :
0000 136 : SIDE EFFECTS:
0000 137 : NONE
0000 138 :
0000 139 :--
0000 140 :
0000 141 CTERM$LOG_IO::
0000 142 :
54 0050' 17 BB 0000 143 PUSHR #*M<R0,R1,R2,R4>
CF 9E 0002 144 MOVAB W*CTERM_RAB,R4 ; address of rab
0007 145 :
52 02 D1 0007 146 CMPL #2,R2 ; Must be at least 2 bytes long
OD 1E 000A 147 BGEQU 10$ ; or skip cr,lf check
000C 148 :
61 0A0D 8F B1 000C 149 CMPW #*X0A0D,(R1) ; CR, LF at start of record?
06 12 0011 150 BNEQ 10$ ; if not, branch
51 02 C0 0013 151 ADDL #2,R1 ; Add
52 02 C2 0016 152 SUBL #2,R2 ; Subtract
0019 153 10$:
28 A4 51 D0 0019 154 MOVL R1,RAB$L_RBF(R4) ; set buffer address
22 A4 52 B0 001D 155 MOVW R2,RAB$W_RSZ(R4) ; and size
0021 156 :
03 50 E9 002A 157 $PUT RAB = R4 ; output line
002D 158 BLBC R0,250$ ; signal error
002D 159 200$:
17 BA 002D 160 POPR #*M<R0,R1,R2,R4> ; restore
05 002F 161 RSB ; return
0030 162 250$:
0030 163

```

		0030	164
		0030	165
00000000'8F	DD	0030	166
50	DD	0036	167
000000F4'EF	9F	0038	168
01	DD	003E	169
00000000'6F	DD	0040	170
00000000'GF	05	FB	0046
			004D
			004D
G077	30		0050
			0050
DB	11		0052
			0052

```

; signal error and turn off logging
PUSHL #SS$_MUSTCLOSEFL ; closing file
PUSHL R0 ; real status
PUSHAB FILESPEC ; file spec
PUSHL #1 ; FAO arg count
PUSHL #SHR$_WRITEERR!<REMS_FACILITY@16> ; logging error
CALLS #5,G^[IB$SIGNAL ; Signal error

BSBW CTERM$CLOSE_LOG ; close file

BRB 200$ ; exit

```

RTLOGIO V04-000


```

0052 178 .SBTTL CTERM$OPEN_LOG - OPEN LOG FILE
0052 179 :++
0052 180 :
0052 181 : FUNCTIONAL DESCRIPTION:
0052 182 :
0052 183 :     Opens log file and turns on logging flag if successful.
0052 184 :
0052 185 : CALLING SEQUENCE:
0052 186 :
0052 187 :     CALLS #1,CTERM$OPEN_LOG
0052 188 :
0052 189 : INPUT PARAMETERS:
0052 190 :
0052 191 :     4(AP) - Descriptor of log file to open (zero if none specified)
0052 192 :
0052 193 : IMPLICIT INPUTS:
0052 194 :     NONE
0052 195 :
0052 196 : OUTPUT PARAMETERS:
0052 197 :     NONE
0052 198 :
0052 199 : IMPLICIT OUTPUTS:
0052 200 :
0052 201 :     If no errors, FLG$M_LOGGING will be set in CTERM_FLAG
0052 202 :
0052 203 : COMPLETION CODES:
0052 204 :
0052 205 :     RMS codes from CREATE and CONNECT
0052 206 :
0052 207 : SIDE EFFECTS:
0052 208 :
0052 209 :
0052 210 :--
0052 211 :
000C 0052 212 .ENTRY CTERM$OPEN_LOG, ^M<R2,R3>
0054 213
18 52 00000000'EF 9E 0054 214     MOVAB  CTERM_FAB,R2           ; address of FAB
      53 04 AC  D0 005B 215     MOVL   4(AP),R3             ; Fetch descriptor
      15 13 005F 216     BEQL   20$                ; If none, then use default
0061 217
      34 A2 63 90 0061 218     MOVB   (R3),FAB$B_FNS(R2)   ; size of file spec
2C  A2 04 A3  D0 0065 219     MOVL   4(R3),FAB$C_FNA(R2) ; address of file spec
      35 A2 04 90 006A 220     MOVB   #LOG_S_EXT,FAB$B_DNS(R2); size of default extension
30 A2 000001FB'EF 9E 006E 221     MOVAB  LOG_A_EXT,FAB$B_DNA(R2); Address of default extension
0076 222 20$:
18 A2 00000000'8F D0 0076 223     MOVL   #SHR$_OPENOUT!<REMS_FACILITY@16>,FAB$B_CTX(R2) ; In case of error
      30 50  E9 007E 224     $CREATE FAB = (R2)         ; Create file
      0087 225     BLBC  R0,50$              ; Branch on error
      008A 226
      52 0094'CF 9E 008A 227     MOVAB  W^CTERM_NAM,R2     ; address of nam
000000F4'EF 0B A2 9B 008F 228     MOVZBW NAM$B_ESL(R2),FILESPEC ; length
      0097 229
      52 00000050'EF 9E 0097 230     MOVAB  CTERM_RAB,R2       ; address of rab
18 A2 00000000'8F D0 009E 231     MOVL   #SHR$_OPENOUT!<REMS_FACILITY@16>,RAB$B_CTX(R2) ; In case of error
      00A6 232     $CONNECT RAB = (R2)      ; Connect to file
      08 50  E9 00AF 233     BLBC  R0,50$              ; Branch on error
      00B2 234

```

RT
Sy
TT
UT
V-
WA
PS
--
\$A
R
\$R
RT
Ph
--
In
Co
Pa
Sy
Pa
Sy
Ps
Cr
As
Th
64
Th
57
39
Ma
--
\$
--
\$
--
\$
TC
13
Th
MA

```
00000000'EF 08 A8 00B2 235 BISW #FLG$M_LOGGING,-  
00B4 236 CTERM_FLAG ; set logging flag  
04 00B9 237 RET ; exit  
00BA 238 ;  
00BA 239 ; Error has occurred  
00BA 240 ;  
00BA 241 50$:  
00BA 242 PUSHL R2 ; address of RMS context block  
00BC 243 CALLS #1,G^UTIL$REPORT_IO_ERROR ; Signal error  
00C3 244 MOVL #1,R0 ; set success  
04 00C6 245 RET ; exit
```

- CTERM LOG IO
CTERM\$CLOSE_LOG

- CLOSE LOG FILE IF OPEN
.SBTTL CTERM\$CLOSE_LOG - CLOSE LOG FILE IF OPEN

```

00C7 247 :
00C7 248 :++
00C7 249 :
00C7 250 : FUNCTIONAL DESCRIPTION:
00C7 251 :
00C7 252 :     Opens log file and turns on logging flag if successful.
00C7 253 :
00C7 254 : CALLING SEQUENCE:
00C7 255 :
00C7 256 :     BSBW     CTERM$CLOSE_LOG
00C7 257 :
00C7 258 : INPUT PARAMETERS:
00C7 259 :     NONE
00C7 260 :
00C7 261 : IMPLICIT INPUTS:
00C7 262 :
00C7 263 :     CTERM_FLAG
00C7 264 :
00C7 265 : OUTPUT PARAMETERS:
00C7 266 :     NONE
00C7 267 :
00C7 268 : IMPLICIT OUTPUTS:
00C7 269 :
00C7 270 :     FLG$M_LOGGING will be cleared
00C7 271 :
00C7 272 : COMPLETION CODES:
00C7 273 :
00C7 274 : SIDE EFFECTS:
00C7 275 :
00C7 276 :--
00C7 277 :

```

```

10 00000000'EF 03
50 00000000'EF

```

```

E5 00C7 281      BBCC  #FLG$V_LOGGING,CTERM_FLAG,100$ ; branch if not logging, clear bit
9E 00CF 282      MOVAB CTERM_FAB,R0          ; FAB address
00D6 283      $CLOSE FAB=R0              ; close
00DF 284 100$:
05 00DF 285      RSB

```

```
00E0 287 .SBTTL RTLOG$WRITE_STRING - write string to log file
00E0 288
00E0 289
00E0 290 ; 4(AP) - address
00E0 291 ; 8(AP) - byte count
00E0 292
00G4 00E0 293 .entry rtlog$write_string,^M<R2>
00E2 294
51 04 AC D0 00E2 295      MOVL    4(AP),R1
52 08 AC D0 00E6 296      MOVL    8(AP),R2
   FF13 30 00EA 297      bsbw    cterm$log_io
   04    04 00ED 298      RET
00EE 299
```

```

00EE 301
00EE 302 ; DEBUG CODE FOR RTPAD/CTERMRT
00EE 303
000000EE 304 .psect RTPAD,nowrt ; READ ONLY DATA
00EE 305
49 2D 4D 45 52 25 000000F6'010E0000' 00EE 306 DEBUG_MSG: .ascid /%REM-I-DEBUGAST, ^\ enabled/
5E 20 2C 54 53 41 47 55 42 45 44 2D 00FC
64 65 6C 62 61 6E 65 20 5C 0108
00000000 0111 307 DEBUG_MASK: .long 0
10000000 0115 308 .long 1@28
0119 309
20 20 20 20 20 20 20 20 20 20 22 0119 310 DBG$L_zeros: .ascii /' /
20 20 20 20 20 0125
63 65 72 20 65 67 61 73 73 65 4D 00' 012A 311
6E 20 6D 6F 72 66 20 64 65 76 69 65 012A 312 DBG$LINKRECV:: .ascic /Message received from net:/
3A 74 65 1A 0142
69 72 77 20 65 67 61 73 73 65 4D 00' 0145 313 DBG$LINKWRITE:: .ascic /Message written to net: /
3A 74 65 6E 20 6F 74 20 6E 65 74 74 0151
20 20 20 015D
41 50 54 52 20 3A 74 6E 65 76 45 00' 0160 314 GOTVM:: .ascic /Event: RTPAD calling LIB$GET_VM/
49 4C 20 67 6E 69 6C 6C 61 63 20 44 016C
4D 56 5F 54 45 47 24 42 0178
1F 0160
20 43 41 21 2F 21 00000188'010E0000' 0180 315
67 6E 65 4C 43 41 21 43 41 21 28 20 0180 316 header_ctr: .ascid ./!AC (!AC!ACLength = !XL.
4C 58 21 20 20 20 000001AA'010E0000' 019A
4C 58 21 20 4C 58 21 20 4C 58 21 20 01A2 317 mem_ctr: .ascid / !XL !XL !XL !XL !AF (!XL)/
4C 58 21 28 20 20 46 41 21 20 20 20 01B0
29 01BC
49 2D 4D 45 52 25 000001D1'010E0000' 01C9 318 logctr: .ascid .%REM-I-TRACING, logging trace output to !AS.
6F 6C 20 2C 47 4E 49 43 41 52 54 2D 01D7
20 65 63 61 72 74 20 67 6E 69 67 67 01E3
41 21 20 6F 74 20 74 75 70 74 75 6F 01EF
53 01FB
0000026B'0000025D'00000254'00000357' 01FC 319
000002A0'00000291'00000287'00000278' 020C 320 message_table: .long 300$,10$,20$,30$,40$,50$,60$,70$,80$,90$,100$,110$,120$
000002E5'000002CE'000002BD'000002AA' 021C
000002F7' 022C
00000335'0000032A'00000315'00000306' 0230 321 .long 130$,140$,150$,160$,170$,300$,300$,300$,300$
00000357'00000357'00000357'00000348' 0240
00000357' 0250
09 09 09 29 54 49 4E 49 00' 0254 322
08 0254 323 10$: .ascic /INIT) /
29 44 41 45 52 20 54 52 41 54 53 00' 025D 324 20$: .ascic /START READ) /
09 09 0269
0D 025D
09 29 41 54 41 44 20 44 41 45 52 00' 0268 325 30$: .ascic /READ DATA) /
09 0277
0C 0268
44 4E 41 42 20 46 4F 20 54 55 4F 00' 0278 326 40$: .ascic /OUT OF BAND) /

```

R1

T1
ME

```

09 09 29 0284
0E 0278
09 09 29 44 41 45 52 4E 55 00' 0287 327 50$: .ascic /UNREAD) /
09 0287
54 55 50 4E 49 20 52 41 45 4C 43 00' 0291 328 60$: .ascic /CLEAR INPUT) /
09 09 29 029D
0E 0291
09 09 09 29 45 54 49 52 57 00' 02A0 329 70$: .ascic /WRITE) /
09 02A0
4C 50 4D 4F 43 20 45 54 49 52 57 00' 02AA 330 80$: .ascic /WRITE COMPLETION) /
09 29 4E 4F 49 54 45 02B6
12 02AA
54 55 4F 20 44 52 41 43 53 49 44 00' 02BD 331 90$: .ascic /DISCARD OUTPUT) /
09 29 54 55 50 02C9
10 02BD
43 41 52 41 48 43 20 44 41 45 52 00' 02CE 332 100$: .ascic /READ CHARACTERISTICS) /
09 29 53 43 49 54 53 49 52 45 54 02DA
16 02CE
53 49 52 45 54 43 41 52 41 48 43 00' 02E5 333 110$: .ascic /CHARACTERISTICS) /
09 29 53 43 49 54 02F1
11 02E5
54 55 50 4E 49 20 48 43 45 48 43 00' 02F7 334 120$: .ascic /CHECK INPUT) /
09 09 29 0303
0E 02F7
54 4E 55 4F 43 20 54 55 50 4E 49 00' 0306 335 130$: .ascic /INPUT COUNT) /
09 09 29 0312
0E 0306
45 54 41 54 53 20 54 55 50 4E 49 00' 0315 336 140$: .ascic /INPUT STATE CHANGE) /
09 29 45 47 4E 41 48 43 20 0321
14 0315
09 09 29 4F 49 51 20 53 4D 56 00' 032A 337 150$: .ascic /VMS QIO) /
0A 032A
41 4F 52 42 20 45 4E 49 4C 50 55 00' 0335 338 160$: .ascic /UPLINE BROADCAST) /
09 29 54 53 41 43 44 0341
12 0335
59 46 49 52 45 56 20 44 41 45 52 00' 0348 339 170$: .ascic /READ VERIFY) /
09 09 29 0354
0E 0348
6C 61 67 65 6C 6C 69 20 2A 2A 2A 00' 0357 340 300$: .ascic /*** illegal/
0B 0357
0363 341
2E 2E 2E 00' 0363 342 null_line: .ascic /.../
03 0363
0367 343
000003B5'000003AC'000003A7'00000401' 0367 344 msg_type_table: .long 150$,10$,20$,30$,40$,50$,60$,70$,80$,90$,100$,150$,150$
000003DB'000003D1'000003C6'000003BA' 0377
00000401'000003F9'000003F0'000003E8' 0387
00000401' 0397
00000401'00000401'00000401' 039B 345 .long 150$,150$,150$
03A7 346
64 6E 69 42 00' 03A7 347 10$: .ascic /Bind/
04 03A7
64 6E 69 62 6E 55 00' 03AC 348 20$: .ascic /Unbind/
06 03AC
64 6E 69 62 65 52 00' 03B3 349 30$: .ascic /Rebind/
06 03B3
74 70 65 63 63 41 20 64 6E 69 42 00' 03BA 350 40$: .ascic /Bind Accept/

```

65 64 6F 4D 20 72 65 74 6E 45	00'	03BA	351	50\$:	.ascic	/Enter Mode/
	0A'	03C6				
65 64 6i 4D 20 74 69 78 45	00'	03D1	352	60\$:	.ascic	/Exit Mode/
	09'	03D1				
64 6F 4D 20 6D 72 69 66 6E 6F 43	00'	03DB	353	70\$:	.ascic	/Confirm Mode/
	65'	03E7				
	0C'	03DB				
65 64 6F 4D 20 6F 4E	00'	03E8	354	80\$:	.ascic	/No Mode/
	07'	03E8				
20 2D 20 6D 72 65 74 43	00'	03F0	355	90\$:	.ascic	/Cterm - /
	08'	03F0				
20 2D 20 65 64 6F 4D	00'	03F9	356	100\$:	.ascic	/Mode - /
	07'	03F9				
6C 61 67 65 6C 6C 69	00'	0401	357	150\$:	.ascic	/illegal/
	07'	0401				
		0409	358			

```

0409 360
000001FF 361 .psect _RTPAD, LONG ; READ/WRITE DATA
01FF 362
01FF 363 .ALIGN LONG
0200 364
0000 0200 365 DEBUG_CHAN: .word 0
00000000 0202 366 DEBUG_BUF: .long 0
0000 0206 367 trace_flag: .word 0
0208 368
00000000 0208 369 v_open = 0
00000001 0208 370 m_open = 1
00000001 0208 371 v_error = 1
00000002 0208 372 m_error = 2
0208 373
00000000 0208 374 save_log: .long 0
020C 375
020C 376 .Align LONG
020C 377
020C 378 Logio_fab: $FAB FAC=PUT,RAT=CR
025C 379 logio_rab: $RAB FAB=logio_fab
24 44 41 50 54 52 000002A8'010E0000' 02A0 380 rtpad_dbg: .ascid /RTPAD$TRACE/
45 43 41 52 54 02AE
02B3 381
00000064 02B3 382 dbg$outbuf: .long 100
000002BB' 02B7 383 .long 10$
0000031F 02BB 384 10$: .blkb 100
031F 385
00000084 031F 386 file_name: .long 132
00000327' 0323 387 .long 1$
000003AB 0327 388 1$: .blkb 132
03AB 389
03AB 390 :
03AB 391 : RTPAD TRACE FACILITY (DEBUG CODE)
03AB 392 :
00000409 0409 393 .PSECT RTPAD,nowrt
0409 394
0409 395 open_trace:
0409 396
01 00000206'EF 00 E3 0409 397 BBS #v_open,trace_flag,5$ ; Branch if we've been here before
05 0411 398 rsb
0412 399
0C BB 0412 400 5$: PUSHR #^M<R2,R3>
50 000002A0'EF 9E 0414 401 movab rtpad_dbg,R0 ; address of logical name
0000031F'EF 00000084 8F D0 041B 402 movl #132,file_name
00000208'EF 50 D0 0426 403 movl r0,save_log
042D 404
042D 405 $STRNLOG_S LOGNAM = (R0),- ; Only trace if logical name defined
042D 406 RSLBUF = file_name,-
042D 407 RSLLEN = file_name
50 00000000'8F D1 0448 408 cml #$$$_NORMAL,R0
05 13 044F 409 BEQL 10$
50 D4 0451 410 CLRL R0
0068 31 0453 411 BRW 50$
0456 412
0456 413 10$: $FAO_S CTRSTR = LOGCTR -
0456 414 outbuf = dbg$outbuf,-
0456 415 outlen = dbg$outbuf,- ; length

```


			0456	416		P1 = #file_name	; logging to...
			0473	417			
	000002B3'EF	DF	0473	418		PUSHAL	DBG\$OUTBUF
	00000000'GF	01	FB	0479	419	CALLS	#1,G^LIB\$PUT_OUTPUT
				0480	420		
000002B3'EF	00000064	8F	D0	0480	421	movl	#100,dbg\$outbuf ; reset length
	52	0000020C'EF	9E	048B	422	movab	logio_fab,R2 ; fab address
	53	0000025C'EF	9E	0492	423	movab	logio_rab,R3 ; rab address
34	A2	0000031F'EF	90	0499	424	movb	file_name,fab\$b_fns(r2)
2C	A2	00000323'EF	D0	04A1	425	movl	file_name+4,fab\$l_fna(R2)
				04A9	426		
				04A9	427	\$CREATE	FAB = (R2)
	0C	50	E9	04B2	428	blbc	r0,50\$
				04B5	429		
				04B5	430	\$CONNECT	RAB = (R3)
	07	50	E8	04BE	431	blbs	r0,100\$
				04C1	432		50\$:
00000206'EF	02	C8	04C1	433		bisl	#m_error_trace_flag ; Set error
	0C	BA	04C8	434	100\$:	POPR	#^R<R2,R3>
		05	04CA	435		RSB	

```

00000200'EF  B5 04CB 437 DEBUG_SETUP:: ; Enable ^\ to take image dump...
      18 12 04CB 438 TSTW  DEBUG_CHAN
      04D1 439 BNEQ  10$
      04D3 440
      04D3 441 $ASSIGN_S CHAN = DEBUG_CHAN,-
      04D3 442 DEVNAM = TTYDESC
      38 50 E9 04E8 443 BLBC  R0,99$ ; error
      50 FC22 CF 9E 04EB 444 10$:
      04EB 445 MOVAB  DEBUG_MASK,R0 ; Address of mask
      04F0 446 $QIO_S CHAN = DEBUG_CHAN,-
      04F0 447 FUNC = #IOS$ SETMODE!IOSM_OUTBAND -
      04F0 448 P1 = DEBUG_AST -
      04F0 449 P2 = R0
      0B 50 E9 0515 450 BLBC  R0,99$
      FBD2 CF DF 0518 451
00000000'GF 01 FB 0518 452 PUSHAL  DEBUG_MSG ; SAY WE SET IT UP...
      051C 453 CALLS  #1,G^LIB$PUT_OUTPUT
      0523 454 99$:
      05 0523 455 RSB
      0524 456
      0000 0524 457 .ENTRY  DEBUG_AST,0
      0526 458
      7E DC 0526 459 MOVPSL -(SP)
      00000202'EF DD 0528 460 PUSHL  DEBUG_BUF
      02 DD 052E 461 PUSHL  #2
      00000000'8F DD 0530 462 PUSHL  #SS$ BREAK!<REMS_FACILITY@16>
00000000'GF 04 FB 0536 463 CALLS  #4,G^LIB$SIGNAL ; wake up Debug
      04 053D 464 RET
      053E 465

```

```

00000000 053E 467
00000004 053E 468 prototype = CTP$B_PRO MSGTYPE - CTP$B_PRO MSGTYPE
053E 469 msgtype = CTP$B_MSGTYPE - CTP$B_PRO_MSGTYPE
053E 470
053E 471
053E 472 DBG$TRACE_IO::
053E 473
053E 474 : R0 - scratch
053E 475 : R1 - length
053E 476 : R2 - address
053E 477 : R3 - header string
053E 478 : R4 - scratch
053E 479 : R5 - scratch
053E 480
01 00000206'EF 01 E1 053E 481 BBC #v_error,trace_flag,10$
05 0546 482 RSB
0547 483 10$:
50 50 52 51 C1 0547 484 ADDL3 R1,R2,R0 ; Address at end of buffer
00009000 8F D1 054B 485 CMPL #^X09000,R0 ; make sure it was an allocated message ***
0A 18 0552 486 BGEQ 11$
3E BB 0554 487 PUSHR #^M<R1,R2,R3,R4,R5>
60 FBBE CF 10 28 0556 488 MOV C3 #16,dbg$l_zeros,(r0) ; zero
3E BA 055C 489 POPR #^M<R1,R2,R3,R4,R5>
055E 490 11$:
55 51 DO 055E 491 MOVL R1,R5 ; save length
FEA5 30 0561 492 bsbw open_trace ; check for open log file
0564 493 :
0564 494 : Output header
0564 495 :
0564 496
0564 497 100$:
54 0000025C'EF 9E 0564 498 MOVAB logio_rab,R4 ; address of rab
28 A4 000002BB'EF 9E 056B 499 MOVAB dbg$outbuf+8,rab$l_rbf(R4) ; set buffer address
0573 500
27 00000000'EF E9 0573 501 blbc cterm_flag,111$ ; branch if VAX
057A 502
50 50 62 9A 057A 503 movzbl prototype(R2),R0 ; type
51 FDE6 CF 9E 057D 504 movab msg_type_table,R1 ; table
6140 DD 0582 505 PUSHL (R1)(R0) ; type
50 FDDA CF 9E 0585 506 movab null_line,10
09 62 91 058A 507 cmpb prototype(r2),#pro$c_data ; cterm data?
OD 12 058D 508 bneq 105$
058F 509
50 04 A2 9A 058F 510 movzbl msgtype(r2),R0 ; byte of message type
51 FC65 CF 9E 0593 511 movab message_table,R1 ; table of message types
50 6140 DO 0598 512 movl (r1)(r0),R0 ; address of message type string
059C 513 105$:
51 8ED0 059C 514 POPL R1
OA 11 059F 515 BRB 115$
05A1 516
50 FDBE CF 9E 05A1 517 111$: movab null_line,R0
51 FDB9 CF 9E 05A6 518 movab null_line,R1
05AB 519
05AB 520 115$: $FAO_S ctrstr = header_ctr,-
05AB 521 outbuf = dbg$outbuf,-
05AB 522 outlen = rab$w_rsz(R4),- ; length
05AB 523 P1 = R3,- ; to/from net

```

```

05AB 524 P2 = R1,- ; foundation type
05AB 525 P3 = R0,- ; message type if cterm
05AB 526 P4 = R5 ; Length
4E 50 E9 05C7 527 BLBC r0,250$
05CA 528
05CA 529 $PUT rab = r4
42 50 E9 05D3 530 blbc r0,250$ ; exit on error
05D6 531 ;
05D6 532 ; output buffer
05D6 533 ;
05D6 534 10$:
05D6 535 $FAO_S Ctrstr = mem_ctr,-
05D6 536 outbuf = dbg$outbuf,-
05D6 537 outlen = rab$w_rsz(R4),- ; length
05D6 538 p1 = 12(R2),- ; contents !XL
05D6 539 p2 = 8(R2),-
05D6 540 p3 = 4(R2),-
05D6 541 p4 = (R2),-
05D6 542 p5 = #16,- ; Length !AF, PART 1
05D6 543 p6 = R2,- ; address !AF, PART 2
05D6 544 p7 = R2 ; address (addr)
1A 50 E9 05FB 545 BLBC R0,250$ ; ERROR?
05FE 546
05FE 547 $PUT rab = logio_rab
0A 50 E9 060B 548 blbc r0,250$
060E 549
52 10 A2 9E 060E 550 MOVAB 16(R2),R2 ; bump address by longword
55 10 C2 0612 551 subl2 #16,R5 ; subtract from length
BF 14 0615 552 bgtr 110$ ; loop if more
0617 553 200$:
05 0617 554 RSB
0618 555 250$:
0618 556 QUIT
DB 11 063D 557 brb 200$
063F 558
063F 559 LOG_ASCIC::
063F 560 ; R0 -> points to ascic string to write to log
063F 561
063F 562
01 0000206'EF 01 E1 063F 563 BBC #v_errcr,trace_flag,10$
05 0647 564 RSB
0648 565 10$:
FDBE 30 0648 566 bsbw open_trace ; open file if needed
51 DD 0649 567 pushl R1 ; save
000025C'EF 9E 064D 568 movab logio_rab,r1 ; rab
22 A1 80 9B 0654 569 movzbw (r0)+,rab$w_rsz(R1) ; size
28 A1 50 D0 0658 570 movl r0,rab$l_rbf(R1) ; address
51 8ED0 065C 571 $put rab=(R1) ; put output
05 0665 572 popl R1 ; restore
0668 573 rsb ; exit
0669 574
0669 575 .end

```

RTLOGIO
Symbol table

- CTERM LOG IO

N 13

16-SEP-1984 02:13:04 VAX/VMS Macro V04-00
5-SEP-1984 03:15:40 [RTPAD.SRC]RTLOGIO.MAR;1

Page 18
(13)

```

$$ .TAB = 0000025C R 02
$$ .TABEND = 000002A0 R 02
$$ .TMP = 00000000
$$ .TMP1 = 00000001
$$ .TMP2 = 00000061
$$ .TMPX = 00000000 R 03
$$ .TMPX1 = 0000000B
$$ T1 = 00000001
$$ T2 = 0000000A
CTERMSCLOSE_LOG = 000000C7 RG 04
CTERMSLOG_IO = 00000000 RG 04
CTERMSOPEN_LOG = 00000052 RG 04
CTERM_FAB = 00000000 R 02
CTERM_FLAG = ***** X 04
CTERM_NAM = 00000094 R 02
CTERM_RAB = 00000050 R 02
CTPSB_MSGTYPE = 0000002A
CTPSB_PRO_MSGTYPE = 00000026
DBGSLINKRCV = 0000012A RG 04
DBGSLINKWRITE = 00000145 RG 04
DBGSL_ZEROS = 00000119 R 04
DBGSOOTBUF = 000002B3 R 02
DBGSTRACE_IO = 0000053E RG 04
DEBUG_AST = 00000524 RG 04
DEBUG_BUF = 00000202 R 02
DEBUG_CHAN = 00000200 R 02
DEBUG_MASK = 00000111 R 04
DEBUG_MSG = 000000EE R 04
DEBUG_SETUP = 000004CB RG 04
FABS_BNS = 00000035
FABS_BNS = 00000034
FABS_BID = 00000003
FABS_BLN = 00000050
FABS_SEQ = 00000000
FABS_VAR = 00000002
FABS_ALO = 00000010
FABS_CTX = 00000018
FABS_DNA = 00000030
FABS_FNA = 0000002C
FABS_FOP = 00000004
FABS_CHAN_MODE = 00000002
FABS_CR = 00000001
FABS_FILE_MODE = 00000004
FABS_LNM_MODE = 00000000
FABS_PUT = 00000000
FABS_SQO = 00000006
FABS_GBC = 00000048
FILESPEC = 000000F4 R 02
FILE_NAME = 0000031F R 02
FILE_STR = 000000FC R 02
FLGSM_LOGGING = 00000008
FLGSV_LOGGING = 00000003
GOTVM = 00000160 RG 04
HEADER_CTR = 00000180 R 04
IOSM_OUTBAND = ***** X 04
IOS_SETMODE = ***** X 04
LIB$PUT_OUTPUT = ***** X 04

```

```

LIB$SIGNAL ***** X 04
LOGCTR = 000001C9 R 04
LOGIO_FAB = 0000020C R 02
LOGIO_RAB = 0000025C R 02
LOG_ASCIC = 0000063F RG 04
LOG_A_EXT = 000001FB R 02
LOG_S_EXT = = 00000004
MEM_CTR = 000001A2 R 04
MESSAGE_TABLE = 000001FC R 04
MSGTYPE = 00000004
MSG_TYPE_TABLE = 00000367 R 04
M_ERROR = 00000002
M_OPEN = 00000001
NAM$B_ESL = 0000000B
NAM$B_ESS = 0000000A
NAM$B_NOP = 00000008
NAM$B_RSS = 00000002
NAM$C_BID = 000000C0
NAM$C_BLN = 00000060
NAM$C_MA = 000000FF
NAM$E_ESA = 0000000C
NAM$E_RSA = 00000004
NULL_CINE = 00000363 R 04
OPEN_TRACE = 00000409 R 04
PROSC_DATA = 00000009
PROTYPE = 00000000
RAB$B_RAC = 0000001E
RAB$C_BID = 00000001
RAB$C_BLN = 00000044
RAB$C_SEQ = 00000000
RAB$E_CTX = 00000018
RAB$E_RBF = 00000028
RAB$E_RBP = 00000004
RAB$V_WBH = 0000000A
RAB$W_RSZ = 00000022
RECORD_QUIT ***** X 04
REMS_FACILITY ***** X 04
RETSTATUS ***** X 04
RTLOG$WRITE_STRING = 000000E0 RG 04
RTPAD_DBG = 000002A0 R 02
SAVE_LOG = 00000208 R 02
SHRS_OPENOUT ***** X 04
SHRS_WRITEERR ***** X 04
SS$BREAK ***** X 04
SS$MUSTCLOSEFL ***** X 04
SS$NORMAL ***** X 04
SYS$ASSIGN ***** GX 04
SYS$CLOSE ***** GX 04
SYS$CONNECT ***** GX 04
SYS$CREATE ***** GX 04
SYS$FAO ***** X 04
SYS$PUT ***** GX 04
SYS$QIO ***** GX 04
SYS$SETAST ***** GX 04
SYS$TRNLOG ***** GX 04
SYS$WAKE ***** GX 04
TRACE_FLAG = 00000206 R 02

```

R
V

RTLOGIO
Symbol table

- CTERM LOG IO

B 14

16-SEP-1984 02:13:04 VAX/VMS Macro V04-00
5-SEP-1984 03:15:40 [RTPAD.SRC]RTLOGIO.MAR;1

Page 19
(13)

RT
VO

TTYDESC	*****	X	04
UTIL\$REPORT_IO_ERROR	*****	X	04
V_ERROR	= 00000001		
V_OPEN	= 00000000		
WAKEFLAG	*****	X	04

! 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
RTPAD	000003AB (939.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG
\$RMSNAM	0000000B (11.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC B:TE
RTPAD	00000669 (1641.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.03	00:00:01.79
Command processing	107	00:00:00.44	00:00:02.18
Pass 1	350	00:00:08.05	00:00:35.64
Symbol table sort	0	00:00:00.99	00:00:02.97
Pass 2	115	00:00:01.74	00:00:05.08
Symbol table output	14	00:00:00.07	00:00:00.31
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	619	00:00:11.35	00:00:47.99

The working set limit was 1500 pages.
64725 bytes (127 pages) of virtual memory were used to buffer the intermediate code.
There were 60 pages of symbol table space allocated to hold 994 non-local and 53 local symbols.
575 source lines were read in Pass 1, producing 32 object records in Pass 2.
39 pages of virtual memory were used to define 32 macros.

! Macro library statistics !

Macro library name	Macros defined
-\$255\$DUA28:[RTPAD.OBJ]RTPAD.MLB;1	3
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	0
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	26
TOTALS (all libraries)	29

1317 GETS were required to define 29 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:RTLOGIO/OBJ=OBJ\$:RTLOGIO MSRC\$:RTLOGIO/UPDATE=(ENH\$:RTLOGIO)+EXECMLS/LIB+LIB\$:RTPAD/LIB

