


```
EEEEEEEEEE MM MM SSSSSSSS GGGGGGGG
EEEEEEEEEE MM MM SSSSSSSS GGGGGGGG
EE MMMM MMMM SS GG
EE MMMM MMMM SS GG
EE MM MM MM SS GG
EE MM MM MM SS GG
EEEEEEEEEE MM MM SSSSSS GG GGGGGG
EEEEEEEEEE MM MM SSSSSS GG GGGGGG
EE MM MM SS GG GGGGGG
EE MM MM SS GG GGGGGG
EE MM MM SS GG GG
EE MM MM SS GG GG
EEEEEEEEEE MM MM SSSSSSSS GGGGGG
EEEEEEEEEE MM MM SSSSSSSS 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
LLLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLLL IIIIII SSSSSSSS
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42

```

0001 0 %TITLE 'VMS erro' message handler'
0002 0 MODULE EMSG (IDENT = 'V04-000',
0003 0 ADDRESSING_MODE(EXTERNAL=LONG_RELATIVE, NONEXTERNAL=LONG_RELATIVE)
0004 0 ) =
0005 1 BEGIN
0006 1
0007 1
0008 1 *****
0009 1 *
0010 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0011 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0012 1 * ALL RIGHTS RESERVED.
0013 1 *
0014 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0015 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0016 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0017 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0018 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0019 1 * TRANSFERRED.
0020 1 *
0021 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0022 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0023 1 * CORPORATION.
0024 1 *
0025 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0026 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0027 1 *
0028 1 *
0029 1 *****
0030 1
0031 1
0032 1 ++
0033 1 FACILITY: DSR (Digital Standard RUNOFF) / DSRPLUS
0034 1
0035 1 ABSTRACT:
0036 1 Handle requests for an error to be reported to the user.
0037 1
0038 1 ENVIRONMENT:
0039 1 VAX/VMS user mode
0040 1
0041 1 AUTHOR: Ray Marshall , CREATION DATE: 5-October-1981
0042 1

```

```
.. 44 0043 1 %SBTTL 'Revision History'
.. 45 0044 1
.. 46 0045 1 MODIFIED BY:
.. 47 0046 1
.. 48 0047 1 008 KFA00008 Ken Alden 13-Jul-1983
.. 49 0048 1 Removed rnfpc5.
.. 50 0049 1
.. 51 0050 1 007 KFA00007 Ken Alden 8-Jul-1983
.. 52 0051 1 Added rnfpc6 to the list of informationals
.. 53 0052 1
.. 54 0053 1 006 KFA00006 Ken Alden 9-May-1983
.. 55 0054 1 Suppress output to the mem file if closed.
.. 56 0055 1
.. 57 0056 1 005 KAD00005 Keith Dawson 3-May-1983
.. 58 0057 1 Suppressed ident, severity, ... for RNFCR1 and CR2.
.. 59 0058 1
.. 60 0059 1 004 KAD00004 Keith Dawson 12-April-1983
.. 61 0060 1 Added support for new termination error messages for
.. 62 0061 1 information written to .BRN file. This involved adding
.. 63 0062 1 messages RNFC2, PC3, PC4, PC5.
.. 64 0063 1
.. 65 0064 1 003 KAD00003 Keith Dawson 07-Mar-1983
.. 66 0065 1 Global edit of all modules. Updated module names, idents,
.. 67 0066 1 copyright dates. Changed require files to BLISS library.
.. 68 0067 1
.. 69 0068 1 --
```

```

71 0069 1 %SBTTL 'Module Level Declarations'
72 0070 1
73 0071 1 : TABLE OF CONTENTS:
74 0072 1
75 0073 1
76 0074 1 FORWARD ROUTINE
77 0075 1     MSG,           ! Send message to system which will invoke EMSGA
78 0076 1     EMSGA;       ! Determine where to write message & make it happen
79 0077 1
80 0078 1
81 0079 1 : INCLUDE FILES:
82 0080 1
83 0081 1 LIBRARY 'SYS$LIBRARY:STARLET'; ! System macros
84 0082 1
85 0083 1 LIBRARY 'NXPORT:XPORT';       ! XPORT Library
86 0084 1 REQUIRE 'REQ:RNODEF';       ! RUNOFF variant definitions
87 0215 1
88 0216 1 undeclare %quote $descriptor;
89 0217 1
90 U 0218 1 %IF DSRPLUS %THEN
91 U 0219 1 LIBRARY 'REQ:DPLLIB';       ! DSRPLUS BLISS Library
92 0220 1 %ELSE
93 0221 1 LIBRARY 'REQ:DSRLIB';       ! DSR BLISS Library
94 0222 1 %FI
95 0223 1
96 0224 1
97 0225 1 : MACROS:
98 0226 1
99 0227 1 MACRO
100 0228 1     L = 0,32,0 %;           ! Longword reference
101 0229 1     W0 = 0,16,0 %;      ! Word reference (low order word of longword)
102 0230 1     W1 = 16,16,0 %;   ! Word reference (high order word of longword)
103 0231 1
104 0232 1
105 0233 1 : EQUATED SYMBOLS:
106 0234 1
107 0235 1 LITERAL
108 0236 1     CR = %'15';           ! <return>
109 0237 1     LF = %'12';       ! <linefeed>
110 0238 1 : None (for now)
111 0239 1
112 0240 1 : OWN STORAGE:
113 0241 1
114 0242 1 : None (for now)
115 0243 1
116 0244 1 : EXTERNAL REFERENCES:
117 0245 1
118 0246 1 : Messages that shouldn't output the facility, severity, and error codes.
119 0247 1
120 0248 1 EXTERNAL LITERAL
121 0249 1     RNFBK, ! See command on input line %C of page %I of file %S
122 0250 1     RNFERD, ! DIGITAL Standard Runoff Version %V: %N diagnostic messages reported
123 0251 1     RNFERR, ! DIGITAL Standard Runoff Version %V: %N diagnostic message reported
124 0252 1     RNFLOC, ! on output page %P; on input line %C of page %I of file %F
125 0253 1     RNFNED, ! DIGITAL Standard Runoff Version %V: No errors detected
126 0254 1     RNFPCT, ! %N page%X written to %O
127 0255 1     RNFP2, ! %N indexing record%X written to %B

```

```
: 128      0256 1      RNFC3, ! %N table-of-contents recordXX written to %B
: 129      U 0257 1 %IF DSRPLUS %THEN
: 130      U 0258 1      RNFC4, ! %N cross-reference recordXX written to %B
: 131      U 0259 1      RNFC5, ! %N cross-reference recordXX read from %A
: 132      U 0260 1      RNFCR1, ! Run DSRPLUS again if you need them correct
: 133      U 0261 1      RNFCR2, ! Or run DSRPLUS/DEBUG to see which one(s) changed
: 134      0262 1 %FI
: 135      0263 1      RNFCR3, ! %S (Used to output diagnostic message text.)
: 136      0264 1
: 137      0265 1 EXTERNAL
: 138      0266 1      GCA: GCA_DEFINITION,
: 139      0267 1      RNOIOB: REF_$XPO_IOB(); !Output file (document being built)
: 140      0268 1
```

```

142 0269 1 %SBTTL 'EMSG -- Routine declaration'
143 0270 1 GLOBAL ROUTINE EMSG (e_code,e_string,no_mem_out) =
144 0271 1
145 0272 1 !++
146 0273 1
147 0274 1 FUNCTIONAL DESCRIPTION:
148 0275 1
149 0276 1     Given the error code and message string, build an appropriate error
150 0277 1     message vector. Then make an appropriate call to the VMS system
151 0278 1     service $PUTMSG with the vector as a parameter. Based on the value of
152 0279 1     NO MEM OUT, the call either does or does NOT specify the action routine
153 0280 1     EMSGA (for Error MeSSaGe Action routine) which will determine whether
154 0281 1     or not to send the message to the .MEM file.
155 0282 1
156 0283 1     The action routine must be used because it isn't till then that the
157 0284 1     whole message has been constructed. $PUTMSG will add in the facility,
158 0285 1     severity code, and error code as a prefix to the text we send to it.
159 0286 1
160 0287 1 FORMAL PARAMETERS:
161 0288 1
162 0289 1     E_CODE           Number of error.
163 0290 1     E_STRING        Descriptor pointing to error string.
164 0291 1     NO_MEM_OUT      True if output should NOT go to the mem file.
165 0292 1
166 0293 1 IMPLICIT INPUTS:
167 0294 1
168 0295 1     None
169 0296 1
170 0297 1 IMPLICIT OUTPUTS:
171 0298 1
172 0299 1     None
173 0300 1
174 0301 1 ROUTINE VALUE:
175 0302 1 COMPLETION CODES:
176 0303 1
177 0304 1     Return status from the VMS $PUTMSG call.
178 0305 1
179 0306 1 SIDE EFFECTS:
180 0307 1
181 0308 1     None
182 0309 1
183 0310 1 --
184 0311 1
185 0312 2 BEGIN
186 0313 2 LOCAL
187 0314 2     msgdesc: BLOCK[5];           ! Descriptor for message argument vector
188 0315 2
189 0316 2     msgdesc[0,W0_] = 3;         ! Always passing one argument, the message text
190 0317 2
191 0318 2 ! Determine whether to print just the text or the whole message.
192 0319 2
193 0320 2 IF (
194 0321 2     OR .E_CODE EQL RNFBK
195 0322 2     OR .E_CODE EQL RNFERD
196 0323 2     OR .E_CODE EQL RNFERR
197 0324 2     OR .E_CODE EQL RNFLOC
198 0325 2     OR .E_CODE EQL RNFNED
198 0325 2     OR .E_CODE EQL RNFPCT

```

```

199      0326      3      OR   .E_CODE EQL RNFPC2
200      0327      3      OR   .E_CODE EQL RNFPC3
201      0328      3      %IF DSRPLUS %TREN
202      0329      3      OR   .E_CODE EQL RNFPC4
203      0330      3      OR   .E_CODE EQL RNFPC5
204      0331      3      OR   .E_CODE EQL RNFCR1
205      0332      3      OR   .E_CODE EQL RNFCR2
206      0333      3      %FI
207      0334      3      OR   .E_CODE EQL RNFSTR )
208      0335      3      THEN
209      0336      3      msgdesc[0,w1_] = 1 ! If any of above, only print text of message!
210      0337      3      ELSE
211      0338      3      msgdesc[0,w1_] = 0; ! Else, default to printing all of message
212      0339      3
213      0340      3      msgdesc[1,l_] = .e_code; ! Error message number.
214      0341      3      msgdesc[2,w0_] = 1; ! There is one FAO arg. -- the message text.
215      0342      3      msgdesc[2,w1_] = 0; ! No new message flags.
216      0343      3      msgdesc[3,l_] = .e_string; ! Pointer to descriptor for message text.
217      0344      3
218      0345      3      ! Send message to VMS:
219      0346      3
220      0347      3      IF NOT .no_mem_out
221      0348      3      THEN ! if mem file shouldn't get message,
222      0349      3      $PUTMSG (msgvec=msgdesc) ! don't specify an action routine
223      0350      3      ELSE
224      0351      3      IF NOT .gca_skip_out
225      0352      3      THEN ! otherwise, action routine will see to
226      0353      3      $PUTMSG (msgvec=msgdesc,actrtn=EMSGA) ! it that the .MEM file get's it.
227      0354      3      ELSE
228      0355      3      0
229      0356      3      END; ! End of routine EMSG

```

.TITLE EMSG VMS error message handler
.IDENT \V04-000\

.EXTRN RNFBK, RNFERD, RNFERR
.EXTRN RNFLOC, RNFNED, RNFPC2
.EXTRN RNFPC3, RNFSTR
.EXTRN GCA, RNOIOB, SYSS\$PUTMSG

.PSECT \$CODE\$,NOWRT,2

		0000	00000	.ENTRY	EMSG, Save nothing	: 0270
	5E	14	C2 00002	SUBL2	#20, SP	
	6E	03	B0 00005	MOVW	#3, MSGDESC	: 0316
	50	AC	D0 00008	MOVL	E_CODE, R0	: 0320
00000000G	8F	50	D1 0000C	CML	R0, #RNFBK	
		48	13 00013	BEQL	1\$	
00000000G	8F	50	D1 00015	CML	R0, #RNFERD	: 0321
		3F	13 0001C	BEQL	1\$	
00000000G	8F	50	D1 0001E	CML	R0, #RNFERR	: 0322
		36	13 00025	BEQL	1\$	
00000000G	8F	50	D1 00027	CML	R0, #RNFLOC	: 0323
		2D	13 0002E	BEQL	1\$	
00000000G	8F	50	D1 00030	CML	R0, #RNFNED	: 0324
		24	13 00037	BEQL	1\$	

EMSG
V04-000

VMS error message handler
EMSG -- Routine declaration

F 15
16-Sep-1984 00:23:01
14-Sep-1984 13:06:07

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[RUNOFF.SRC]EMSG.B32;1

Page 7
(4)

EN
VO

00000000G	8F		50	D1	00039		CMPL	RC, #RNFPC1	:	0325	
			1B	13	00040		BEQL	1\$:		
00000000G	8F		50	D1	00042		CMPL	RO, #RNFPC2	:	0326	
			12	13	00049		BEQL	1\$:		
00000000G	8F		50	D1	00048		CMPL	RO, #RNFPC3	:	0327	
			09	13	00052		BEQL	1\$:		
00000000G	8F		50	D1	00054		CMPL	RO, #RNFSTR	:	0334	
			06	12	0005B		BNEQ	2\$:		
	02	AE	01	B0	0005D	1\$:	MOVW	#1, MSGDESC+2	:	0336	
			03	11	00061		BRB	3\$:		
			02	AE	B4	00063	2\$:	CLRW	MSGDESC+2	:	0338
	04	AE	50	D0	00066	3\$:	MOVL	RO, MSGDESC+4	:	0340	
	08	AE	01	D0	0006A		MOVL	#1, MSGDESC+8	:	0341	
	0C	AE	08	AC	D0	0006E	MOVL	E STRING, MSGDESC+12	:	0343	
		06	0C	AC	E8	00073	BLBS	NO MEM_OUT, 4\$:	0347	
			7E	7C	00077		CLRW	-(SP)	:	0349	
			7E	D4	00079		CLRL	-(SP)	:		
			0F	11	0007B		BRB	5\$:		
		13	00000000G	EF	E8	0007D	4\$:	BLBS	GCA+112, 6\$:	0351
				7E	7C	00084		CLRW	-(SP)	:	0353
		00000000V		EF	9F	00086	PUSHAB	EMSGA	:		
			0C	AE	9F	0008C	5\$:	PUSHAB	MSGDESC	:	
00000000G	00		04	FB	0008F		CALLS	#4, SYS\$PUTMSG	:		
				04	00096		RET		:		
			50	D4	00097	6\$:	CLRL	RO	:	0351	
				04	00099		RET		:	0356	

; Routine Size: 154 bytes, Routine Base: \$CODE\$ + 0000

00

```

: 231 0357 1 %SBITL 'EMSGA -- Action routine for all messages'
: 232 0358 1 GLOBAL ROUTINE EMSGA (full_message_p) =
: 233 0359 1
: 234 0360 1  +-+
: 235 0361 1
: 236 0362 1  FUNCTIONAL DESCRIPTION:
: 237 0363 1
: 238 0364 1      Intercept message before VMS outputs it. Determine just where it
: 239 0365 1      should be written, and see to it that it goes there.
: 240 0366 1
: 241 0367 1  FORMAL PARAMETERS:
: 242 0368 1
: 243 0369 1      FULL_MESSAGE_P  A pointer to the message text as expanded by VMS.
: 244 0370 1
: 245 0371 1  IMPLICIT INPUTS:
: 246 0372 1
: 247 0373 1      None
: 248 0374 1
: 249 0375 1  IMPLICIT OUTPUTS:
: 250 0376 1
: 251 0377 1      None
: 252 0378 1
: 253 0379 1  ROUTINE VALUE:
: 254 0380 1  COMPLETION CODES:
: 255 0381 1
: 256 0382 1      FALSE = 0 -- directs VMS NOT to send the message to SYS$ERROR.
: 257 0383 1      TRUE  = 1 -- directs VMS to output the message.
: 258 0384 1
: 259 0385 1  SIDE EFFECTS:
: 260 0386 1
: 261 0387 1      None
: 262 0388 1
: 263 0389 1  --
: 264 0390 1
: 265 0391 2  BEGIN
: 266 0392 2  LOCAL
: 267 0393 2      hack: $STR_DESCRIPTOR(),
: 268 0394 2      STATUS;
: 269 0395 2  MAP
: 270 0396 2      full_message_p: REF $STR_DESCRIPTOR();
: 271 0397 2
: 272 P 0398 2  $STR_DESC_INIT(DESCRIPTOR=HACK,STRING=(.full_message_p[str$h_length],
: 273 0399 2      .full_message_p[str$a_pointer]));
: 274 0400 2
: 275 0401 2  !The line is now packed correctly, so output it to the requested places.
: 276 0402 2  CASE .GCA_ERR_DIR FROM REPORT_ERR_NONE TO REPORT_ERR_BOTH OF
: 277 0403 2  SET
: 278 0404 2  [REPORT_ERR_NONE]:
: 279 0405 2      BEGIN
: 280 0406 2      STATUS = FALSE; !Don't bother to output the message
: 281 0407 2      END;
: 282 0408 2
: 283 0409 2  [REPORT_ERR_FILE]:
: 284 0410 2      BEGIN
: 285 0411 2      !Report error in output file
: 286 0412 2      STATUS = $XPO_PUT( IOB = .RNOIOB, STRING = hack );
: 287 0413 2      !Add carriage control information to the end of the error message

```

288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314

```

P 0414 3      STATUS = $XPO_PUT( IOB = .RNOIOB,
P 0415      STRING = (2, CH$PTR(UPLIT (%STRING (%CHAR(CR,LF))))))
0416      );
0417      STATUS = FALSE;
0418      END;
0419
[REPORT_ERR_STD]:
0420      BEGIN
0421      !Report error on standard error log
0422      STATUS = TRUE;
0423      END;
0424
[REPORT_ERR_BOTH]:
0425      !Report error both places
0426      BEGIN
0427      !Report error in output file
0428      STATUS = $XPO_PUT( IOB = .RNOIOB, STRING = hack ),
0429      !Add carriage control information to the end of the error message
0430      STATUS = $XPO_PUT( IOB = .RNOIOB,
P 0431      STRING = (2, CH$PTR(UPLIT (%STRING (%CHAR(CR,LF))))))
P 0432      );
0433      STATUS = TRUE;
0434      END;
0435      TES;
0436      .STATUS
0437      END;
0438      ! End of EMSGA
0439
0440      1
    
```

						PSECT	\$SPLITS,NOWRT,NOEXE,2	
		00	00	0A	0D	00000	P.AAD: .ASCII	<13><10><0><0>
		00	00	0A	0D	00004	P.AAH: .ASCII	<13><10><0><0>
						.EXTRN	XPOS\$PUT, XPOS\$FAILURE	
						.PSECT	\$CODE\$,NOWRT,2	
					001C 00000	.ENTRY	EMSGA, Save R2,R3,R4	0358
	54	00000000G	EF	9E	00002	MOVAB	XPOS\$PUT, R4	
	53	00000000G	EF	9E	00009	MOVAB	XPOS\$FAILURE, R3	
	52	00000000G	EF	9E	00010	MOVAB	RNOIOB, R2	
	5E		10	C2	00017	SUBL2	#16, SP	
	50	04	AC	D0	0001A	MOVL	FULL_MESSAGE P, R0	0399
	08	AE	04	60	0001E	MOVW	(R0), \$STR\$DESC	
	0A	AE	010E	8F	00022	MOVW	#270, \$STR\$DESC+2	
	0C	AE	04	A0	00028	MOVL	4(R0), \$STR\$DESC+4	
0049		00	00000000G	EF	CF	C002D	CASEL	GCA+200, #0, #3
	0085	000A		0046	00035	1\$:	.WORD	0402
							3\$-1\$,-	
							2\$-1\$,-	
							5\$-1\$,-	
							4\$-1\$	
							3\$	
						BRB		0406
	44	51		3C	11	0003D	MOVW	RNOIOB, R1
	2C	A1	08	62	D0	0003F	2\$:	0412
							MOVAB	\$IOB\$OUTPUT, 68(R1)
							MOVW	#7, 44(R1)

			53 DD 0004B	PUSHL R3	
			7E D4 0004D	CLRL -(SP)	
			51 DD 0004F	PUSHL R1	
	64		03 FB 00051	CALLS #3, XPO\$PUT	
	51		62 D0 00054	MOVL RNOIOB, R1	0416
	6E		02 B0 00057	MOVW #2, \$IOB\$OUTPUT	
02	AE		0E 90 0005A	MOVB #14, \$IOB\$OUTPUT+2	
03	AE		01 90 0005E	MOVB #1, \$IOB\$OUTPUT+3	
04	AE	'00000000'	EF 9E 00062	MOVAB P.AAD, \$IOB\$OUTPUT+4	
44	A1		6E 9E 0006A	MOVAB \$IOB\$OUTPUT, 68(R1)	
2C	A1		07 90 0006E	MOVB #7, 44(R1)	
			53 DD 00072	PUSHL R3	
			7E D4 00074	CLRL -(SP)	
			51 DD 00076	PUSHL R1	
	64		03 FB 00078	CALLS #3, XPO\$PUT	
			50 D4 0007B 3\$:	CLRL STATUS	0417
			04 0007D	RET	0402
	51		62 D0 0007E 4\$:	MOVL RNOIOB, R1	0430
44	A1	08	AE 9E 00081	MOVAB \$IOB\$OUTPUT, 68(R1)	
2C	A1		07 90 00086	MOVB #7, 44(R1)	
			53 DD 0008A	PUSHL R3	
			7E D4 0008C	CLRL -(SP)	
			51 DD 0008E	PUSHL R1	
	64		03 FB 00090	CALLS #3, XPO\$PUT	
	51		62 D0 00093	MOVL RNOIOB, R1	0434
	6E		02 B0 00096	MOVW #2, \$IOB\$OUTPUT	
02	AE		0E 90 00099	MOVB #14, \$IOB\$OUTPUT+2	
03	AE		01 90 0009D	MOVB #1, \$IOB\$OUTPUT+3	
04	AE	'00000000'	EF 9E 000A1	MOVAB P.AAH, \$IOB\$OUTPUT+4	
44	A1		6E 9E 000A9	MOVAB \$IOB\$OUTPUT, 68(R1)	
2C	A1		07 90 000AD	MOVB #7, 44(R1)	
			53 DD 000B1	PUSHL R3	
			7E D4 000B3	CLRL -(SP)	
			51 DD 000B5	PUSHL R1	
	64		03 FB 000B7	CALLS #3, XPO\$PUT	
	50		01 D0 000BA 5\$:	MOVL #1, STATUS	0435
			04 000BD	RET	0440

: Routine Size: 190 bytes. Routine Base: \$CODE\$ + 009A

```

: 315      0441 1
: 316      0442 1 END           ! End of module
: 317      0443 0 ELUDOM

```

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	344	NOVEC,NOWRT, RD, EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
\$PLITS	8	NOVEC,NOWRT, RD, NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

EMSG
V04-000

VMS error message handler
EMSGA -- Action routine for all messages

J 15
16-Sep-1984 00:23:01
14-Sep-1984 13:06:07

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[RUNOFF.SRC]EMSG.B32;1 Page 11
(5)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
-\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	4	0	581	00:01.0
-\$255\$DUA28:[SYSLIB]XPORT.L32;1	590	114	19	252	00:00.5
-\$255\$DUA28:[RUNOFF.SRC]DSRLIB.L32;1	1248	11	0	86	00:01.0

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:EMSG/OBJ=OBJ\$:EMSG MSRC\$:EMSG/UPDATE=(ENHS\$:EMSG)

: Size: 344 code + 8 data bytes
: Run Time: 00:15.2
: Elapsed Time: 00:38.1
: Lines/CPU Min: 1748
: Lexemes/CPU-Min: 56214
: Memory Used: 150 pages
: Compilation Complete

0340 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

The image displays a grid of 100 small terminal window screenshots, arranged in 10 rows and 10 columns. Each window shows a different screen from the VAX/VMS V4.0 system. The screens contain various text-based data, including system status, error messages, and user prompts. Some screens are clearly labeled with text like "ELSEN LIS", "ENDCMT LIS", "ENDCHR LIS", "DSRMSG LIS", and "EMSG LIS". The overall appearance is that of a multi-processor system's output, possibly a diagnostic or test run.