

LL	IIIIII	TTTTTTTT	
LL	IIIIII	TTTTTTTT	
LL	II	TT	
LL	II	TT	
LL	II	TT	
LL	II	TT	
LL	II	TT	
LL	II	TT	
LL	II	TT	
LL	II	TT	
LL	II	TT	
LL	II	TT	
LL	II	TT	
LL	II	TT	
LL	II	TT	
LL	II	TT	
LLLLLLLLLL	IIIIII	TT
LLLLLLLLLL	IIIIII	TT

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
LL	II		SS
LL	II		SS
LL	II		SS
LLLLLLLLLL	IIIIII	SSSSSSSS	
LLLLLLLLLL	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 MODULE lit ( IDENT = 'V04-000'  
P 0002 0 %BLISS32C, ADDRESSING_MODE (EXTERNAL = LONG_RELATIVE,  
0003 0 NONEXTERNAL = LONG_RELATIVE)  
0004 0 ) =  
0005 1 BEGIN  
0006 1  
0007 1 *****  
0008 1 *  
0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *  
0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *  
0011 1 * ALL RIGHTS RESERVED. *  
0012 1 *  
0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *  
0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *  
0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *  
0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *  
0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *  
0018 1 * TRANSFERRED. *  
0019 1 *  
0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *  
0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *  
0022 1 * CORPORATION. *  
0023 1 *  
0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *  
0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *  
0026 1 *  
0027 1 *  
0028 1 *****  
0029 1  
0030 1 ++  
0031 1 FACILITY: DSR (Digital Standard RUNOFF) / DSRPLUS  
0032 1  
0033 1 ABSTRACT:  
0034 1  
0035 1 Processes the .LITERAL and .END LITERAL commands, as well  
0036 1 as the text within the literal.  
0037 1  
0038 1 ENVIRONMENT: Transportable  
0039 1  
0040 1 AUTHOR: R.W.Friday CREATION DATE: June, 1978  
0041 1  
0042 1
```

Revision History

B 1
16-Sep-1984 00:48:45
14-Sep-1984 13:06:53

44 0043 1 %SBTTL 'Revision History'
45 0044 1
46 0045 1
47 0046 1
48 0047 1
49 0048 1
50 0049 1
51 0050 1
52 0051 1
53 0052 1

MODIFIED BY:

003 RER00003 Ron Randall 07-Mar-1983
Global edit of all modules. Updated module names, idents,
copyright dates. Changed require files to BLISS library.

Module Level Declarations

```

: 55      0053 1 %SBTTL 'Module Level Declarations'
: 56      0054 1
: 57      0055 1
: 58      0056 1 : TABLE OF CONTENTS:
: 59      0057 1
: 60      0058 1 : INCLUDE FILES:
: 61      0059 1
: 62      0060 1 LIBRARY 'NXPORT:XPORT';      ! XPORT Library
: 63      0061 1 REQUIRE 'REQ:RNODEF';      ! RUNOFF variant definitions
: 64      0192 1
: 65      U 0193 1 %IF DSRPLUS %THEN
: 66      U 0194 1 LIBRARY 'REQ:DPLLIB';      ! DSRPLUS BLISS Library
: 67      0195 1 %ELSE
: 68      0196 1 LIBRARY 'REQ:DSRLIB';      ! DSR BLISS Library
: 69      0197 1 %FI
: 70      0198 1
: 71      0199 1
: 72      0200 1 : EXTERNAL REFERENCES:
: 73      0201 1
: 74      0202 1 EXTERNAL
: 75      0203 1     GCA : GCA DEFINITION,
: 76      0204 1     IRA : FIXED STRING,
: 77      0205 1     NUMPRM : NUMPRM DEFINE,
: 78      0206 1     SCA : SCA DEFINITION,
: 79      0207 1     SCALIT : VECTOR [SCA_CASE_SIZE];
: 80      0208 1
: 81      0209 1 EXTERNAL LITERAL      ! Error messages
: 82      0210 1     RNFELD,
: 83      0211 1     RNFINM,
: 84      0212 1     RNFLDE;
: 85      0213 1
: 86      0214 1 EXTERNAL ROUTINE
: 87      0215 1     ERMA,
: 88      0216 1     ERMB,
: 89      0217 1     OFT,
: 90      0218 1     STKFRM;

```

```

: 92 0219 1 GLOBAL ROUTINE LIT (HANDLER_CODE) : NOVALUE =
: 93 0220 1
: 94 0221 1 +-
: 95 0222 1 FUNCTIONAL DESCRIPTION:
: 96 0223 1
: 97 0224 1     See the ABSTRACT, above.
: 98 0225 1
: 99 0226 1 FORMAL PARAMETERS:
100 0227 1
101 0228 1     If HANDLER_CODE is zero, process text instead of commands.
102 0229 1
103 0230 1 IMPLICIT INPUTS:
104 0231 1
105 0232 1     NUMPRM contains a number, as processed by GETNUM.
106 0233 1
107 0234 1 IMPLICIT OUTPUTS:     None
108 0235 1
109 0236 1 ROUTINE VALUE:
110 0237 1 COMPLETION CODES:     None
111 0238 1
112 0239 1 SIDE EFFECTS:         None
113 0240 1 --
114 0241 1
115 0242 2     BEGIN
116 0243 2
117 0244 2     SELECTONE .HANDLER_CODE OF
118 0245 2     SET
119 0246 2
120 0247 2     [0] :
121 0248 3         BEGIN                               !Count text inside a literal.
122 0249 3
123 0250 3         LOCAL
124 0251 3             KOUNT;
125 0252 3
126 0253 3         IF .GCA_CLIT
127 0254 3         THEN
128 0255 3
129 0256 3             IF .GCA_LIT_COUNT LEQ 0
130 0257 3             THEN
131 0258 4                 BEGIN                               !Literal count is wrong (too big)
132 0259 4                 ERMB (RNFLDE, FALSE);
133 0260 4                 GCA_LITERAL = FALSE;                 !Turn off literal mode
134 0261 4                 STKFRM (-1);                         !Restore previous settings
135 0262 4                 OFT (.GCA_LIT_FLAG, 0);             !Restore flag status
136 0263 4
137 0264 4                 INCR I FROM 0 TO SCA_CASE_SIZE - 1 DO
138 0265 4                     SCA [I] = .SCALIT [I];         !Restore case rules.
139 0266 4
140 0267 4                 END
141 0268 3             ELSE
142 0269 3                 GCA_LITERAL = .GCA_LITERAL - 1;
143 0270 3
144 0271 2         END;
145 0272 2
146 0273 2     [H_LITERAL] :
147 0274 3         BEGIN
148 0275 3             GCA_LITERAL = TRUE;                         !Turn on literal mode.
```

```
149 0276 3 GCA_CLIT = FALSE; !Assume not a counted literal
150 0277 3 GCA_LIT_COUNT = 0;
151 0278 3 !Remember current flag recognition status.
152 0279 3 GCA_LIT_FLAG = .GCA_FLAG_CMD;
153 0280 3 !Turn off flags
154 0281 3 !Save current status
155 0282 3 STKFRM (0);
156 0283 3 OFT (H_NO_FLAGS_ALL, 0);
157 0284 3 !Turn off filling and justifying
158 0285 3 SCA_FILL = FALSE;
159 0286 3 SCA_JUSTIFY = FALSE;
160 0287 3 SCA_RM = 150;
161 0288 3 SCA_KER = TRUE;
162 0289 3
163 0290 3 !Save case rules.
164 0291 3 INCR I FROM 0 TO SCA_CASE_SIZE - 1 DO SCALIT [.I] = .SCA [.I];
165 0292 3
166 0293 3 INCR I FROM 0 TO SCA_CASE_SIZE - 1 DO SCA[.I] = 0;
167 0294 3
168 0295 3 !Validate the count, if given. If the count
169 0296 3 !is erroneous, the intended counted literal is
170 0297 3 !handled like an uncounted literal instead.
171 0298 3
172 0299 3 IF NOT .NUM_RESULT
173 0300 3 THEN
174 0301 3 RETURN;
175 0302 3
176 0303 3 IF .NUM_LENGTH NEQ 0
177 0304 3 THEN
178 0305 4 BEGIN !It's a counted literal
179 0306 4
180 0307 4 IF .NUM_VALUE LEQ 0
181 0308 4 THEN
182 0309 5 BEGIN
183 0310 5 ERMA (RNFIM, FALSE);
184 0311 5 RETURN;
185 0312 4 END;
186 0313 4
187 0314 4 GCA_CLIT = TRUE;
188 0315 4 GCA_LIT_COUNT = .NUM_VALUE;
189 0316 3 END;
190 0317 3
191 0318 2 END;
192 0319 2
193 0320 2 [H_END_LITERAL] :
194 0321 3 BEGIN
195 0322 3
196 0323 3 IF NOT .GCA_LITERAL
197 0324 3 THEN
198 0325 4 BEGIN !.END LITERAL out of context
199 0326 4
200 0327 4 IF .GCA_CLIT
201 0328 4 THEN
202 0329 5 BEGIN !Left over from mis-counted literal
203 0330 5 GCA_CLIT = FALSE;
204 0331 5 RETURN;
205 0332 4 END;
```

```

: 206 0333 4
: 207 0334 4      ERMA (RNFELD, FALSE);
: 208 0335 4      RETURN;
: 209 0336 4      END
: 210 0337 3      ELSE
: 211 0338 4      BEGIN                                !.END LITERAL in proper context
: 212 0339 4      GCA_LITERAL = FALSE;                !Turn off literal mode.
: 213 0340 4      !Restore flag rules, filling, and justifying
: 214 0341 4      OFT (.GCA_LIT_FLAG, 0);
: 215 0342 4      !Restore previous status
: 216 0343 4      STKFRM (-1);
: 217 0344 4
: 218 0345 4      INCR I FROM 0 TO SCA CASE SIZE - 1 DO
: 219 0346 4          SCA [I] = .SCALIT [I];          !Restore case rules.
: 220 0347 4
: 221 0348 3      END;
: 222 0349 2      END;
: 223 0350 2      TES;
: 224 0351 1      END;

```

!End of LIT

```

.TITLE LIT
.IDENT \V04-000\

.EXTRN GCA, IRA, NUMPRM
.EXTRN SCA, SCALIT, RNFELD
.EXTRN RNFINM, RNFLDE, ERMA
.EXTRN ERMB, OFT, STKFRM

```

.PSECT \$CODE\$,NOWRT,2

```

                                .ENTRY LIT, Save R2,R3,R4,R5,R6,R7
57 00000000G EF 9E 00002 MOVAB SCALIT, R7
56 00000000G EF 9E 00009 MOVAB OFT, R6
55 00000000G EF 9E 00010 MOVAB STKFRM, R5
54 00000000G EF 9E 00017 MOVAB NUMPRM+4, R4
53 00000000G EF 9E 0001E MOVAB SCA, R3
52 00000000G EF 9E 00025 MOVAB GCA+144, R2
50      04 AC D0 0002C MOVL HANDLER_CODE, R0
                                BNEQ 4$
01      62 E8 00032 BLBS GCA+144, 1$
                                RET
                                TSTL GCA+56
                                BGTR 3$
                                CLRL -(SP)
                                00000000G EF 00000000G 8F DD 0003D PUSHL #RNFLDE
                                02 FB 00043 CALLS #2, ERMB
                                C0 A2 D4 0004A CLRL GCA+80
                                7E 01 CE 0004D MNEGL #1, -(SP)
                                65 01 FB 00050 CALLS #1, STKFRM
                                7E D4 00053 CLRL -(SP)
                                B8 A2 DD 00055 PUSHL GCA+72
                                66 02 FB 00058 CALLS #2, OFT
                                50 D4 0005B CLRL I
                                F7 6340 6740 DO 0005D 2$: MOVL SCALIT[I], SCAL[I]
                                50 18 F3 00062 AOBLEQ #24, I, 2$
                                04 00066 RET

```

: 0219
: 0244
: 0247
: 0253
: 0256
: 0259
: 0260
: 0261
: 0262
: 0264
: 0265
: 0256

Module Level Declarations

		A8	A2	D7	00067	3\$:	DECL	GCA+56	:	0269			
				04	0006A		RET		:	0256			
	00000070	8F		50	D1	0006B	4\$:	CMPL	R0, #112	:	0273		
				5F	12	00072		BNEQ	8\$:			
		C0	A2	01	D0	00074		MOVL	#1, GCA+80	:	0275		
				62	D4	00078		CLRL	GCA+144	:	0276		
		B8	A2	A8	A2	D4	0007A	CLRL	GCA+56	:	0277		
				F4	B2	D0	0007D	MOVL	@GCA+132, GCA+72	:	0279		
					7E	D4	00082	CLRL	-(SP)	:	0282		
					01	FB	00084	CALLS	#1, STKFRM	:			
					7E	D4	00087	CLRL	-(SP)	:	0283		
					7E	9A	00089	MOVZBL	#124, -(SP)	:			
					02	FB	0008D	CALLS	#2, OFT	:			
					68	B3	D4	00090	CLRL	@SCA+104	:	0285	
					64	B3	D4	00093	CLRL	@SCA+100	:	0286	
		78	B3	96	8F	9A	00096	MOVZBL	#150, @SCA+120	:	0287		
	0084		D3		01	D0	0009B	MOVL	#1, @SCA+132	:	0288		
					50	D4	000A0	CLRL	I	:	0291		
					6340	D0	000A2	5\$:	MOVL	SCA[I], SCALIT[I]	:		
F7		6740			18	F3	000A7	AOBLEQ	#24, I, 5\$:			
					50	D4	000AB	CLRL	I	:	0293		
					6340	D4	000AD	6\$:	CLRL	SCA[I]	:		
F9					18	F3	000B0	AOBLEQ	#24, I, 6\$:			
					50					:			
					56					:			
					FC	A4	E9	000B4	BLBC	NUMPRM, 13\$:	0299	
					08	A4	D5	000B8	TSTL	NUMPRM+12	:	0303	
						51	13	000BB	BEQL	13\$:		
						64	D5	000BD	TSTL	NUMPRM+4	:	0307	
						0A	14	000BF	BGTR	7\$:		
						7E	D4	000C1	CLRL	-(SP)	:	0310	
					00000000G	8F	DD	000C3	PUSHL	#RNF INM	:		
						1F	11	000C9	BRB	10\$:		
						01	D0	000CB	7\$:	MOVL	#1, GCA+144	:	0314
		A8	A2		64	D0	000CE	MOVL	NUMPRM+4, GCA+56	:	0315		
						04	000D2	RET		:	0244		
						50	D1	000D3	8\$:	CMPL	R0, #60	:	0320
						36	12	000D6	BNEQ	13\$:		
						A2	E8	000D8	BLBS	GCA+80, 11\$:	0323	
						62	E9	000DC	BLBC	GCA+144, 9\$:	0327	
						62	D4	000DF	CLRL	GCA+144	:	0330	
						04	000E1	RET		:	0329		
						7E	D4	000E2	9\$:	CLRL	-(SP)	:	0334
					00000000G	8F	DD	000E4	PUSHL	#RNFELD	:		
						02	FB	000EA	10\$:	CALLS	#2, ERMA	:	
						04	000F1	RET		:	0325		
						A2	D4	000F2	11\$:	CLRL	GCA+80	:	0339
						7E	D4	000F5	CLRL	-(SP)	:	0341	
						A2	DD	000F7	PUSHL	GCA+72	:		
						02	FB	000FA	CALLS	#2, OFT	:		
						01	CE	000FD	MNEGL	#1, -(SP)	:	0343	
						01	FB	00100	CALLS	#1, STKFRM	:		
						50	D4	00103	CLRL	!	:	0345	
						6340	D0	00105	12\$:	MOVL	SCALIT[I], SCA[I]	:	0346
F7						18	F3	0010A	AOBLEQ	#24, I, 12\$:		
						04	0010E	13\$:	RET		:	0351	

; Routine Size: 271 bytes, Routine Base: \$CODE\$ + 0000

: 205 0352 1
: 226 0353 1 END !End of module
: 227 0354 0 ELUDOM

PSECT SUMMARY

Name Bytes Attributes
: \$CODE\$ 271 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]XPORT.L32;1	590	0	0	252	00:00.1
\$255\$DUA28:[RUNOFF.SRC]DSRLIB.L32;1	1248	24	1	86	00:00.3

COMMAND QUALIFIERS

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

: Size: 271 code + 0 data bytes
: Run Time: 00:06.4
: Elapsed Time: 00:26.3
: Lines/CPU Min: 3344
: Lexemes/CPU-Min: 15703
: Memory Used: 75 pages
: Compilation Complete

