


```

BBBBBBBB      AAAAAA      SSSSSSSS      MM      MM      SSSSSSSS      GGGGGGGG      GGGGGGGG      EEEEEEEEEE      NN      NN
BBBBBBBB      AAAAAA      SSSSSSSS      MM      MM      SSSSSSSS      GGGGGGGG      GGGGGGGG      EEEEEEEEEE      NN      NN
BB      BB      AA      AA      SS      MMMM      MMMM      SS      GG      GG      EE      NN      NN
BB      BB      AA      AA      SS      MM      MM      SS      GG      GG      EE      NN      NN
BB      BB      AA      AA      SS      MM      MM      SS      GG      GG      EE      NNNN      NN
BBBBBBBB      AA      AA      SSSSSS      MM      MM      SSSSSS      GG      GG      EE      NNNN      NN
BBBBBBBB      AA      AA      SSSSSS      MM      MM      SSSSSS      GG      GG      EE      NN      NN
BB      BB      AAAAAAAAAA      SS      MM      MM      SS      GG      GGGGGG      GG      GGGGGG      EE      NN      NNNN
BB      BB      AAAAAAAAAA      SS      MM      MM      SS      GG      GGGGGG      GG      GGGGGG      EE      NN      NNNN
BB      BB      AA      AA      SS      MM      MM      SS      GG      GG      GG      NN      NN
BB      BB      AA      AA      SS      MM      MM      SS      GG      GG      GG      NN      NN
BBBBBBBB      AA      AA      SSSSSSSS      MM      MM      SSSSSSSS      GGGGGG      GGGGGG      EEEEEEEEEE      NN      NN
BBBBBBBB      AA      AA      SSSSSSSS      MM      MM      SSSSSSSS      GGGGGG      GGGGGG      EEEEEEEEEE      NN      NN

```

```

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
LLLLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLLLL      IIIIII      SSSSSSSS

```

```

1 0001 0 %TITLE 'BASMSG - write BASMSG.MSG'
2 0002 0 MODULE BASMSG (           ! Write BASMSG.MSG
3 0003 0             IDENT = '2-003', ! File: BASMSG.B32
4 0004 0             MAIN = BASMSG   ! Edit: MDL2003
5 0005 0             ) =
6 0006 1 BEGIN
7 0007 1
8 0008 1 *****
9 0009 1 *
10 0010 1 *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
11 0011 1 *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
12 0012 1 *  ALL RIGHTS RESERVED.
13 0013 1 *
14 0014 1 *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
15 0015 1 *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
16 0016 1 *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
17 0017 1 *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
18 0018 1 *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
19 0019 1 *  TRANSFERRED.
20 0020 1 *
21 0021 1 *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
22 0022 1 *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
23 0023 1 *  CORPORATION.
24 0024 1 *
25 0025 1 *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
26 0026 1 *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
27 0027 1 *
28 0028 1 *
29 0029 1 *****
30 0030 1
31 0031 1
32 0032 1 **
33 0033 1 FACILITY:      BASIC
34 0034 1
35 0035 1 ABSTRACT:
36 0036 1
37 0037 1     This module, BASMSG.B32 is a Bliss32 program that writes
38 0038 1     the file BASMSG.MSG, which is read by the message compiler
39 0039 1     to produce BASIC's run-time messages.
40 0040 1
41 0041 1 ENVIRONMENT:  Runs at any access mode - AST reentrant
42 0042 1
43 0043 1 AUTHOR: John Sauter, CREATION DATE: 03-Nov-78
44 0044 1
45 0045 1 MODIFIED BY:
46 0046 1
47 0047 1 1-001 - Write BASMSG.MDL on LIB$: JBS 27-NOV-78
48 0048 1 1-002 - Change PC printed by ON CHAFIL to be hex. JBS 19-DEC-78
49 0049 1 1-003 - Add I/O List message. JBS 08-FEB-1979
50 0050 1 1-004 - Correct a typo in the I/O List message. JBS 09-FEB-1979
51 0051 1 1-005 - Write output on SRC$: JBS 21-MAR-1979
52 0052 1 1-006 - Correct spelling of I/O List message. JBS 20-APR-1979
53 0053 1 1-007 - Change the prefixes of the traceback messages to make
54 0054 1     them less confusing. JBS 20-APR-1979
55 0055 1 1-008 - Change the text of the traceback messages to refer to a SUB
56 0056 1     as a SUBPROGRAM, not a SUBROUTINE. JBS 07-MAY-1979
57 0057 1 1-009 - Add a message for showing the user PC and PSL at the time

```

```
: 58      0058 1  : of the error. JBS 11-MAY-1979
: 59      0059 1  : 1-010 - Correct a typo in edit 009. JBS 13-MAY-1979
: 60      0060 1  : 1-011 - Add an I/O error message which does not have a channel number,
: 61      0061 1  :         for the non-channel I/O statements. JBS 09-AUG-1979
: 62      0062 1  : 1-012 - Change I/O List to Immediate Mode. JBS 17-SEP-1979
: 63      0063 1  : 1-013 - Write output on LIBS. JBS 17-SEP-1979
: 64      0064 1  : 1-014 - Write output on SRCS, to conform to SBL's new procedure.
: 65      0065 1  :         JBS 20-SEP-1979
: 66      0066 1  : 1-015 - Allow up to 60-character message text. JBS 24-SEP-1979
: 67      0067 1  : 1-016 - convert program to generate .MSG file rather than .MDL
: 68      0068 1  :         file. MDL 23-Jun-1982
: 69      0069 1  : 2-001 - convert program to Bliss32; adapted from EDTMSG. MDL 25-Jun-1982
: 70      0070 1  : 2-002 - change to write BASMSG.TMP, which is compared with BASMSG.MSG
: 71      0071 1  :         for a consistency check. BLS 6-May-1983
: 72      0072 1  : 2-003 - facility name is BAS, not BASIC. MDL 30-Jun-1983
: 73      0073 1  :
: 74      0074 1  :
: 75      0075 1  : --
: 76      0076 1  :
```

```

: 78 0077 1 %SBTTL 'Declarations'
: 79 0078 1
: 80 0079 1 SWITCHES:
: 81 0080 1
: 82 0081 1
: 83 0082 1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
: 84 0083 1
: 85 0084 1
: 86 0085 1 LINKAGES:
: 87 0086 1
: 88 0087 1 NONE
: 89 0088 1
: 90 0089 1 TABLE OF CONTENTS:
: 91 0090 1
: 92 0091 1
: 93 0092 1 FORWARD ROUTINE
: 94 0093 1 BASMSG,
: 95 0094 1 WRITE_FILE,
: 96 0095 1 PRINT,
: 97 0096 1 HEX TEXT,
: 98 0097 1 PRINTABLE_TEXT;
: 99 0098 1
100 0099 1
101 0100 1 INCLUDE FILES:
102 0101 1
103 0102 1 LIBRARY 'SYSSLIBRARY:STARLET';
104 0103 1
105 0104 1 LITERAL
106 0105 1 BAS$K_FAC_NO = 26;
107 0106 1
108 0107 1
109 0108 1 MACROS:
110 0109 1
111 0110 1 NONE
112 0111 1
113 0112 1 EQUATED SYMBOLS:
114 0113 1
115 0114 1
116 0115 1
117 0116 1
118 0117 1 FIELDS:
119 0118 1
120 0119 1 NONE
121 0120 1
122 0121 1 STRUCTURES:
123 0122 1
124 0123 1 NONE
125 0124 1
126 0125 1 PSECTS:
127 0126 1
128 0127 1
129 0128 1 OWN STORAGE:
130 0129 1
131 0130 1 NONE
132 0131 1
133 0132 1 EXTERNAL REFERENCES:
134 0133 1

```

```

! Write BASMSG.TMP
! Actually write the text
! Print a line of text
! Convert binary to hexadecimal
! Convert binary to ASCII, printable

```

BASMSG
2-003

BASMSG - write BASMSG.MSG
Declarations

J 9
16-Sep-1984 01:42:08
14-Sep-1984 11:55:22

VAX-11 Bliss-32 V4.0-742
[BASRTL.SRC]BASMSGGEN.B32;1

Page 4
(2)

```
.. 135      0134 1
.. 136      0135 1 EXTERNAL ROUTINE
.. 137      0136 1   STRSCOPY_DX,
.. 138      0137 1   STRSCONCAT,
.. 139      0138 1   LIB$GET_INPUT,
.. 140      0139 1   STRSCOPY_R,
.. 141      0140 1   STR$FREET_DX,
.. 142      0141 1   ERRTXT;
.. 143      0142 1
```

```
! Copy a string, by descriptor
! Concatenate strings
! Get a line from SYSSINPUT
! Copy a string, by reference
! Free a dynamic string
! Return the text of a message
```

```

: 145      0143 1 %SBTTL 'Package of macros for string processing'
: 146      0144 1
: 147      0145 1 :- Macro to initialize a dynamic descriptor.
: 148      0146 1
: 149      0147 1
: 150      0148 1 MACRO
: 151      M 0149 1     INIT_DESCRIPTOR (DESCR) =
: 152      M 0150 1         DESCR [DSCSW_LENGTH] = 0;
: 153      M 0151 1         DESCR [DSCSB_DTYPE] = DSC$K_DTYPE_T;
: 154      M 0152 1         DESCR [DSCSB_CLASS] = DSC$K_CLASS_D;
: 155      M 0153 1         DESCR [DSCSA_POINTER] = 0;
: 156      0154 1 %
: 157      0155 1 :- <BLF/MACRO>
: 158      0156 1
: 159      0157 1 :- Macro to discard a dynamic descriptor.
: 160      0158 1
: 161      M 0159 1     DISCARD_DESCRIPTOR (DESCR) =
: 162      M 0160 1         BEGIN
: 163      M 0161 1
: 164      M 0162 1         LOCAL
: 165      M 0163 1             FREE_STATUS;
: 166      M 0164 1
: 167      M 0165 1         FREE_STATUS = STR$FREE1_DX (DESCR);
: 168      M 0166 1
: 169      M 0167 1         IF ( NOT .FREE_STATUS) THEN SIGNAL_STOP (.FREE_STATUS);
: 170      M 0168 1
: 171      M 0169 1         END;
: 172      0170 1 %
: 173      0171 1 :-
: 174      0172 1 :- Macro to build a text line using FAO. This is a convenience macro.
: 175      0173 1
: 176      M 0174 1     BUILD_TEXT_LINE (DESCR, CTL_STRING, FAO_ARGS) =
: 177      M 0175 1         BEGIN
: 178      M 0176 1
: 179      M 0177 1         LOCAL
: 180      M 0178 1             FAO_STATUS,
: 181      M 0179 1             COPY_STATUS;
: 182      M 0180 1
: 183      M 0181 1         CTL_STR_DSC [DSCSW_LENGTH] = %CHARCOUNT (CTL_STRING);
: 184      M 0182 1         CTL_STR_DSC [DSCSB_DTYPE] = DSC$K_DTYPE_T;
: 185      M 0183 1         CTL_STR_DSC [DSCSB_CLASS] = DSC$K_CLASS_S;
: 186      M 0184 1         CTL_STR_DSC [DSCSA_POINTER] = CH$PTR (UPLIT (CTL_STRING));
: 187      M 0185 1         FAO_STATUS = $FAO ?
: 188      M 0186 1             CTL_STR_DSC,
: 189      M 0187 1             OUT_LENGTH,
: 190      M 0188 1             TEMP_STR_DSC,
: 191      M 0189 1             %REMOVE (?FAO_ARGS));
: 192      M 0190 1
: 193      M 0191 1         IF ( NOT .FAO_STATUS) THEN SIGNAL_STOP (.FAO_STATUS);
: 194      M 0192 1
: 195      M 0193 1         COPY_STATUS = STR$COPY_R (DESCR, OUT_LENGTH, .TEMP_STR_DSC [DSCSA_POINTER]);
: 196      M 0194 1         .COPY_STATUS
: 197      M 0195 1         END
: 198      0196 1 %
: 199      0197 1 :-
: 200      0198 1 :- Macro to format and print a line. Errors are returned to the caller.
: 201      0199 1 :- This is a convenience macro.

```

```

: 202      0200  1  !-
: 203      0201  1  PRINT_LINE (TEXT, VARS) =
: 204      0202  1  BEGIN
: 205      0203  1
: 206      0204  1  LOCAL
: 207      0205  1  BUILD_STATUS,
: 208      0206  1  PRINT_STATUS;
: 209      0207  1
: 210      0208  1  BUILD_STATUS = BUILD_TEXT_LINE (LINE_DESC, %STRING (%REMOVE (TEXT)), VARS);
: 211      0209  1
: 212      0210  1  IF ( NOT .BUILD_STATUS) THEN RETURN (.BUILD_STATUS);
: 213      0211  1
: 214      0212  1  PRINT_STATUS = PRINT (.OUTPUT_RAB, LINE_DESC);
: 215      0213  1
: 216      0214  1  IF ( NOT .PRINT_STATUS) THEN RETURN (.PRINT_STATUS);
: 217      0215  1
: 218      0216  1  END
: 219      0217  1  %:
: 220      0218  1

```



```

222 0219 1 %SBTTL 'BAS$MSG - Write BASMSG.TMP'
223 0220 1 ROUTINE BAS$MSG ! Write BASMSG.TMP
224 0221 1 =
225 0222 1
226 0223 1 !++
227 0224 1 FUNCTIONAL DESCRIPTION:
228 0225 1
229 0226 1 This routine writes the file BASMSG.TMP.
230 0227 1
231 0228 1 CALLING SEQUENCE:
232 0229 1
233 0230 1 ret_status.wlc.v = BAS$MSG ()
234 0231 1
235 0232 1 FORMAL PARAMETERS:
236 0233 1
237 0234 1 NONE
238 0235 1
239 0236 1 IMPLICIT INPUTS:
240 0237 1
241 0238 1 NONE
242 0239 1
243 0240 1 IMPLICIT OUTPUTS:
244 0241 1
245 0242 1 NONE
246 0243 1
247 0244 1 COMPLETION STATUS:
248 0245 1
249 0246 1 $$$_NORMAL Normal successful completion
250 0247 1 Any error from LIB$GET_INPUT or STR$FREE1_DX
251 0248 1
252 0249 1 SIDE EFFECTS:
253 0250 1
254 0251 1 Writes a file.
255 0252 1 Any errors from RMS$CREATE, RMS$OPEN, RMS$CONNECT or RMS$CLOSE
256 0253 1 are signalled.
257 0254 1
258 0255 1 --
259 0256 1
260 0257 2 BEGIN
261 0258 2
262 0259 2 LOCAL
263 0260 2 OUTPUT_BUFFER : BLOCK [132, BYTE], ! output buffer, for RMS
264 0261 2 OUTPUT_FAB : $FAB_DECL, ! RMS FAB for the output file
265 0262 2 OUTPUT_NAM : $NAM_DECL, ! RMS NAM for the output file
266 0263 2 OUTPUT_RAB : $RAB_DECL, ! RMS RAB for the output file
267 0264 2 OUTPUT_FILE_NAME : BLOCK [8, BYTE], ! Name of output file
268 0265 2 OUTPUT_RESULT_NAME : BLOCK [NAM$C_MAXRSS, BYTE]; ! Place to store output file name
269 0266 2
270 0267 2
271 0268 2 OUTPUT_FILE_NAME [DSC$W_LENGTH] = %CHARCOUNT ('BASMSG');
272 0269 2 OUTPUT_FILE_NAME [DSC$B_DTYPE] = DSC$K_DTYPE_T;
273 0270 2 OUTPUT_FILE_NAME [DSC$B_CLASS] = DSC$K_CLASS_S;
274 0271 2 OUTPUT_FILE_NAME [DSC$A_POINTER] = UPLIT ('BASMSG');
275 0272 2 !+
276 0273 2 Initialize the FAB, NAM and RAB for the output file
277 0274 2 --
278 P 0275 2 $FAB_INIT (FAB = OUTPUT_FAB, !

```

```

279 P 0276 2 FAC = (PUT),
280 P 0277 2 FOP = (OFP, SQO, DFW),
281 P 0278 2 ORG = SEQ,
282 P 0279 2 SHR = NIL,
283 P 0280 2 MRS = 132,
284 P 0281 2 RAT = CR,
285 P 0282 2 RFM = VAR,
286 P 0283 2 FNA = .OUTPUT_FILE_NAME [DSC$A_POINTER],
287 P 0284 2 FNS = .OUTPUT_FILE_NAME [DSC$W_LENGTH],
288 P 0285 2 DNA = UPLIT ('SRC$:TMP'),
289 P 0286 2 DNS = %CHARCOUNT ('SRC$:TMP'),
290 P 0287 2 NAM = OUTPUT_NAME);
291 P 0288 2 $NAM_INIT (NAM = OUTPUT_NAME,
292 P 0289 2 RSA = OUTPUT_RESULT_NAME,
293 P 0290 2 RSS = NAM$C_MAXRSS);
294 P 0291 2 $RAB_INIT (RAB = OUTPUT_RAB,
295 P 0292 2 RAC = SEQ,
296 P 0293 2 ROP = WBH,
297 P 0294 2 USZ = 132,
298 P 0295 2 UBF = OUTPUT_BUFFER,
299 0296 2 FAB = OUTPUT_FAB);
300 0297 2
301 0298 2 + Create the output file, and do the $CONNECT.
302 0299 2 -
303 0300 2 BEGIN
304 0301 2 LOCAL
305 0302 2 CREATE_STATUS,
306 0303 2 CONNECT_STATUS;
307 0304 2
308 0305 2 CREATE_STATUS = $CREATE (FAB = OUTPUT_FAB);
309 0306 2
310 0307 2 IF ( NOT .CREATE_STATUS)
311 0308 2 THEN
312 0309 2 SIGNAL_STOP (
313 0310 2 SHR$_OPENOUT + (BAS$K_FAC_NO*65536) + STS$K_SEVERE,
314 0311 2 1,
315 0312 2 OUTPUT_FILE_NAME,
316 0313 2 .OUTPUT_FAB [FAB$L_STS], .OUTPUT_FAB [FAB$L_STV]);
317 0314 2
318 0315 2 CONNECT_STATUS = $CONNECT (RAB = OUTPUT_RAB);
319 0316 2
320 0317 2 IF ( NOT .CONNECT_STATUS)
321 0318 2 THEN
322 0319 2 SIGNAL_STOP (
323 0320 2 SHR$_OPENOUT + (BAS$K_FAC_NO*65536) + STS$K_SEVERE,
324 0321 2 1,
325 0322 2 OUTPUT_FILE_NAME,
326 0323 2 .OUTPUT_RAB [RAB$L_STS], .OUTPUT_RAB [RAB$L_STV]);
327 0324 2
328 0325 2 END;
329 0326 2
330 0327 2 + Point the file name descriptor to the resultant name string.
331 0328 2 -
332 0329 2
333 0330 2 OUTPUT_FILE_NAME [DSC$W_LENGTH] = .OUTPUT_NAME [NAM$B_RSL];
334 0331 2 OUTPUT_FILE_NAME [DSC$B_DTYPE] = DSC$K_DTYPE_T;
335 0332 2 OUTPUT_FILE_NAME [DSC$B_CLASS] = DSC$K_CLASS_S;

```

```

336 0333 2 OUTPUT_FILE_NAME [DSC$A_POINTER] = .OUTPUT_NAM [NAM$L_RSA];
337 0334 2
338 0335 2 IF ( NOT WRITE_FILE (OUTPUT_RAB))
339 0336 2 THEN
340 0337 2 SIGNAL STOP (
341 0338 2 SHR$WRITEERR + (BAS$K_FAC_NO*65536) + STS$K_SEVERE, !
342 0339 2 1, !
343 0340 2 OUTPUT_FILE_NAME, !
344 0341 2 .OUTPUT_RAB [RAB$L_STS], .OUTPUT_RAB [RAB$L_STV]);
345 0342 2
346 0343 2 BEGIN
347 0344 2 !+
348 0345 2 !- Close the output file.
349 0346 2
350 0347 2
351 0348 2 LOCAL
352 0349 2 CLOSE_STATUS;
353 0350 2
354 0351 2 CLOSE_STATUS = $CLOSE (FAB = OUTPUT_FAB);
355 0352 2
356 0353 2 IF ( NOT .CLOSE_STATUS)
357 0354 2 THEN
358 0355 2 SIGNAL STOP (
359 0356 2 SHR$CLOSEOUT + (BAS$K_FAC_NO*65536) + STS$K_SEVERE, !
360 0357 2 1, !
361 0358 2 OUTPUT_FILE_NAME, !
362 0359 2 .OUTPUT_FAB [FAB$L_STS], .OUTPUT_FAB [FAB$L_STV]);
363 0360 2
364 0361 2 END;
365 0362 2 RETURN (SS$NORMAL);
366 0363 2 END;

```

! End of routine BASMSG

.TITLE BASMSG BASMSG - write BASMSG.MSG
.IDENT \2-003\

.PSECT \$SPLITS,NOWRT,NOEXE,2

00	00	00	50	00	00	47	53	4D	53	41	42	00000	P.AAA:	.ASCII	\BASMSG\<0><0>	:
				4D	54	2E	3A	24	43	52	53	00008	P.AAB:	.ASCII	\SRCS:.TMP\<0><0><0>	:

.EXTRN STR\$COPY DX, STR\$CONCAT
.EXTRN LIB\$GET_INPUT, STR\$COPY_R
.EXTRN STR\$FREE1 DX, ERRTXT
.EXTRN SYSS\$CREATE, SYSS\$CONNECT
.EXTRN SYSS\$CLOSE

.PSECT \$CODE\$,NOWRT,2

0050	8F	00	0100	CE	010E0006	8F	00	0000E	00	2C	0001E	MOVAB	P.AAA, OUTPUT_FILE_NAME+4	0271
			0104	CE	0000'	CF	9E	00017	00	2C	0001E	MOVAB	#0, (SP), #0, #80, \$RMS_PTR	0287
			FF2C	CD	5003	8F	B0	00028				MOVW	#20483, \$RMS_PTR	:

007C 00000 BASMSG: .WORD Save R2,R3,R4,R5,R6

MOVAB LIB\$STOP, R6
MOVAB -640(SP), SP
MOVL #17694726, OUTPUT_FILE_NAME
MOVAB P.AAA, OUTPUT_FILE_NAME+4
MOVCS #0, (SP), #0, #80, \$RMS_PTR
MOVW #20483, \$RMS_PTR

: 0220
:
:
: 0268
: 0271
: 0287
:

			FF30	CD	20000060	8F	D0	0002F	MOVL	#536871008, \$RMS_PTR+4	
			FF42	CD	2001	8F	B0	00038	MOVW	#8193, \$RMS_PTR+22	
			FF49	CD	0200	8F	B0	0003F	MOVW	#512, \$RMS_PTR+29	
			FF4B	CD		02	90	00046	MOVB	#2, \$RMS_PTR+31	
			FF54	CD	FECC	CD	9E	0004B	MOVAB	OUTPUT_NAM, \$RMS_PTR+40	
			FF58	CD	0104	CE	D0	00052	MOVL	OUTPUT_FILE_NAME+4, \$RMS_PTR+44	
			FF5C	CD	0000	CF	9E	00059	MOVAB	P.AAB, \$RMS_PTR+48	
			FF60	CD	0100	CE	90	00060	MOVB	OUTPUT_FILE_NAME, \$RMS_PTR+52	
			FF61	CD		09	90	00067	MOVB	#9, \$RMS_PTR+53	
0060	8F	00	FF62	CD	84	8F	9B	0006C	MOVZBW	#132, \$RMS_PTR+54	
				6E		00	2C	00072	MOVCS	#0, (SP), #0, #96, \$RMS_PTR	0290
					FECC	CD		00079			
			FECC	CD	6002	8F	B0	0007C	MOVW	#24578, \$RMS_PTR	
			FECE	CD		01	8E	00083	MNEGB	#1, \$RMS_PTR+2	
0044	8F	00	FED0	CD		6E	9E	00088	MOVAB	OUTPUT_RESULT_NAME, \$RMS_PTR+4	
				6E		00	2C	0008D	MOVCS	#0, (SP), #0, #68, \$RMS_PTR	0296
					0108	CE		00094			
			0108	CE	4401	8F	B0	00097	MOVW	#17409, \$RMS_PTR	
			010C	CE	0400	8F	3C	0009E	MOVZWL	#1024, \$RMS_PTR+4	
					0126	CE	94	000A5	CLRB	\$RMS_PTR+30	
			0128	CE	84	8F	9B	000A9	MOVZBW	#132, \$RMS_PTR+32	
			012C	CE	FF7C	CD	9E	000AF	MOVAB	OUTPUT_BUFFER, \$RMS_PTR+36	
			FEC4	CD	FF2C	CD	9E	000B6	MOVAB	OUTPUT_FAB, \$RMS_PTR+60	
					FF2C	CD	9F	000BD	PUSHAB	OUTPUT_FAB	0306
		00000000G		00		01	FB	000C1	CALLS	#1, SYS\$CREATE	
				14		50	E8	000C8	BLBS	CREATE_STATUS, 1\$	0308
				7E	FF34	CD	7D	000CB	MOVQ	OUTPUT_FAB+8, -(SP)	0314
					0108	CE	9F	000D0	PUSHAB	OUTPUT_FILE_NAME	0310
						01	DD	000D4	PUSHL	#1	
					001A10A4	8F	DD	000D6	PUSHL	#1708196	0311
		66				05	FB	000DC	CALLS	#5, LIB\$STOP	
					0108	CE	9F	000DF	PUSHAB	OUTPUT_RAB	0316
		00000000G		00		01	FB	000E3	CALLS	#1, SYS\$CONNECT	
				17		50	E8	000EA	BLBS	CONNECT_STATUS, 2\$	0318
					0114	CE	DD	000ED	PUSHL	OUTPUT_RAB+12	0324
					0114	CE	DD	000F1	PUSHL	OUTPUT_RAB+8	
					0108	CE	9F	000F5	PUSHAB	OUTPUT_FILE_NAME	0320
						01	DD	000F9	PUSHL	#1	
					001A10A4	8F	DD	000FB	PUSHL	#1708196	0321
		66				05	FB	00101	CALLS	#5, LIB\$STOP	
			0100	CE	FECF	CD	9B	00104	MOVZBW	OUTPUT_NAM+3, OUTPUT_FILE_NAME	0330
			0102	CE	010E	8F	B0	0010B	MOVW	#270, OUTPUT_FILE_NAME+2	0331
			0104	CE	FED0	CD	D0	00112	MOVL	OUTPUT_NAM+4, OUTPUT_FILE_NAME+4	0333
					0108	CE	9F	00119	PUSHAB	OUTPUT_RAB	0335
		0000V		CF		01	FB	0011D	CALLS	#1, WRITE_FILE	
				17		50	E8	00122	BLBS	R0, 3\$	
					0114	CE	DD	00125	PUSHL	OUTPUT_RAB+12	0341
					0114	CE	DD	00129	PUSHL	OUTPUT_RAB+8	
					0108	CE	9F	0012D	PUSHAB	OUTPUT_FILE_NAME	0337
						01	DD	00131	PUSHL	#1	
					001A10D4	8F	DD	00133	PUSHL	#1708244	0338
		66				05	FB	00139	CALLS	#5, LIB\$STOP	
					FF2C	CD	9F	0013C	PUSHAB	OUTPUT_FAB	0351
		00000000G		00		01	FB	00140	CALLS	#1, SYS\$CLOSE	
				14		50	E8	00147	BLBS	CLOSE_STATUS, 4\$	0353
				7E	FF34	CD	7D	0014A	MOVQ	OUTPUT_FAB+8, -(SP)	0359
					0108	CE	9F	0014F	PUSHAB	OUTPUT_FILE_NAME	0355

BASMSG
2-003

BASMSG - write BASMSG.MSG
BASMSG - Write BASMSG.TMP

D 10
16-Sep-1984 01:42:08
14-Sep-1984 11:55:22

VAX-11 Bliss-32 V4.0-742
[BASRTL.SPC]BASMSGGEN.B32;1

Page 11
(4)

	001A105C	01	DD	00153	PUSHL	#1
		8F	DD	00155	PUSHL	#1708124
66		05	FB	0015B	CALLS	#5, LIB\$STOP
50		01	DO	0015E 4\$:	MOVL	#1, R0
		04		00161	RET	

:
: 0356
:
: 0362
: 0363

; Routine Size: 354 bytes, Routine Base: \$CODE\$ + 0000

```

: 368 0364 1 %SBTTL 'WRITE_FILE - Actually write the file'
: 369 0365 1 ROUTINE WRITE_FILE (           ! Actually write the file
: 370 0366 1           OUTPUT_RAB           ! Where to write each record
: 371 0367 1           ) =
: 372 0368 1
: 373 0369 1  +-+
: 374 0370 1  FUNCTIONAL DESCRIPTION:
: 375 0371 1
: 376 0372 1           This routine writes each record on the specified RAB.
: 377 0373 1
: 378 0374 1  CALLING SEQUENCE:
: 379 0375 1
: 380 0376 1           ret_status.wlc.v = WRITE_FILE (OUTPUT_RAB.mz.r)
: 381 0377 1
: 382 0378 1  FORMAL PARAMETERS:
: 383 0379 1
: 384 0380 1           OUTPUT_RAB           RAB onto which to write the text
: 385 0381 1
: 386 0382 1  IMPLICIT INPUTS:
: 387 0383 1
: 388 0384 1           NONE
: 389 0385 1
: 390 0386 1  IMPLICIT OUTPUTS:
: 391 0387 1
: 392 0388 1           None
: 393 0389 1
: 394 0390 1  COMPLETION STATUS:
: 395 0391 1
: 396 0392 1           $$$_NORMAL           Normal successful completion
: 397 0393 1           Any errors from RMS $PUT
: 398 0394 1
: 399 0395 1  SIDE EFFECTS:
: 400 0396 1
: 401 0397 1           Writes on the file connected to OUTPUT_RAB
: 402 0398 1
: 403 0399 1  --
: 404 0400 1
: 405 0401 2  BEGIN
: 406 0402 2
: 407 0403 2  MAP
: 408 0404 2  OUTPUT_RAB : REF $RAB_DECL;
: 409 0405 2
: 410 0406 2  LOCAL
: 411 0407 2  +-+
: 412 0408 2  Stuff for BUILD_TEXT_LINE
: 413 0409 2  -
: 414 0410 2  CTL_STR_DSC : BLOCK [8, BYTE],
: 415 0411 2  TEMP_STR_DSC : BLOCK [8, BYTE],
: 416 0412 2  TEMP_STRING : VECTOR [132, BYTE],
: 417 0413 2  OUT_LENGTH,
: 418 0414 2  +-+
: 419 0415 2  Stuff for PRINT_LINE
: 420 0416 2  -
: 421 0417 2  LINE_DESC : BLOCK [8, BYTE],
: 422 0418 2  +-+
: 423 0419 2  End of stuff for PRINT_LINE
: 424 0420 2  -

```

```

: 425      0421 2      PRINTABLE_DESC : BLOCK [8, BYTE],
: 426      0422 2      HEX_DESC : BLOCK [8, BYTE],
: 427      0423 2      TEXT : BLOCK [8, BYTE],
: 428      0424 2      NAME : BLOCK [8, BYTE],
: 429      0425 2      SEVERITY : BLOCK [8, BYTE],
: 430      0426 2      NAMLEN,
: 431      0427 2      TXTLEN,
: 432      0428 2      SEVLEN,
: 433      0429 2      SEVERITY_ADDR : REF VECTOR [, BYTE];
: 434      0430 2
: 435      0431 2      +
: 436      0432 2      :- Set up TEMP_STR_DSC for BUILD_TEXT_LINE
: 437      0433 2
: 438      0434 2      TEMP_STR_DSC [DSC$W_LENGTH] = 132;
: 439      0435 2      TEMP_STR_DSC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
: 440      0436 2      TEMP_STR_DSC [DSC$B_CLASS] = DSC$K_CLASS_S;
: 441      0437 2      TEMP_STR_DSC [DSC$A_POINTER] = (H$PTR (TEMP_STRING));
: 442      0438 2
: 443      0439 2      +
: 444      0440 2      :- Set up LINE_DESC for PRINT_LINE, etc.
: 445      0441 2
: 446      0442 2      INIT_DESCRIPTOR (LINE_DESC);
: 447      0443 2      INIT_DESCRIPTOR (PRINTABLE_DESC);
: 448      0444 2      INIT_DESCRIPTOR (HEX_DESC);
: 449      0445 2      INIT_DESCRIPTOR (TEXT);
: 450      0446 2      INIT_DESCRIPTOR (NAME);
: 451      0447 2      INIT_DESCRIPTOR (SEVERITY);
: 452      0448 2
: 453      0449 2      +
: 454      0450 2      :- Put out the initial information.
: 455      0451 2
: 456      0452 2      PRINT_LINE (<'!!! This file, BASMSG.TMP, contains the definitions of the BASIC'>, <' '>);
: 457      0453 2      PRINT_LINE (<'!!! messages for VAX/VMS. This file is read by the MESSAGE compiler'>, <' '>);
: 458      0454 2      PRINT_LINE (<'!!! to build an object file containing the BASIC messages.'>, <' '>);
: 459      0455 2      PRINT_LINE (<'!!!'>, <' '>);
: 460      0456 2      PRINT_LINE (<' .TITLE BASIC's message text'>, <' '>);
: 461      0457 2      PRINT_LINE (<'!!!'>, <' '>);
: 462      0458 2      PRINT_LINE (<' .FACILITY/SYSTEM BAS, !SL'>, <BAS$K_FAC_NO>);
: 463      0459 2      PRINT_LINE (<' .BASE 0'>, <' '>);
: 464      0460 2
: 465      0461 2      +
: 466      0462 2      :- Write a line for each of the 256 BASIC messages
: 467      0463 2
: 468      0464 2      INCR CODE FROM 0 TO 255 DO
: 469      0465 2      BEGIN
: 470      0466 2      ERRTXT (CODE, NAMLEN, NAME, TXTLEN, TEXT, SEVLEN, SEVERITY);
: 471      0467 2      PRINT_LINE (<'!AS/!AS <!AS'>'>, <NAME, SEVERITY, TEXT>);
: 472      0468 2      END;
: 473      0469 2
: 474      0470 2      +
: 475      0471 2      :- Write out the 11 trailing error codes beginning at code 4085.
: 476      0472 2
: 477      0473 2      PRINT_LINE (<' .BASE 4085'>, <' '>);
: 478      0474 2      PRINT_LINE (<'FORFILUSE/INFO <for file !!AS at user PC !!XL> /FAO=2'>, <' '>);
: 479      0475 2      PRINT_LINE (<'USEPC PSL/INFO <at user PC=!!XL, PSL=!!XL> /FAO=2'>, <' '>);
: 480      0476 2      PRINT_LINE (<'FROIOC/INFO <from immediate mode in module !!AC> /FAO=1'>, <' '>);
: 481      0477 2      PRINT_LINE (<'ON CHAFIL/INFO <on channel !!SL for file !!AS at user PC !!XL> /FAO=3'>, <' '>);
:         0477 2      PRINT_LINE (<'FRÖLINMOD/INFO <from line !!SL in module !!AC> /FAO=2'>, <' '>);
```

```

: 482 0478 2 PRINT_LINE (<'FROLINSUB/INFO <from line !!SL in subprogram !!AC> /FAO=2'>, <' '>);
: 483 0479 2 PRINT_LINE (<'FROLINFUN/INFO <from line !!SL in external function !!AC> /FAO=2'>, <' '>);
: 484 0480 2 PRINT_LINE (<'FROLINDEF/INFO <from line !!SL in DEF !!AC in module !!AC> /FAO=3'>, <' '>);
: 485 0481 2 PRINT_LINE (<'FROLINDFS/INFO <from line !!SL in DEF* !!AC in module !!AC> /FAO=3'>, <' '>);
: 486 0482 2 PRINT_LINE (<'FROLINGSB/INFO <from line !!SL in GOSUB !!SL in module !!AC> /FAO=3'>, <' '>);
: 487 0483 2 PRINT_LINE (<'FROLINDEG/INFO <from line !!SL in ON ERROR GOTO !!SL in module !!AC> /FAO=3'>, <' '>);
: 488 0484 2
: 489 0485 2 !+
: 490 0486 2 !- Write out the trailer line
: 491 0487 2
: 492 0488 2 PRINT_LINE (<'.END'>, <' '>);
: 493 0489 2 !+
: 494 0490 2 !- All done.
: 495 0491 2
: 496 0492 2 DISCARD_DESCRIPTOR (LINE_DESC);
: 497 0493 2 DISCARD_DESCRIPTOR (PRINTABLE_DESC);
: 498 0494 2 DISCARD_DESCRIPTOR (HEX_DESC);
: 499 0495 2 DISCARD_DESCRIPTOR (TEXT);
: 500 0496 2 DISCARD_DESCRIPTOR (NAME);
: 501 0497 2 DISCARD_DESCRIPTOR (SEVERITY);
: 502 0498 2 RETURN (SS$_NORMAL);
: 503 0499 1 END;

```

! End of routine WRITE_FILE

```

.PSECT $SPLITS,NOWRT,NOEXE,2
42 20 2C 65 6C 69 66 20 73 69 68 54 20 21 21 00014 P.AAC: .ASCII \!! This file, BASMSG.TMP, contains the d\
74 6E 6F 63 20 2C 50 4D 54 2E 47 53 4D 53 41 00023
: 74 20 66 6F 20 73 6E 65 68 74 20 73 6E 69 61 00032
: 66 6F 69 74 69 6E 69 66 65 0003C
: 72 6F 66 20 73 65 67 61 73 73 65 6D 20 21 21 0004B
: 73 69 68 54 20 2E 53 4D 56 2F 58 41 56 20 00054 P.AAD: .ASCII \!! messages for VAX/VMS. This file is r\
: 68 54 20 72 20 73 69 20 65 6C 69 66 20 00063
: 53 53 45 4D 20 65 68 74 20 79 62 20 64 61 65 00072
: 00 72 65 6C 69 70 6D 6F 63 20 45 47 41 0007C .ASCII \lead by the MESSAGE compiler\<0>
: 20 6E 61 20 64 6C 69 75 62 20 6F 74 20 21 21 0008B
: 6E 6F 63 20 65 6C 69 66 20 74 63 65 6A 62 6F 00098 P.AAE: .ASCII \!! to build an object file containing th\
: 65 67 61 73 73 65 6D 20 43 49 53 41 42 20 65 000A7
: 61 73 73 65 6D 20 43 49 53 41 42 20 65 000B6
: 67 61 73 73 65 6D 20 43 49 53 41 42 20 65 000C0 .ASCII \e BASIC messages.\<0><0><0>
: 61 73 73 65 6D 20 43 49 53 41 42 20 65 000CF
: 20 45 4C 54 49 54 2E 20 20 20 20 20 20 20 20 000D4 P.AAF: .ASCII \!!!\<0><0>
: 65 67 61 73 73 65 6D 20 73 27 43 49 53 41 42 000D8 P.AAG: .ASCII \ .TITLE BASIC's message text\<0>
: 67 61 73 73 65 6D 20 73 27 43 49 53 41 42 000E7
: 61 73 73 65 6D 20 73 27 43 49 53 41 42 000F6
: 45 54 53 59 53 2F 59 54 49 4C 49 43 41 46 2E 000FC P.AAH: .ASCII \!!!\<0><0>
: 00 00 00 4C 53 21 20 2C 53 41 42 20 4D 00100 P.AAI: .ASCII \.FACILITY/SYSTEM BAS, !SL\<0><0><0>
: 00 00 3E 53 41 21 3C 20 53 41 21 2F 53 41 21 0010F
: 00 00 3E 53 41 21 3C 20 53 41 21 2F 53 41 21 0011C P.AAJ: .ASCII \.BASE 0\<0>
: 00 00 3E 53 41 21 3C 20 53 41 21 2F 53 41 21 00124 P.AAK: .ASCII \!AS/!AS <!AS>\<0><0><0>
: 00 00 3E 53 41 21 3C 20 53 41 21 2F 53 41 21 00133
: 20 4F 46 4E 49 2F 45 53 55 4C 49 46 52 4F 46 00134 P.AAL: .ASCII \.BASE 4085\<0><0>
: 20 53 41 21 21 20 65 6C 69 66 20 72 6F 66 3C 00140 P.AAM: .ASCII \FORFILUSE/INFO <for file !!AS at user PC\
: 43 50 20 72 65 73 75 20 74 61 0014F
: 00 00 32 3D 4F 41 46 2F 20 3E 4C 58 21 21 20 0015E
: 00 00 32 3D 4F 41 46 2F 20 3E 4C 58 21 21 20 00168 .ASCII \ !!XL> /FAO=2\<0><0><0>

```



```

20 4F 46 4E 49 2F 4C 53 50 5F 43 50 45 53 00 00177
58 21 21 3D 43 50 20 72 65 73 75 20 74 61 3C 00178 P.AAN: .ASCII \USEPC_PSL/INFO <at user PC=!!XL, PSL=!!X\
    58 21 21 3D 43 58 21 21 3D 4C 53 50 20 74 61 3C 00187
    00 00 00 32 3D 4F 41 46 2F 20 3E 4C 00196
72 66 3C 20 4F 46 4E 49 2F 4C 4F 49 4F 52 46 001A0 .ASCII \L> /FAO=2\<0><0><0>
6F 6D 20 65 74 61 69 64 65 6D 6D 69 20 6D 6F 001AC P.AAO: .ASCII \FROIOL/INFO <from immediate mode in modul
    31 3D 4F 41 46 2F 20 3E 43 41 21 21 20 65 64 001BB
    75 64 6F 6D 20 6E 69 20 65 64 001CA
    2F 20 3E 43 41 21 21 20 65 6C 001D4 .ASCII \le !!AC> /FAO=1\<0>
    00 001E3
20 4F 46 4E 49 2F 4C 49 46 41 48 43 5F 4E 4F 001E4 P.AAP: .ASCII \ON_CHAFIL/INFO <on channel !!SL for file\
53 21 21 20 6C 65 6E 6E 61 68 63 20 6E 6F 3C 001F3
    65 6C 69 66 20 72 6F 66 20 4C 00202
50 20 72 65 73 75 20 74 61 20 53 41 21 21 20 0020C .ASCII \ !!AS at user PC !!XL> /FAO=3\<0><0>
00 33 3D 4F 41 46 2F 20 3E 4C 58 21 21 20 43 0021B
    00 0022A
    00 0022B .ASCII <0>
20 4F 46 4E 49 2F 44 4F 4D 4E 49 4C 4F 52 46 0022C P.AAQ: .ASCII \FROLINMOD/INFO <from line !!SL in module\
4C 53 21 21 20 65 6E 69 6C 20 6D 6F 72 66 3C 0023B
    65 6C 75 64 6F 6D 20 6E 69 20 0024A
00 00 32 3D 4F 41 46 2F 20 3E 43 41 21 21 20 00254 .ASCII \ !!AC> /FAO=2\<0><0><0>
    00 00263
20 4F 46 4E 49 2F 42 55 53 4E 49 4C 4F 52 46 00264 P.AAR: .ASCII \FROLINSUB/INFO <from line !!SL in subpro\
4C 53 21 21 20 65 6E 69 6C 20 6D 6F 72 66 3C 00273
    6F 72 70 62 75 73 20 6E 69 20 00282
4F 41 46 2F 20 3E 43 41 21 21 20 6D 61 72 67 0028C .ASCII \gram !!AC> /FAO=2\<0><0><0>
    00 00 00 32 3D 0029B
20 4F 46 4E 49 2F 4E 55 46 4E 49 4C 4F 52 46 002A0 P.AAS: .ASCII \FROLINFUN/INFO <from line !!SL in extern\
4C 53 21 21 20 65 6E 69 6C 20 6D 6F 72 66 3C 002AF
    6E 72 65 74 78 65 20 6E 69 20 002BE
41 21 21 20 6E 6F 69 74 63 6E 75 66 20 6C 61 61 002C8 .ASCII \al function !!AC> /FAO=2\
    32 3D 4F 41 46 2F 20 3E 43 002D7
20 4F 46 4E 49 2F 46 45 44 4E 49 4C 4F 52 46 002E0 P.AAT: .ASCII \FROLINDEF/INFO <from line !!SL in DEF !!\
4C 53 21 21 20 65 6E 69 6C 20 6D 6F 72 66 3C 002EF
    21 21 20 46 45 44 20 6E 69 20 002FE
21 21 20 65 6C 75 64 6F 6D 20 6E 69 20 43 41 00308 .ASCII \AC in module !!AC> /FAO=3\<0><0><0>
    00 00 00 33 3D 4F 41 46 2F 20 3E 43 41 00317
20 4F 46 4E 49 2F 53 46 44 4E 49 4C 4F 52 46 00324 P.AAU: .ASCII \FROLINDFS/INFO <from line !!SL in DEF* !!\
4C 53 21 21 20 65 6E 69 6C 20 6D 6F 72 66 3C 00333
    21 20 2A 46 45 44 20 6E 69 20 00342
21 20 65 6C 75 64 6F 6D 20 6E 69 20 43 41 21 0034C .ASCII \!AC in module !!AC> /FAO=3\<0><0>
    00 00 33 3D 4F 41 46 2F 20 3E 43 41 21 0035B
20 4F 46 4E 49 2F 42 53 47 4E 49 4C 4F 52 46 00368 P.AAV: .ASCII \FROLINGSB/INFO <from line !!SL in GOSUB \
4C 53 21 21 20 65 6E 69 6C 20 6D 6F 72 66 3C 00377
    20 42 55 53 4F 47 20 6E 69 20 00386
20 65 6C 75 64 6F 6D 20 6E 69 20 4C 53 21 21 00390 .ASCII \!!SL in module !!AC> /FAO=3\<0>
    00 33 3D 4F 41 46 2F 20 3E 43 41 21 21 0039F
20 4F 46 4E 49 2F 47 45 4F 4E 49 4C 4F 52 46 003AC P.AAW: .ASCII \FROLINOEG/INFO <from line !!SL in ON ERR\
4C 53 21 21 20 65 6E 69 6C 20 6D 6F 72 66 3C 003BB
    52 52 45 20 4E 4F 20 6E 69 20 003CA
6E 69 20 4C 53 21 21 20 4F 54 4F 47 20 52 4F 003D4 .ASCII \OR GOTO !!SL in module !!AC> /FAO=3\<0>
2F 20 3E 43 41 21 21 20 65 6C 75 64 6F 6D 20 003E3
    00 33 3D 4F 41 46 003F2
    44 4E 45 2E 003F8 P.AAX: .ASCII \.END\
    .EXTRN SYSSFAO

```

.PSECT \$CODE\$,NOWRT,2

00FC 00000 WRITE_FILE:

					.WORD	Save R2,R3,R4,R5,R6,R7		0365
	57	00000000G	00	9E	00002	MOVAB	STR\$FREE1_DX, R7	
	56	0000V	CF	9E	00009	MOVAB	PRINT, R6	
	55	00000000G	00	9E	0000E	MOVAB	STR\$COPY_R, R5	
	54	00000000G	00	9E	00015	MOVAB	SY\$FAO, R4	
	53	00000000G	00	9E	0001C	MOVAB	LIB\$STOP, R3	
	5E	FF28	CE	9E	00023	MOVAB	-216(SP), SP	
F0	AD	010E0084	8F	D0	00028	MOVL	#17694852, TEMP_STR_DSC	0434
F4	AD	44	AE	9E	00030	MOVAB	TEMP_STRING, TEMP_STR_DSC+4	0437
3C	AE	020E0000	8F	D0	00035	MOVL	#34471936, LINE_DESC	0441
		40	AE	D4	0003D	CLRL	LINE_DESC+4	
34	AE	020E0000	8F	D0	00040	MOVL	#34471936, PRINTABLE_DESC	0442
		38	AE	D4	00048	CLRL	PRINTABLE_DESC+4	
2C	AE	020E0000	8F	D0	0004B	MOVL	#34471936, HEX_DESC	0443
		30	AE	D4	00053	CLRL	HEX_DESC+4	
24	AE	020E0000	8F	D0	00056	MOVL	#34471936, TEXT	0444
		28	AE	D4	0005E	CLRL	TEXT+4	
1C	AE	020E0000	8F	D0	00061	MOVL	#34471936, NAME	0445
		20	AE	D4	00069	CLRL	NAME+4	
14	AE	020E0000	8F	D0	0006C	MOVL	#34471936, SEVERITY	0446
		18	AE	D4	00074	CLRL	SEVERITY+4	
F8	AD	010E003F	8F	D0	00077	MOVL	#17694783, CTL_STR_DSC	0450
FC	AD	0000	CF	9E	0007F	MOVAB	P.AAC, CTL_STR_DSC+4	
			20	DD	00085	PUSHL	#32	
		F0	AD	9F	00087	PUSHAB	TEMP_STR_DSC	
		18	AE	9F	0008A	PUSHAB	OUT_LENGTH	
		F8	AD	9F	0008D	PUSHAB	CTL_STR_DSC	
64			04	FB	00090	CALLS	#4, SY\$FAO	
05			50	EB	00093	BLBS	FAO_STATUS, 1\$	
			50	DD	00096	PUSHL	FAO_STATUS	
63			01	FB	00098	CALLS	#1, LIB\$STOP	
		F4	AD	DD	0009B	PUSHL	TEMP_STR_DSC+4	
		14	AE	9F	0009E	PUSHAB	OUT_LENGTH	
		44	AE	9F	000A1	PUSHAB	LINE_DESC	
65			03	FB	000A4	CALLS	#3, STR\$COPY_R	
7D			50	E9	000A7	BLBC	BUILD_STATUS, 4\$	
		3C	AE	9F	000AA	PUSHAB	LINE_DESC	
52		04	AC	D0	000AD	MOVL	OUTPUT_RAB, R2	
			52	DD	000B1	PUSHL	R2	
66			02	FB	000B3	CALLS	#2, PRINT	
79			50	E9	000B6	BLBC	PRINT_STATUS, 5\$	
F8	AD	010E0043	8F	D0	000B9	MOVL	#17694787, CTL_STR_DSC	0451
FC	AD	0000	CF	9E	000C1	MOVAB	P.AAD, CTL_STR_DSC+4	
			20	DD	000C7	PUSHL	#32	
		F0	AD	9F	000C9	PUSHAB	TEMP_STR_DSC	
		18	AE	9F	000CC	PUSHAB	OUT_LENGTH	
		F8	AD	9F	000CF	PUSHAB	CTL_STR_DSC	
64			04	FB	000D2	CALLS	#4, SY\$FAO	
05			50	EB	000D5	BLBS	FAO_STATUS, 2\$	
			50	DD	000D8	PUSHL	FAO_STATUS	
63			01	FB	000DA	CALLS	#1, LIB\$STOP	
		F4	AD	DD	000DD	PUSHL	TEMP_STR_DSC+4	
		14	AE	9F	000E0	PUSHAB	OUT_LENGTH	
		44	AE	9F	000E3	PUSHAB	LINE_DESC	

	65		03	FB	000E6	CALLS	#3, STR\$COPY_R	
	79		50	E9	000E9	BLBC	BUILD STATUS, 7\$	
		3C	AE	9F	000EC	PUSHAB	LINE_DESC	
			52	DD	000EF	PUSHL	R2	
	66		02	FB	000F1	CALLS	#2, PRINT	
	79		50	E9	000F4	BLBC	PRINT STATUS, 8\$	
F8	AD	010E0039	8F	DD	000F7	MOVL	#17694777, CTL_STR_DSC	
FC	AD	0000	CF	9E	000FF	MOVAB	P.AAE, CTL_STR_DSC+4	
			20	DD	00105	PUSHL	#32	
		F0	AD	9F	00107	PUSHAB	TEMP_STR_DSC	
		18	AE	9F	0010A	PUSHAB	OUT_LENGTH	
		F8	AD	9F	0010D	PUSHAB	CTL_STR_DSC	
	64		04	FB	00110	CALLS	#4, -SYS\$FAO	
	05		50	E8	00113	BLBS	FAO_STATUS, 3\$	
			50	DD	00116	PUSHL	FAO_STATUS	
	63		01	FB	00118	CALLS	#1, -LIB\$STOP	
		F4	AD	DD	0011B	PUSHL	TEMP_STR_DSC+4	3\$:
		14	AE	9F	0011E	PUSHAB	OUT_LENGTH	
		44	AE	9F	00121	PUSHAB	LINE_DESC	
	65		03	FB	00124	CALLS	#3, STR\$COPY_R	
	79		50	E9	00127	BLBC	BUILD STATUS, 10\$	4\$:
		3C	AE	9F	0012A	PUSHAB	LINE_DESC	
			52	DD	0012D	PUSHL	R2	
	66		02	FB	0012F	CALLS	#2, PRINT	
	79		50	E9	00132	BLBC	PRINT STATUS, 11\$	5\$:
F8	AD	010E0002	8F	DD	00135	MOVL	#17694722, CTL_STR_DSC	
FC	AD	0000	CF	9E	0013D	MOVAB	P.AAF, CTL_STR_DSC+4	
			20	DD	00143	PUSHL	#32	
		F0	AD	9F	00145	PUSHAB	TEMP_STR_DSC	
		18	AE	9F	00148	PUSHAB	OUT_LENGTH	
		F8	AD	9F	0014B	PUSHAB	CTL_STR_DSC	
	64		04	FB	0014E	CALLS	#4, -SYS\$FAO	
	05		50	E8	00151	BLBS	FAO_STATUS, 6\$	
			50	DD	00154	PUSHL	FAO_STATUS	
	63		01	FB	00156	CALLS	#1, -LIB\$STOP	
		F4	AD	DD	00159	PUSHL	TEMP_STR_DSC+4	6\$:
		14	AE	9F	0015C	PUSHAB	OUT_LENGTH	
		44	AE	9F	0015F	PUSHAB	LINE_DESC	
	65		03	FB	00162	CALLS	#3, STR\$COPY_R	
	79		50	E9	00165	BLBC	BUILD STATUS, 13\$	7\$:
		3C	AE	9F	00168	PUSHAB	LINE_DESC	
			52	DD	0016B	PUSHL	R2	
	66		02	FB	0016D	CALLS	#2, PRINT	
	79		50	E9	00170	BLBC	PRINT STATUS, 14\$	8\$:
F8	AD	010E0023	8F	DD	00173	MOVL	#17694755, CTL_STR_DSC	
FC	AD	0000	CF	9E	0017B	MOVAB	P.AAG, CTL_STR_DSC+4	
			20	DD	00181	PUSHL	#32	
		F0	AD	9F	00183	PUSHAB	TEMP_STR_DSC	
		18	AE	9F	00186	PUSHAB	OUT_LENGTH	
		F8	AD	9F	00189	PUSHAB	CTL_STR_DSC	
	64		04	FB	0018C	CALLS	#4, -SYS\$FAO	
	05		50	E8	0018F	BLBS	FAO_STATUS, 9\$	
			50	DD	00192	PUSHL	FAO_STATUS	
	63		01	FB	00194	CALLS	#1, -LIB\$STOP	
		F4	AD	DD	00197	PUSHL	TEMP_STR_DSC+4	9\$:
		14	AE	9F	0019A	PUSHAB	OUT_LENGTH	
		44	AE	9F	0019D	PUSHAB	LINE_DESC	

0452

0453

0454

	65		03	FB	001A0		CALLS	#3, STR\$COPY_R	
	79		50	E9	001A3	10\$:	BLBC	BUILD STATUS, 16\$	
		3C	AE	9F	001A6		PUSHAB	LINE_DESC	
			52	DD	001A9		PUSHL	R2	
	66		02	FB	001AB		CALLS	#2, PRINT	
	79		50	E9	001AE	11\$:	BLBC	PRINT STATUS, 17\$	
FB	AD	010E0002	8F	DO	001B1		MOVL	#17694722, CTL_STR_DSC	
FC	AD	0000	CF	9E	001B9		MOVAB	P.AAH, CTL_STR_DSC+4	
			20	DD	001BF		PUSHL	#32	
		F0	AD	9F	001C1		PUSHAB	TEMP_STR_DSC	
		18	AE	9F	001C4		PUSHAB	OUT_LENGTH	
		F8	AD	9F	001C7		PUSHAB	CTL_STR_DSC	
	64		04	FB	001CA		CALLS	#4, -SYS\$FAO	
	05		50	E8	001CD		BLBS	FAO_STATUS, 12\$	
			50	DD	001D0		PUSHL	FAO_STATUS	
	63		01	FB	001D2		CALLS	#1, -LIB\$STOP	
		F4	AD	DD	001D5	12\$:	PUSHL	TEMP_STR_DSC+4	
		14	AE	9F	001D8		PUSHAB	OUT_LENGTH	
		44	AE	9F	001DB		PUSHAB	LINE_DESC	
	65		03	FB	001DE		CALLS	#3, STR\$COPY_R	
	79		50	E9	001E1	13\$:	BLBC	BUILD STATUS, 19\$	
		3C	AE	9F	001E4		PUSHAB	LINE_DESC	
			52	DD	001E7		PUSHL	R2	
	66		02	FB	001E9		CALLS	#2, PRINT	
	79		50	E9	001EC	14\$:	BLBC	PRINT STATUS, 20\$	
FB	AD	010E0019	8F	DO	001EF		MOVL	#17694745, CTL_STR_DSC	
FC	AD	0000	CF	9E	001F7		MOVAB	P.AAI, CTL_STR_DSC+4	
			1A	DD	001FD		PUSHL	#26	
		F0	AD	9F	001FF		PUSHAB	TEMP_STR_DSC	
		18	AE	9F	00202		PUSHAB	OUT_LENGTH	
		F8	AD	9F	00205		PUSHAB	CTL_STR_DSC	
	64		04	FB	00208		CALLS	#4, -SYS\$FAO	
	05		50	E8	0020B		BLBS	FAO_STATUS, 15\$	
			50	DD	0020E		PUSHL	FAO_STATUS	
	63		01	FB	00210		CALLS	#1, -LIB\$STOP	
		F4	AD	DD	00213	15\$:	PUSHL	TEMP_STR_DSC+4	
		14	AE	9F	00216		PUSHAB	OUT_LENGTH	
		44	AE	9F	00219		PUSHAB	LINE_DESC	
	65		03	FB	0021C		CALLS	#3, STR\$COPY_R	
	46		50	E9	0021F	16\$:	BLBC	BUILD STATUS, 20\$	
		3C	AE	9F	00222		PUSHAB	LINE_DESC	
			52	DD	00225		PUSHL	R2	
	66		02	FB	00227		CALLS	#2, PRINT	
	38		50	E9	0022A	17\$:	BLBC	PRINT STATUS, 20\$	
FB	AD	010E0007	8F	DO	0022D		MOVL	#17694727, CTL_STR_DSC	
FC	AD	0000	CF	9E	00235		MOVAB	P.AAJ, CTL_STR_DSC+4	
			20	DD	00238		PUSHL	#32	
		F0	AD	9F	0023D		PUSHAB	TEMP_STR_DSC	
		18	AE	9F	00240		PUSHAB	OUT_LENGTH	
		F8	AD	9F	00243		PUSHAB	CTL_STR_DSC	
	64		04	FB	00246		CALLS	#4, -SYS\$FAO	
	05		50	E8	00249		BLBS	FAO_STATUS, 18\$	
			50	DD	0024C		PUSHL	FAO_STATUS	
	63		01	FB	0024E		CALLS	#1, -LIB\$STOP	
		F4	AD	DD	00251	18\$:	PUSHL	TEMP_STR_DSC+4	
		14	AE	9F	00254		PUSHAB	OUT_LENGTH	
		44	AE	9F	00257		PUSHAB	LINE_DESC	

.....
0455
.....
0456
.....
0457
.....

65		03	FB	0025A	CALLS	#3, STR\$COPY_R	
6C		50	E9	0025D	19\$:	BLBC	BUILD STATUS, 23\$
	3C	AE	9F	00260	PUSHAB	LINE_DESC	
		52	DD	00263	PUSHL	R2	
66		02	FB	00265	CALLS	#2, PRINT	
61		50	E9	00268	20\$:	BLBC	PRINT STATUS, 23\$
	0C	AE	D4	0026B	CLRL	CODE	0462
	14	AE	9F	0026E	21\$:	PUSHAB	SEVERITY
	04	AE	9F	00271	PUSHAB	SEVLEN	0464
	2C	AE	9F	00274	PUSHAB	TEXT	
	10	AE	9F	00277	PUSHAB	TXLEN	
	2C	AE	9F	0027A	PUSHAB	NAME	
	1C	AE	9F	0027D	PUSHAB	NAMLEN	
	24	AE	9F	00280	PUSHAB	CODE	
00000000G	00	07	FB	00283	CALLS	#7, ERRTXT	
	F8	AD	010E000D	8F	DO	0028A	
	FC	AD	0000	CF	9E	00292	
		24	AE	9F	00298	PUSHAB	TEXT
		18	AE	9F	0029B	PUSHAB	SEVERITY
		24	AE	9F	0029E	PUSHAB	NAME
		F0	AD	9F	002A1	PUSHAB	TEMP_STR_DSC
		20	AE	9F	002A4	PUSHAB	OUT_LENGTH
		F8	AD	9F	002A7	PUSHAB	CTL_STR_DSC
64		06	FB	002AA	CALLS	#6, SYS\$FAO	
05		50	E8	002AD	BLBS	FAO STATUS, 22\$	
		50	DD	002B0	PUSHL	FAO STATUS	
63		01	FB	002B2	CALLS	#1, LIB\$STOP	
	F4	AD	DD	002B5	22\$:	PUSHL	TEMP_STR_DSC+4
	14	AE	9F	002B8	PUSHAB	OUT_LENGTH	
	44	AE	9F	002BB	PUSHAB	LINE_DESC	
65		03	FB	002BE	CALLS	#3, STR\$COPY_R	
4F		50	E9	002C1	BLBC	BUILD STATUS, 25\$	
	3C	AE	9F	002C4	PUSHAB	LINE_DESC	
		52	DD	002C7	PUSHL	R2	
66		02	FB	002C9	CALLS	#2, PRINT	
77		50	E9	002CC	23\$:	BLBC	PRINT STATUS, 27\$
96	0C	AE	000000FF	8F	F3	002CF	
	F8	AD	010E000A	8F	DO	002D8	
	FC	AD	0000	CF	9E	002E0	
		20	DD	002E6	PUSHL	#32	
		F0	AD	9F	002E8	PUSHAB	TEMP_STR_DSC
		18	AE	9F	002EB	PUSHAB	OUT_LENGTH
		F8	AD	9F	002EE	PUSHAB	CTL_STR_DSC
64		04	FB	002F1	CALLS	#4, SYS\$FAO	
05		50	E8	002F4	BLBS	FAO STATUS, 24\$	
		50	DD	002F7	PUSHL	FAO STATUS	
63		01	FB	002F9	CALLS	#1, LIB\$STOP	
	F4	AD	DD	002FC	24\$:	PUSHL	TEMP_STR_DSC+4
	14	AE	9F	002FF	PUSHAB	OUT_LENGTH	
	44	AE	9F	00302	PUSHAB	LINE_DESC	
65		03	FB	00305	CALLS	#3, STR\$COPY_R	
79		50	E9	00308	BLBC	BUILD STATUS, 29\$	
	3C	AE	9F	0030B	PUSHAB	LINE_DESC	
		52	DD	0030E	PUSHL	R2	
66		02	FB	00310	CALLS	#2, PRINT	
79		50	E9	00313	25\$:	BLBC	PRINT STATUS, 30\$
F8	AD	010E0035	8F	DO	00316	MOVL	#17694773, CTL_STR_DSC

.....
0462
0464
.....
0466
.....
0462
0472
.....
0473

FC	AD	0000'	CF	9E	0031E	MOVAB	P.AAM, CTL_STR_DSC+4
			20	DD	00324	PUSHL	#32
		F0	AD	9F	00326	PUSHAB	TEMP_STR_DSC
		18	AE	9F	00329	PUSHAB	OUT_LENGTH
		F8	AD	9F	0032C	PUSHAB	CTL_STR_DSC
64			04	FB	0032F	CALLS	#4, -SYS\$FAO
05			50	E8	00332	BLBS	FAO_STATUS, 26\$
			50	DD	00335	PUSHL	FAO_STATUS
63			01	FB	00337	CALLS	#1, -LIB\$STOP
		F4	AD	DD	0033A	PUSHL	TEMP_STR_DSC+4
		14	AE	9F	0033D	PUSHAB	OUT_LENGTH
		44	AE	9F	00340	PUSHAB	LINE_DESC
65			03	FB	00343	CALLS	#3, STR\$COPY_R
79			50	E9	00346	BLBC	BUILD_STATUS, 27\$
		3C	AE	9F	00349	PUSHAB	LINE_DESC
			52	DD	0034C	PUSHL	R2
66			02	FB	0034E	CALLS	#2, PRINT
79			50	E9	00351	BLBC	PRINT_STATUS, 33\$
F8	AD	010E0031	8F	DO	00354	MOVL	#17694769, CTL_STR_DSC
FC	AD	0000'	CF	9E	0035C	MOVAB	P.AAN, CTL_STR_DSC+4
			20	DD	00362	PUSHL	#32
		F0	AD	9F	00364	PUSHAB	TEMP_STR_DSC
		18	AE	9F	00367	PUSHAB	OUT_LENGTH
		F8	AD	9F	0036A	PUSHAB	CTL_STR_DSC
64			04	FB	0036D	CALLS	#4, -SYS\$FAO
05			50	E8	00370	BLBS	FAO_STATUS, 28\$
			50	DD	00373	PUSHL	FAO_STATUS
63			01	FB	00375	CALLS	#1, -LIB\$STOP
		F4	AD	DD	00378	PUSHL	TEMP_STR_DSC+4
		14	AE	9F	0037B	PUSHAB	OUT_LENGTH
		44	AE	9F	0037E	PUSHAB	LINE_DESC
65			03	FB	00381	CALLS	#3, STR\$COPY_R
79			50	E9	00384	BLBC	BUILD_STATUS, 29\$
		3C	AE	9F	00387	PUSHAB	LINE_DESC
			52	DD	0038A	PUSHL	R2
66			02	FB	0038C	CALLS	#2, PRINT
79			50	E9	0038F	BLBC	PRINT_STATUS, 30\$
F8	AD	010E0037	8F	DO	00392	MOVL	#17694775, CTL_STR_DSC
FC	AD	0000'	CF	9E	0039A	MOVAB	P.AAO, CTL_STR_DSC+4
			20	DD	003A0	PUSHL	#32
		F0	AD	9F	003A2	PUSHAB	TEMP_STR_DSC
		18	AE	9F	003A5	PUSHAB	OUT_LENGTH
		F8	AD	9F	003A8	PUSHAB	CTL_STR_DSC
64			04	FB	003AB	CALLS	#4, -SYS\$FAO
05			50	E8	003AE	BLBS	FAO_STATUS, 31\$
			50	DD	003B1	PUSHL	FAO_STATUS
63			01	FB	003B3	CALLS	#1, -LIB\$STOP
		F4	AD	DD	003B6	PUSHL	TEMP_STR_DSC+4
		14	AE	9F	003B9	PUSHAB	OUT_LENGTH
		44	AE	9F	003BC	PUSHAB	LINE_DESC
65			03	FB	003BF	CALLS	#3, STR\$COPY_R
79			50	E9	003C2	BLBC	BUILD_STATUS, 32\$
		3C	AE	9F	003C5	PUSHAB	LINE_DESC
			52	DD	003C8	PUSHL	R2
66			02	FB	003CA	CALLS	#2, PRINT
79			50	E9	003CD	BLBC	PRINT_STATUS, 33\$
F8	AD	010E0045	8F	DO	003D0	MOVL	#17694789, CTL_STR_DSC

0474

0475

0476

FC	AD	0000'	CF	9E	003D8	MOVAB	P.AAP, CTL_STR_DSC+4
			20	DD	003DE	PUSHL	#32
		F0	AD	9F	003E0	PUSHAB	TEMP_STR_DSC
		18	AE	9F	003E3	PUSHAB	OUT_LENGTH
		F8	AD	9F	003E6	PUSHAB	CTL_STR_DSC
64			04	FB	003E9	CALLS	#4, -SYS\$FAO
05			50	E8	003EC	BLBS	FAO_STATUS, 34\$
			50	DD	003EF	PUSHL	FAO_STATUS
63			01	FB	003F1	CALLS	#1, -LIB\$STOP
		F4	AD	DD	003F4	PUSHL	TEMP_STR_DSC+4
		14	AE	9F	003F7	PUSHAB	OUT_LENGTH
		44	AE	9F	003FA	PUSHAB	LINE_DESC
65			03	FB	003FD	CALLS	#3, STR\$COPY_R
79			50	E9	00400	BLBC	BUILD_STATUS, 41\$
		3C	AE	9F	00403	PUSHAB	LINE_DESC
			52	DD	00406	PUSHL	R2
66			02	FB	00408	CALLS	#2, PRINT
79			50	E9	0040B	BLBC	PRINT_STATUS, 42\$
F8	AD	010E0035	8F	DO	0040E	MOVL	#17694773, CTL_STR_DSC
FC	AD	0000'	CF	9E	00416	MOVAB	P.AAQ, CTL_STR_DSC+4
			20	DD	0041C	PUSHL	#32
		F0	AD	9F	0041E	PUSHAB	TEMP_STR_DSC
		18	AE	9F	00421	PUSHAB	OUT_LENGTH
		F8	AD	9F	00424	PUSHAB	CTL_STR_DSC
64			04	FB	00427	CALLS	#4, -SYS\$FAO
05			50	E8	0042A	BLBS	FAO_STATUS, 37\$
			50	DD	0042D	PUSHL	FAO_STATUS
63			01	FB	0042F	CALLS	#1, -LIB\$STOP
		F4	AD	DD	00432	PUSHL	TEMP_STR_DSC+4
		14	AE	9F	00435	PUSHAB	OUT_LENGTH
		44	AE	9F	00438	PUSHAB	LINE_DESC
65			03	FB	0043B	CALLS	#3, STR\$COPY_R
79			50	E9	0043E	BLBC	BUILD_STATUS, 44\$
		3C	AE	9F	00441	PUSHAB	LINE_DESC
			52	DD	00444	PUSHL	R2
66			02	FB	00446	CALLS	#2, PRINT
79			50	E9	00449	BLBC	PRINT_STATUS, 45\$
F8	AD	010E0039	8F	DO	0044C	MOVL	#17694777, CTL_STR_DSC
FC	AD	0000'	CF	9E	00454	MOVAB	P.AAR, CTL_STR_DSC+4
			20	DD	0045A	PUSHL	#32
		F0	AD	9F	0045C	PUSHAB	TEMP_STR_DSC
		18	AE	9F	0045F	PUSHAB	OUT_LENGTH
		F8	AD	9F	00462	PUSHAB	CTL_STR_DSC
64			04	FB	00465	CALLS	#4, -SYS\$FAO
05			50	E8	00468	BLBS	FAO_STATUS, 40\$
			50	DD	0046B	PUSHL	FAO_STATUS
63			01	FB	0046D	CALLS	#1, -LIB\$STOP
		F4	AD	DD	00470	PUSHL	TEMP_STR_DSC+4
		14	AE	9F	00473	PUSHAB	OUT_LENGTH
		44	AE	9F	00476	PUSHAB	LINE_DESC
65			03	FB	00479	CALLS	#3, STR\$COPY_R
79			50	E9	0047C	BLBC	BUILD_STATUS, 47\$
		3C	AE	9F	0047F	PUSHAB	LINE_DESC
			52	DD	00482	PUSHL	R2
66			02	FB	00484	CALLS	#2, PRINT
79			50	E9	00487	BLBC	PRINT_STATUS, 48\$
F8	AD	010E0040	8F	DO	0048A	MOVL	#17694784, CTL_STR_DSC

0477

0478

0479

FC	AD	0000'	CF	9E	00492	MOVAB	P.AAS, CTL_STR_DSC+4
			20	DD	00498	PUSHL	#32
		F0	AD	9F	0049A	PUSHAB	TEMP_STR_DSC
		18	AE	9F	0049D	PUSHAB	OUT_LENGTH
		F8	AD	9F	004A0	PUSHAB	CTL_STR_DSC
64			04	FB	004A3	CALLS	#4, -SYS\$FAO
05			50	E8	004A6	BLBS	FAO_STATUS, 43\$
			50	DD	004A9	PUSHL	FAO_STATUS
63			01	FB	004AB	CALLS	#1, -LIB\$STOP
		F4	AD	DD	004AE	43\$: PUSHL	TEMP_STR_DSC+4
		14	AE	9F	004B1	PUSHAB	OUT_LENGTH
		44	AE	9F	004B4	PUSHAB	LINE_DESC
65			03	FB	004B7	CALLS	#3, STR\$COPY_R
79			50	E9	004BA	44\$: BLBC	BUILD_STATUS, 50\$
		3C	AE	9F	004BD	PUSHAB	LINE_DESC
			52	DD	004C0	PUSHL	R2
66			02	FB	004C2	CALLS	#2, PRINT
79			50	E9	004C5	45\$: BLBC	PRINT_STATUS, 51\$
F8	AD	010E0041	8F	DD	004C8	MOVL	#17694785, CTL_STR_DSC
FC	AD	0000'	CF	9E	004D0	MOVAB	P.AAT, CTL_STR_DSC+4
			20	DD	004D6	PUSHL	#32
		F0	AD	9F	004DB	PUSHAB	TEMP_STR_DSC
		18	AE	9F	004DB	PUSHAB	OUT_LENGTH
		F8	AD	9F	004DE	PUSHAB	CTL_STR_DSC
64			04	FB	004E1	CALLS	#4, -SYS\$FAO
05			50	E8	004E4	BLBS	FAO_STATUS, 46\$
			50	DD	004E7	PUSHL	FAO_STATUS
63			01	FB	004E9	CALLS	#1, -LIB\$STOP
		F4	AD	DD	004EC	46\$: PUSHL	TEMP_STR_DSC+4
		14	AE	9F	004EF	PUSHAB	OUT_LENGTH
		44	AE	9F	004F2	PUSHAB	LINE_DESC
65			03	FB	004F5	CALLS	#3, STR\$COPY_R
79			50	E9	004F8	47\$: BLBC	BUILD_STATUS, 53\$
		3C	AE	9F	004FB	PUSHAB	LINE_DESC
			52	DD	004FE	PUSHL	R2
66			02	FB	00500	CALLS	#2, PRINT
79			50	E9	00503	48\$: BLBC	PRINT_STATUS, 54\$
F8	AD	010E0042	8F	DD	00506	MOVL	#17694786, CTL_STR_DSC
FC	AD	0000'	CF	9E	0050E	MOVAB	P.AAU, CTL_STR_DSC+4
			20	DD	00514	PUSHL	#32
		F0	AD	9F	00516	PUSHAB	TEMP_STR_DSC
		18	AE	9F	00519	PUSHAB	OUT_LENGTH
		F8	AD	9F	0051C	PUSHAB	CTL_STR_DSC
64			04	FB	0051F	CALLS	#4, -SYS\$FAO
05			50	E8	00522	BLBS	FAO_STATUS, 49\$
			50	DD	00525	PUSHL	FAO_STATUS
63			01	FB	00527	CALLS	#1, -LIB\$STOP
		F4	AD	DD	0052A	49\$: PUSHL	TEMP_STR_DSC+4
		14	AE	9F	0052D	PUSHAB	OUT_LENGTH
		44	AE	9F	00530	PUSHAB	LINE_DESC
65			03	FB	00533	CALLS	#3, STR\$COPY_R
79			50	E9	00536	50\$: BLBC	BUILD_STATUS, 56\$
		3C	AE	9F	00539	PUSHAB	LINE_DESC
			52	DD	0053C	PUSHL	R2
66			02	FB	0053E	CALLS	#2, PRINT
79			50	E9	00541	51\$: BLBC	PRINT_STATUS, 57\$
F8	AD	010E0043	8F	DD	00544	MOVL	#17694787, CTL_STR_DSC

0480

0481

0482

FC	AD	0000'	CF	9E	0054C	MOVAB	P.AAV, CTL_STR_DSC+4
			20	DD	00552	PUSHL	#32
		F0	AD	9F	00554	PUSHAB	TEMP_STR_DSC
		18	AE	9F	00557	PUSHAB	OUT_LENGTH
		F8	AD	9F	0055A	PUSHAB	CTL_STR_DSC
64			04	FB	0055D	CALLS	#4, -SYS\$FAO
05			50	E8	00560	BLBS	FAO_STATUS, 52\$
			50	DD	00563	PUSHL	FAO_STATUS
63			01	FB	00565	CALLS	#1, -LIB\$STOP
		F4	AD	DD	00568	52\$: PUSHL	TEMP_STR_DSC+4
		14	AE	9F	0056B	PUSHAB	OUT_LENGTH
		44	AE	9F	0056E	PUSHAB	LINE_DESC
65			03	FB	00571	CALLS	#3, STR\$COPY_R
79			50	E9	00574	53\$: BLBC	BUILD_STATUS, 59\$
		3C	AE	9F	00577	PUSHAB	LINE_DESC
			52	DD	0057A	PUSHL	R2
66			02	FB	0057C	CALLS	#2, PRINT
79			50	E9	0057F	54\$: BLBC	PRINT_STATUS, 60\$
F8	AD	010E004B	8F	DD	00582	MOVL	#17697795, CTL_STR_DSC
FC	AD	0000'	CF	9E	0058A	MOVAB	P.AAV, CTL_STR_DSC+4
			20	DD	00590	PUSHL	#32
		F0	AD	9F	00592	PUSHAB	TEMP_STR_DSC
		18	AE	9F	00595	PUSHAB	OUT_LENGTH
		F8	AD	9F	00598	PUSHAB	CTL_STR_DSC
64			04	FB	0059B	CALLS	#4, -SYS\$FAO
05			50	E8	0059E	BLBS	FAO_STATUS, 55\$
			50	DD	005A1	PUSHL	FAO_STATUS
63			01	FB	005A3	CALLS	#1, -LIB\$STOP
		F4	AD	DD	005A6	55\$: PUSHL	TEMP_STR_DSC+4
		14	AE	9F	005A9	PUSHAB	OUT_LENGTH
		44	AE	9F	005AC	PUSHAB	LINE_DESC
65			03	FB	005AF	CALLS	#3, STR\$COPY_R
46			50	E9	005B2	56\$: BLBC	BUILD_STATUS, 60\$
		3C	AE	9F	005B5	PUSHAB	LINE_DESC
			52	DD	005B8	PUSHL	R2
66			02	FB	005BA	CALLS	#2, PRINT
38			50	E9	005BD	57\$: BLBC	PRINT_STATUS, 60\$
F8	AD	010E0004	8F	DD	005C0	MOVL	#17697724, CTL_STR_DSC
FC	AD	0000'	CF	9E	005C8	MOVAB	P.AAX, CTL_STR_DSC+4
			20	DD	005CE	PUSHL	#32
		F0	AD	9F	005D0	PUSHAB	TEMP_STR_DSC
		18	AE	9F	005D3	PUSHAB	OUT_LENGTH
		F8	AD	9F	005D6	PUSHAB	CTL_STR_DSC
64			04	FB	005D9	CALLS	#4, -SYS\$FAO
05			50	E8	005DC	BLBS	FAO_STATUS, 58\$
			50	DD	005DF	PUSHL	FAO_STATUS
63			01	FB	005E1	CALLS	#1, -LIB\$STOP
		F4	AD	DD	005E4	58\$: PUSHL	TEMP_STR_DSC+4
		14	AE	9F	005E7	PUSHAB	OUT_LENGTH
		44	AE	9F	005EA	PUSHAB	LINE_DESC
65			03	FB	005ED	CALLS	#3, STR\$COPY_R
62			50	E9	005F0	59\$: BLBC	BUILD_STATUS, 67\$
		3C	AE	9F	005F3	PUSHAB	LINE_DESC
			52	DD	005F6	PUSHL	R2
66			02	FB	005F8	CALLS	#2, PRINT
57			50	E9	005FB	60\$: BLBC	PRINT_STATUS, 67\$
		3C	AE	9F	005FE	PUSHAB	LINE_DESC

0483

0488

0492

67		01	FB	00601	CALLS	#1, STR\$FREE1 DX	:
05		50	E8	00604	BLBS	FREE_STATUS, 81\$:
		50	DD	00607	PUSHL	FREE_STATUS	:
63		01	FB	00609	CALLS	#1, [IB\$STOP	:
	34	AE	9F	0060C	PUSHAB	PRINTABLE_DESC	0493
67		01	FB	0060F	CALLS	#1, STR\$FREE1 DX	:
05		50	E8	00612	BLBS	FREE_STATUS, 82\$:
		50	DD	00615	PUSHL	FREE_STATUS	:
63		01	FB	00617	CALLS	#1, [IB\$STOP	:
	2C	AE	9F	0061A	PUSHAB	HEX_DESC	0494
67		01	FB	0061D	CALLS	#1, STR\$FREE1 DX	:
05		50	E8	00620	BLBS	FREE_STATUS, 83\$:
		50	DD	00623	PUSHL	FREE_STATUS	:
63		01	FB	00625	CALLS	#1, [IB\$STOP	:
	24	AE	9F	00628	PUSHAB	TEXT	0495
67		01	FB	0062B	CALLS	#1, STR\$FREE1 DX	:
05		50	E8	0062E	BLBS	FREE_STATUS, 84\$:
		50	DD	00631	PUSHL	FREE_STATUS	:
63		01	FB	00633	CALLS	#1, [IB\$STOP	:
	1C	AE	9F	00636	PUSHAB	NAME	0496
67		01	FB	00639	CALLS	#1, STR\$FREE1 DX	:
05		50	E8	0063C	BLBS	FREE_STATUS, 85\$:
		50	DD	0063F	PUSHL	FREE_STATUS	:
63		01	FB	00641	CALLS	#1, [IB\$STOP	:
	14	AE	9F	00644	PUSHAB	SEVERITY	0497
67		01	FB	00647	CALLS	#1, STR\$FREE1 DX	:
05		50	E8	0064A	BLBS	FREE_STATUS, 86\$:
		50	DD	0064D	PUSHL	FREE_STATUS	:
63		01	FB	0064F	CALLS	#1, [IB\$STOP	:
50		01	DD	00652	MOVL	#1, R0	0498
		04	00655	67\$:	RET		0499

; Routine Size: 1622 bytes, Routine Base: \$CODE\$ + 0162

```

505 0500 1 %SBTTL 'PRINT - print a text line on a file'
506 0501 1 ROUTINE PRINT (          ! Print a text line on a file
507 0502 1     RAB_ADDR          ! RAB onto which to print the line
508 0503 1     TEXT_LINE       ! Descriptor of the line to print
509 0504 1 ) =
510 0505 1
511 0506 1 +-+
512 0507 1 FUNCTIONAL DESCRIPTION:
513 0508 1
514 0509 1     This routine interfaces to RMS to print a line of text.
515 0510 1
516 0511 1 CALLING SEQUENCE:
517 0512 1
518 0513 1     ret_status.wlc.v = PRINT (RAB_ADDR.mz.r, TEXT_LINE.rt.dx)
519 0514 1
520 0515 1 FORMAL PARAMETERS:
521 0516 1
522 0517 1     RAB_ADDR          Pointer to the RAB onto which to print the line of text
523 0518 1     TEXT_LINE       Descriptor for the line of text to be printed.
524 0519 1
525 0520 1 IMPLICIT INPUTS:
526 0521 1
527 0522 1     NONE
528 0523 1
529 0524 1 IMPLICIT OUTPUTS:
530 0525 1
531 0526 1     NONE
532 0527 1
533 0528 1 COMPLETION STATUS:
534 0529 1
535 0530 1     $$$_NORMAL      Normal successful completion
536 0531 1     All RMS errors are returned to the caller, so that they can be
537 0532 1     signalled with the file name.
538 0533 1
539 0534 1 SIDE EFFECTS:
540 0535 1
541 0536 1     Does a $PUT to the RAB.
542 0537 1
543 0538 1 --
544 0539 1
545 0540 2 BEGIN
546 0541 2
547 0542 2 MAP
548 0543 2     RAB_ADDR : REF $RAB DECL,
549 0544 2     TEXT_LINE : REF BLOCK [8, BYTE];
550 0545 2
551 0546 2 LOCAL
552 0547 2     PUT_STATUS;
553 0548 2
554 0549 2 +-+
555 0550 2 Fill in the RAB fields.
556 0551 2 -
557 0552 2     RAB_ADDR [RAB$W_RSZ] = .TEXT_LINE [DSC$W_LENGTH];
558 0553 2     RAB_ADDR [RAB$L_RBF] = .TEXT_LINE [DSC$A_POINTER];
559 0554 2 +-+
560 0555 2 Now do the $PUT
561 0556 2 -

```

BASMSG
2-003

BASMSG - write BASMSG.MSG
PRINT - print a text line on a file

F 11
16-Sep-1984 01:42:08
14-Sep-1984 11:55:22

VAX-11 Bliss-32 V4.0-742
[BASRTL.SRC]BASMSGGEN.B32;1

Page 26
(6)

```

: 562      0557 2      PUT_STATUS = $PUT (RAB = .RAB_ADDR);
: 563      0558 2
: 564      0559 2      IF ( NOT .PUT_STATUS) THEN RETURN (.PUT_STATUS);
: 565      0560 2
: 566      0561 2      RETURN (SS$_NORMAL);
: 567      0562 1      END;
                                ! End of routine PRINT

```

```

                                .EXTRN  SYSSPUT
                                0000 00000 PRINT: .WORD  Save nothing
                                51      04  AC  D0 00002      MOVL  RAB_ADDR, R1
                                50      08  AC  D0 00006      MOVL  TEXT_LINE, R0
                                22  A1   60  B0 0000A      MOVW  (R0), 34(R1)
                                28  A1   A0  D0 0000E      MOVL  4(R0), 40(R1)
                                00000000G 00  51  DD 00013      PUSHL R1
                                03      01  FB 00015      CALLS #1, SYSSPUT
                                50      50  E9 0001C      BLBC  PUT_STATUS, 1$
                                50      01  D0 0001F      MOVL  #1, R0
                                04 00022 1$:  RET
                                : 0501
                                : 0552
                                :
                                : 0553
                                : 0557
                                :
                                : 0559
                                : 0561
                                : 0562

```

; Routine Size: 35 bytes, Routine Base: \$CODE\$ + 07B8

```

569 0563 1 %SBTTL 'HEX_TEXT - Return a binary string in hexadecimal'
570 0564 1 ROUTINE HEX_TEXT (
571 0565 1     OUTPUT_DESC,
572 0566 1     INPUT_LEN,
573 0567 1     INPUT_ADDR
574 0568 1 ) =
575 0569 1
576 0570 1
577 0571 1 **
578 0572 1 FUNCTIONAL DESCRIPTION:
579 0573 1     This routine converts an arbitrary string of bytes into hex, so it
580 0574 1     can be printed.  Early bytes are put to the right of later bytes.
581 0575 1
582 0576 1 CALLING SEQUENCE:
583 0577 1
584 0578 1     status.wlc.v = HEX_TEXT (OUTPUT_desc.wt.dx, INPUT_LEN.rl.v, INPUT_ADDR.ra.v)
585 0579 1
586 0580 1 FORMAL PARAMETERS:
587 0581 1
588 0582 1     output_desc      Where the result text is stored.
589 0583 1     input_len        Number of bytes of input
590 0584 1     input_addr       Address of first input byte
591 0585 1
592 0586 1 IMPLICIT INPUTS:
593 0587 1
594 0588 1     NONE
595 0589 1
596 0590 1 IMPLICIT OUTPUTS:
597 0591 1
598 0592 1     NONE
599 0593 1
600 0594 1 COMPLETION STATUS:
601 0595 1
602 0596 1     $$$_NORMAL      Normal successful completion
603 0597 1     Any errors from STR$CONCAT
604 0598 1     Any errors from STR$COPY_DX
605 0599 1
606 0600 1 SIDE EFFECTS:
607 0601 1
608 0602 1     Calls STR$CONCAT and STR$COPY_DX, thus manipulating string storage.
609 0603 1
610 0604 1 --
611 0605 1
612 0606 2 BEGIN
613 0607 2
614 0608 2 MAP
615 0609 2     INPUT_ADDR : REF VECTOR [, BYTE],
616 0610 2     OUTPUT_DESC : REF BLOCK [8, BYTE];
617 0611 2
618 0612 2 LOCAL
619 0613 2     INTER_DESC : BLOCK [8, BYTE],
620 0614 2     DIGIT_DESC : BLOCK [8, BYTE],
621 0615 2     DIGIT,
622 0616 2     STATUS;
623 0617 2
624 0618 2 INIT_DESCRIPTOR (INTER_DESC);
625 0619 2 DIGIT_DESC [DSC$W_LENGTH] = 1;

```

```

: 626      0620 2      DIGIT_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
: 627      0621 2      DIGIT_DESC [DSC$B_CLASS] = DSC$K_CLASS_S;
: 628      0622 2      DIGIT_DESC [DSC$A_POINTER] = DIGIT;
: 629      0623 2
: 630      0624 2      INCR CHAR_NO FROM 1 TO .INPUT_LEN DO
: 631      0625 2      BEGIN
: 632      0626 2
: 633      0627 2      LOCAL
: 634      0628 2      CHAR;
: 635      0629 2
: 636      0630 2      CHAR = .INPUT_ADDR [.CHAR_NO - 1];
: 637      0631 2      DIGIT = (.CHAR AND 15) + '0';
: 638      0632 2
: 639      0633 2      IF (.DIGIT GTR '9') THEN DIGIT = .DIGIT - 10 - '0' + 'A';
: 640      0634 2
: 641      0635 2      STATUS = STR$CONCAT (INTER_DESC, DIGIT_DESC, INTER_DESC);
: 642      0636 2
: 643      0637 2      IF ( NOT .STATUS) THEN RETURN (.STATUS);
: 644      0638 2
: 645      0639 2      DIGIT = (.CHAR^-4) + '0';
: 646      0640 2
: 647      0641 2      IF (.DIGIT GTR '9') THEN DIGIT = .DIGIT - 10 - '0' + 'A';
: 648      0642 2
: 649      0643 2      STATUS = STR$CONCAT (INTER_DESC, DIGIT_DESC, INTER_DESC);
: 650      0644 2
: 651      0645 2      IF ( NOT .STATUS) THEN RETURN (.STATUS);
: 652      0646 2
: 653      0647 2      END;
: 654      0648 2
: 655      0649 2      STATUS = STR$COPY DX (.OUTPUT_DESC, INTER_DESC);
: 656      0650 2      DISCARD_DESCRIPTOR (INTER_DESC);
: 657      0651 2      RETURN (.STATUS);
: 658      0652 1      END;

```

! End of routine HEX_TEXT

003C 0000 HEX_TEXT:

					.WORD	Save R2,R3,R4,R5	: 0564
		55	00000000G	00	9E	00002	
		5E		14	C2	00009	
	0C	AE	020E0000	8F	D0	0000C	MOVAB STR\$CONCAT, R5
			10	AE	D4	00014	SUBL2 #20, SP
	04	AE	010E0001	8F	D0	00017	MOVL #34471936, INTER_DESC
	08	AE		6E	9E	0001F	CLRL INTER_DESC+4
				53	D4	00023	MOVL #17694721, DIGIT_DESC
				4E	11	00025	MOVAB DIGIT, DIGIT_DESC+4
				AC	C1	00027	CLRL CHAR_NO
	50		0C	AC	C1	00027	BRB 4\$
			FF	A0	9A	0002C	ADDL3 INPUT_ADDR, CHAR_NO, R0
6E	52			00	EF	00030	MOVZBL -1(R0), CHAR
				30	C0	00035	EXTZV #0, #4, CHAR, DIGIT
				6E	D1	00038	ADDL2 #48, DIGIT
				03	15	0003B	CML DIGIT, #57
				07	C0	0003D	BLEQ 2\$
			0C	AE	9F	00040	ADDL2 #7, DIGIT
			08	AE	9F	00043	PUSHAB INTER_DESC
							PUSHAB DIGIT_DESC
							: 0618
							: 0619
							: 0622
							: 0630
							: 0631
							: 0633
							: 0635

		14	AE 9F 00046		PUSHAB INTER_DESC	
	65		03 FB 00049		CALLS #3, STR\$CONCAT	
	54		50 D0 0004C		MOVL R0, STATUS	
	4E		54 E9 0004F		BLBC STATUS, 5\$	0637
52	52	FC	8F 78 00052		ASHL #4, CHAR, R2	0639
	6E	30	A2 9E 00057		MOVAB 48(R2), DIGIT	
	39		6E D1 0005B		CMPL DIGIT, #57	0641
			03 15 0005E		BLEQ 3\$	
	6E		07 C0 00060		ADDL2 #7, DIGIT	
		0C	AE 9F 00063	3\$:	PUSHAB INTER_DESC	0643
		08	AE 9F 00066		PUSHAB DIGIT_DESC	
		14	AE 9F 00069		PUSHAB INTER_DESC	
	65		03 FB 0006C		CALLS #3, STR\$CONCAT	
	54		50 D0 0006F		MOVL R0, STATUS	
	2B		54 E9 00072		BLBC STATUS, 5\$	0645
AD	53	08	AC F3 00075	4\$:	AOBLEQ INPUT_LEN, CHAR_NO, 1\$	0624
		0C	AE 9F 0007A		PUSHAB INTER_DESC	0649
		04	AC DD 0007D		PUSHL OUTPUT_DESC	
	00000000G	00	02 FB 00080		CALLS #2, STR\$COPY_DX	
		54	50 D0 00087		MOVL R0, STATUS	
		0C	AE 9F 0008A		PUSHAB INTER_DESC	0650
	00000000G	00	01 FB 0008D		CALLS #1, STR\$FREE1_DX	
		09	50 E8 00094		BLBS FREE_STATUS, 5\$	
			50 DD 00097		PUSHL FREE_STATUS	
	00000000G	00	01 FB 00099		CALLS #1, CIB\$STOP	
		50	54 D0 000A0	5\$:	MOVL STATUS, R0	0651
			04 000A3		RET	0652

; Routine Size: 164 bytes, Routine Base: \$CODE\$ + 07DB

```

660 0653 1 %SBTTL 'PRINTABLE_TEXT - Return a binary string in ASCII, printable'
661 0654 1 ROUTINE PRINTABLE_TEXT (
662 0655 1     OUTPUT_DESC,
663 0656 1     INPUT_LEN,
664 0657 1     INPUT_ADDR
665 0658 1 ) =
666 0659 1
667 0660 1 ++
668 0661 1 FUNCTIONAL DESCRIPTION:
669 0662 1
670 0663 1     This routine converts an arbitrary string of bytes into ASCII, representing
671 0664 1     unprintable characters in hexadecimal so the result can be printed.
672 0665 1
673 0666 1 CALLING SEQUENCE:
674 0667 1
675 0668 1     status.wlc.v = PRINTABLE_TEXT (OUTPUT_desc.wt.dx, INPUT_LEN.rl.v, INPUT_ADDR.ra.v)
676 0669 1
677 0670 1 FORMAL PARAMETERS:
678 0671 1
679 0672 1     OUTPUT_DESC     Where the result text is stored.
680 0673 1     INPUT_LEN       Number of bytes of input
681 0674 1     INPUT_ADDR      Address of first input byte
682 0675 1
683 0676 1 IMPLICIT INPUTS:
684 0677 1
685 0678 1     NONE
686 0679 1
687 0680 1 IMPLICIT OUTPUTS:
688 0681 1
689 0682 1     NONE
690 0683 1
691 0684 1 COMPLETION STATUS:
692 0685 1
693 0686 1     $$$_NORMAL      Normal successful completion
694 0687 1     Any errors from STR$CONCAT
695 0688 1     Any errors from STR$COPY_DX
696 0689 1
697 0690 1 SIDE EFFECTS:
698 0691 1
699 0692 1     Calls STR$CONCAT and STR$COPY_DX, thus manipulating string storage.
700 0693 1
701 0694 1 --
702 0695 1
703 0696 2 BEGIN
704 0697 2
705 0698 2 MAP
706 0699 2     INPUT_ADDR : REF VECTOR [ , BYTE],
707 0700 2     OUTPUT_DESC : REF BLOCK [8, BYTE];
708 0701 2
709 0702 2 LOCAL
710 0703 2     INTER_DESC : BLOCK [8, BYTE],
711 0704 2     CHAR_DESC  : BLOCK [8, BYTE],
712 0705 2     CHAR_REP   : VECTOR [4, BYTE],
713 0706 2     STATOS;
714 0707 2
715 0708 2 INIT_DESCRIPTOR (INTER_DESC);
716 0709 2 CHAR_DESC [DSC$W_LENGTH] = 1;

```



```

717 0710 2 CHAR_DESC [DSC$B_DTYPE] = DSC$K_DTYPE
718 0711 2 CHAR_DESC [DSC$B_CLASS] = DSC$K_CLASS
719 0712 2 CHAR_DESC [DSC$A_POINTER] = CHAR_REP [0];
720 0713 2
721 0714 2 INCR CHAR_NO FROM 1 TO .INPUT_LEN DO
722 0715 2 BEGIN
723 0716 2
724 0717 2 LOCAL
725 0718 2 CHAR;
726 0719 2
727 0720 2 CHAR = .INPUT_ADDR [.CHAR_NO - 1];
728 0721 2
729 0722 4 IF ((.CHAR GEQ %X'20') AND
730 0723 4 (.CHAR LSS %X'7F') AND
731 0724 4 (.CHAR NEQ '<') AND
732 0725 4 (.CHAR NEQ '''))
733 0726 3 THEN
734 0727 4 BEGIN
735 0728 4 !+
736 0729 4 !- Show character as itself.
737 0730 4
738 0731 4 CHAR_REP [0] = .CHAR;
739 0732 4 CHAR_DESC [DSC$W_LENGTH] = 1;
740 0733 4 END
741 0734 3 ELSE
742 0735 4 BEGIN
743 0736 4 !+
744 0737 4 !- The character is not printable. Represent it by <>. To avoid
745 0738 4 !- ambiguity, '<' and '' are also represented this way. Control characters
746 0739 4 !- SOH through SUB are represented by <^letter>; others characters by <hex>.
747 0740 4 !-
748 0741 4 CHAR_REP [0] = '<';
749 0742 4
750 0743 5 IF (((.CHAR + %X'40') GEQ 'A') AND ((.CHAR + %X'40') LEQ 'Z'))
751 0744 4 THEN
752 0745 5 BEGIN
753 0746 5 CHAR_REP [1] = '^';
754 0747 5 CHAR_REP [2] = .CHAR + %X'40';
755 0748 5 END
756 0749 4 ELSE
757 0750 5 BEGIN
758 0751 5 LOCAL
759 0752 5 DIGIT;
760 0753 5
761 0754 5 DIGIT = (.CHAR^-4) + '0';
762 0755 5
763 0756 5 IF (.DIGIT GTR '9') THEN DIGIT = .DIGIT - 10 - '0' + 'A';
764 0757 5
765 0758 5 CHAR_REP [1] = .DIGIT;
766 0759 5 DIGIT = (.CHAR AND 15) + '0';
767 0760 5
768 0761 5 IF (.DIGIT GTR '9') THEN DIGIT = .DIGIT - 10 - '0' + 'A';
769 0762 5
770 0763 5 CHAR_REP [2] = .DIGIT;
771 0764 5
772 0765 4 END;
773 0766 4

```

```

: 774 0767 4          CHAR_REP [3] = '>';
: 775 0768 4          CHAR_DESC [DSC$W_LENGTH] = 4;
: 776 0769 3          END;
: 777 0770 3
: 778 0771 3          STATUS = STR$CONCAT (INTER_DESC, INTER_DESC, CHAR_DESC);
: 779 0772 3
: 780 0773 3          IF ( NOT .STATUS) THEN RETURN (.STATUS);
: 781 0774 2
: 782 0775 2          END;
: 783 0776 2
: 784 0777 2          STATUS = STR$COPY DX (.OUTPUT_DESC, INTER_DESC);
: 785 0778 2          DISCARD_DESCRIPTOR (INTER_DESC);
: 786 0779 2          RETURN (.STATUS);
: 787 078C 1          END;
                                ! End of routine PRINTABLE_TEXT

```

```

                                000C 0000 PRINTABLE_TEXT:
                                .WORD      Save R2,R3
                                SUBL2      #20, SP
                                MOVL       #34471936, INTER_DESC
                                CLRL       INTER_DESC+4
                                MOVL       #17694721, CHAR_DESC
                                MOVAB      CHAR_REP, CHAR_DESC+4
                                CLRL       CHAR_NO
                                BRW        7$
50                                52      OC 0090 AC C1 00021 1$: ADDL3      INPUT_ADDR, CHAR_NO, R0
                                51      FF AO 9A 00026 MOVZBL     -1(R0), CHAR
                                20      51 D1 0002A CMPL      CHAR, #32
                                0000007F 8F 51 D1 0002F CMPL      CHAR, #127
                                3C      51 D1 00038 CMPL      CHAR, #60
                                22      0E 13 0003B BEQL      2$
                                6E      51 D1 0003D CMPL      CHAR, #34
                                04      09 13 00040 BEQL      2$
                                6E      51 90 00042 MOVB      CHAR, CHAR_REP
                                04      AE 01 B0 00045 MOVW      #1, CHAR_DESC
                                6E      50 11 00049 BRB       6$
                                50      3C 90 0004B 2$: MOVB      #60, CHAR_REP
                                00000041 8F 40 A1 9E 0004E MOVAB     64(R1), R0
                                8F 50 D1 00052 CMPL      R0, #65
                                0000005A 8F 50 D1 0005B BLSS     3$
                                07 14 00062 CMPL      R0, #90
                                01      AE 5E 8F 90 00064 BGTR     3$
                                50      24 11 00069 MOVB      #94, CHAR_REP+1
                                51      51 FC 8F 78 0006B BRB       5$
                                50      30 C0 0007C 3$: ASHL     #-4, CHAR, R0
                                39      50 D1 00073 ADDL2     #48, DIGIT
                                03 15 00076 CMPL     DIGIT, #57
                                07 C0 00078 BLEQ     4$
                                01      50 07 C0 00078 ADDL2     #7, DIGIT
                                51      50 90 0007B 4$: MOVB     DIGIT, CHAR_REP+1
                                04      00 EF 0007F EXTZV    #0, #4, CHAR, DIGIT
                                50      30 C0 00084 ADDL2     #48, DIGIT

```

BASMSG
2-003

BASMSG - write BASMSG.MSG
PRINTABLE_TEXT - Return a binary string in ASCII

M 11
16-Sep-1984 01:42:08
14-Sep-1984 11:55:22

VAX-11 Bliss-32 V4.0-742
[BASRTL.SRC]BASMSGGEN.B32;1

Page 33
(8)

		39		50	D1	00087		CMPL	DIGIT, #57		: 0762
				03	15	0008A		BLEQ	5\$: 0764
		50		07	C0	0008C		ADDL2	#7, DIGIT		: 0767
	02	AE		50	90	0008F	5\$:	MOVW	DIGIT, CHAR REP+2		: 0768
	03	AE		3E	90	00093		MOVW	#62, CHAR REP+3		: 0771
	04	AE		04	B0	00097		MOVW	#4, CHAR DESC		: 0773
			04	AE	9F	0009B	6\$:	PUSHAB	CHAR DESC		: 0774
			10	AE	9F	0009E		PUSHAB	INTER_DESC		: 0775
			14	AE	9F	000A1		PUSHAB	INTER_DESC		: 0776
	00000000G	00		03	FB	000A4		CALLS	#3, STR\$CONCAT		: 0777
		53		50	DO	000AB		MOVL	R0, STATUS		: 0778
		2D		53	E9	000AE		BLBC	STATUS, 8\$: 0779
FF69		01		08	AC	F1 000B1	7\$:	ACBL	INPUT_LEN, #1, CHAR_NO, 1\$: 0780
			0C	AE	9F	000B8		PUSHAB	INTER_DESC		: 0781
			04	AC	DD	000BB		PUSHL	OUTPUT_DESC		: 0782
	00000000G	00		02	FB	000BE		CALLS	#2, STR\$COPY_DX		: 0783
		53		50	DO	000C5		MOVL	R0, STATUS		: 0784
			0C	AE	9F	000C8		PUSHAB	INTER_DESC		: 0785
	00000000G	00		01	FB	000CB		CALLS	#1, STR\$FREE1_DX		: 0786
		09		50	E8	000D2		BLBS	FREE_STATUS, 8\$: 0787
				50	DD	000D5		PUSHL	FREE_STATUS		: 0788
	00000000G	00		01	FB	000D7		CALLS	#1, [IB\$STOP		: 0789
		50		53	DO	000DE	8\$:	MOVL	STATUS, R0		: 0790
		50		04	000E1			RET			: 0791

: Routine Size: 226 bytes, Routine Base: \$CODE\$ + 087F

: 788 0781 1 !<BLF/PAGE>

BASMSG
2-003

BASMSG - write BASMSG.MSG

PRINTABLE_TEXT - Return a binary string in ASCII

N 11

16-Sep-1984 01:42:08

14-Sep-1984 11:55.22

VAX-11 Bliss-32 V4.0-742

[BASRTL.SRC]BASMSGGEN.B32;1

Page 34

(9)

: 790 0782 1 END
: 791 0783 1
: 792 0784 0 ELUDOM

! End of module

.EXTRN LIB\$STOP

PSECT SUMMARY

Name	Bytes	Attributes
\$PLITS	1020	NOVEC,NOWRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
\$CODES	2401	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	106	1	581	00:01.0

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS\$:BASMSGGEN/OBJ=OBJ\$:BASMSGGEN MSRC\$:BASMSGGEN/UPDATE=(ENH\$:BASMSGGEN)

: Size: 2401 code + 1020 data bytes
: Run Time: 00:54.8
: Elapsed Time: 01:50.6
: Lines/CPU Min: 858
: Lexemes/CPU-Min: 21680
: Memory Used: 601 pages
: Compilation Complete

BASMTD
LIS

BASMLD01
LIS

BASNOTMP
LIS

BASMOVEAR
LIS

BASMSGDEF
LIS

BASMSGGEN
LIS

BASONECHR
LIS

BASMOVE
LIS

BASNUM
LIS

BASNAMEAS
LIS

BASNUM
LIS