



```

EEEEEEEEEE LL      SSSSSSSS EEEEEEEEEEE NN      NN
EEEEEEEEEE LL      SSSSSSSS EEEEEEEEEEE NN      NN
EE          LL      SS          EE          NN      NN
EE          LL      SS          EE          NN      NN
EE          LL      SS          EE          NN      NN
EE          LL      SS          EE          NN      NN
EEEEEEEEEE LL      SSSSSS   EEEEEEEEEEE NN  NN  NN
EEEEEEEEEE LL      SSSSSS   EEEEEEEEEEE NN  NN  NN
EE          LL          SS     EE          NN  NNNN
EE          LL          SS     EE          NN  NNNN
EE          LL          SS     EE          NN  NN
EE          LL          SS     EE          NN  NN
EEEEEEEEEE LLLLLLLLLL SSSSSSSS EEEEEEEEEEE NN  NN
EEEEEEEEEE LLLLLLLLLL SSSSSSSS EEEEEEEEEEE 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
LL          II      SS
LLLLLLLLLL IIIIII  SSSSSSSS
LLLLLLLLLL IIIIII  SSSSSSSS

```

.....

```

1 0001 0 MODULE ELSEN (
2 0002 0 IDENT = 'V04-000'
3 0003 0 %BLISS32[
4 P 0004 0 ADDRESSING_MODE(EXTERNAL=LONG_RELATIVE, NONEXTERNAL=LONG_RELATIVE)
5 0005 0 ]
6 0006 0 ) =
7 0007 1 BEGIN
8 0008 1
9 0009 1 *****
10 0010 1 *
11 0011 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
12 0012 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
13 0013 1 * ALL RIGHTS RESERVED. *
14 0014 1 *
15 0015 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
16 0016 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
17 0017 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
18 0018 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
19 0019 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
20 0020 1 * TRANSFERRED. *
21 0021 1 *
22 0022 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
23 0023 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
24 0024 1 * CORPORATION. *
25 0025 1 *
26 0026 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
27 0027 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
28 0028 1 *
29 0029 1 *
30 0030 1 *****
31 0031 1
32 0032 1 **
33 0033 1 FACILITY: DSR (Digital Standard RUNOFF) / DSRPLUS
34 0034 1
35 0035 1 ABSTRACT: Processes .END LIST and .END NOTE commands
36 0036 1
37 0037 1
38 0038 1 ENVIRONMENT: Transportable
39 0039 1
40 0040 1 AUTHOR: R.W.Friday CREATION DATE: January, 1980
41 0041 1

```

Revision History

:	43	0042	1	%SBTTL 'Revision History'
:	44	0043	1	
:	45	0044	1	MODIFIED BY:
:	46	0045	1	
:	47	0046	1	008 KAD00008 Keith Dawson 5-July-1983
:	48	0047	1	Made default spacing agter a note 1 instead of 2.
:	49	0048	1	
:	50	0049	1	007 KFA00007 Ken Alden 07-Mar-1983
:	51	0050	1	Global edit of all modules. Updated module names, idents,
:	52	0051	1	copyright dates. C.ranged require files to BLISS library.
:	53	0052	1	
:	54	0053	1	--

Module Level Declarations

```

56 0054 1 %SBTTL 'Module Level Declarations'
57 0055 1
58 0056 1
59 0057 1 : TABLE OF CONTENTS:
60 0058 1
61 0059 1
62 0060 1 : INCLUDE FILES:
63 0061 1
64 0062 1
65 0063 1 LIBRARY 'NXPORT:XPORT';           ! XPORT Library
66 0064 1 REQUIRE 'REQ:RNODEF';         ! RUNOFF variant definitions
67 0195 1
68 U 0196 1 %IF DSRPLUS %THEN
69 U 0197 1 LIBRARY 'REQ:DPLLIB';         ! DSRPLUS BLISS Library
70 0198 1 %ELSE
71 0199 1 LIBRARY 'REQ:DSRLIB';         ! DSR BLISS Library
72 0200 1 %FI
73 0201 1
74 0202 1
75 0203 1 : MACROS:
76 0204 1
77 0205 1
78 0206 1 : EQUATED SYMBOLS:
79 0207 1
80 0208 1
81 0209 1 LITERAL
82 0210 1     LFSPAC = 1;                 !Spacing after end of a list
83 0211 1     NFSPAC = 1;                 !Spacing after end of a note
84 0212 1
85 0213 1
86 0214 1 : OWN STORAGE:
87 0215 1
88 0216 1
89 0217 1 : EXTERNAL REFERENCES:
90 0218 1
91 0219 1
92 0220 1 EXTERNAL
93 0221 1     FNCT : FNCT DEFINITION,
94 0222 1     GCA : GCA DEFINITION,
95 0223 1     LSTCNT : REF COUNTED LIST,   !.LIST ELEMENT counters and controls.
96 0224 1     NUMPRM : NUMPRM DEFINE,
97 0225 1     PHAN : PHAN DEFINITION,
98 0226 1     SCA : SCA_DEFINITION;
99 0227 1
100 0228 1 !
101 0229 1
102 0230 1 EXTERNAL LITERAL                !Error messages
103 0231 1     RNFI:T,
104 0232 1     RNFINM;
105 0233 1
106 0234 1 EXTERNAL ROUTINE
107 0235 1     ERMA,
108 0236 1     GCPOS,
109 0237 1     GCSKIP,
110 0238 1     STKFRM;
111 0239 1

```

```

: 113 0240 1 GLOBAL ROUTINE ELSEN (HANDLER_CODE) : NOVALUE =
: 114 0241 1
: 115 0242 1
: 116 0243 1
: 117 0244 1
: 118 0245 1
: 119 0246 1
: 120 0247 1
: 121 0248 1
: 122 0249 1
: 123 0250 1
: 124 0251 1
: 125 0252 1
: 126 0253 1
: 127 0254 1
: 128 0255 1
: 129 0256 1
: 130 0257 1
: 131 0258 1
: 132 0259 1
: 133 0260 1
: 134 0261 1
: 135 0262 1
: 136 0263 1
: 137 0264 2
: 138 0265 2
: 139 0266 2
: 140 0267 2
: 141 0268 2
: 142 0269 2
: 143 0270 2
: 144 0271 2
: 145 0272 2
: 146 0273 2
: 147 0274 2
: 148 0275 2
: 149 0276 2
: 150 0277 2
: 151 0278 2
: 152 0279 2
: 153 0280 2
: 154 0281 2
: 155 0282 2
: 156 0283 2
: 157 0284 2
: 158 0285 2
: 159 0286 2
: 160 0287 2
: 161 0288 2
: 162 0289 2
: 163 0290 2
: 164 0291 2
: 165 0292 2
: 166 0293 2
: 167 0294 2
: 168 0295 2
: 169 0296 2

++
FUNCTIONAL DESCRIPTION:
    See ABSTRACT, above.

FORMAL PARAMETERS:
    HANDLER_CODE indicates which command is to be processed.

IMPLICIT INPUTS:
    NUMPRM contains a number, as processed by GETNUM.

IMPLICIT OUTPUTS:    None

ROUTINE VALUE:
COMPLETION CODES:    None

SIDE EFFECTS: None
--
BEGIN
LOCAL
    DEFAULT_SPACING,
    FINAL_SPACING;

!Pop the .LIST and .NOTE stack.
IF NOT STKFRM (.HANDLER_CODE)
THEN
    RETURN;                                !Stack underflow, etc

!Set up default .END xxxx spacing.
SELECTONE .HANDLER_CODE OF
SET

[H_END_NOTE] :
    BEGIN
    DEFAULT_SPACING = NFSPAC*.SCA_SPACING;
    END;

[H_END_LIST] :
    BEGIN
    DEFAULT_SPACING = LFSPAC*.SCA_SPACING;
    !Drop back one level in list counters.
    LSTCNT [CL_INDEX] = .LSTCNT [CL_INDEX] - 1;
    END;

TES;

!General coding for both .END LIST and .END NOTE.

!Generate final spacing after list
IF (NOT .NUM_RESULT)
    OR (.NUM_LENGTH EQL 0)                !If the number is bad,
                                           !or no number was specified,

```

```

170 0297 2 THEN
171 0298 !Use default .END xxxx spacing.
172 0299 FINAL_SPACING = .DEFAULT_SPACING
173 0300 ELSE
174 0301 !User has specified the .END xxxx skip.
175 0302 !Validate it and then do it.
176 0303 BEGIN
177 0304 FINAL_SPACING = .NUM_VALUE*.SCA_SPACING; !Convert to .SPACING lines.
178 0305
179 0306 IF .FINAL_SPACING LSS 0
180 0307 THEN
181 0308 !User said something like .END xxxx -5,
182 0309 !which means to position to 5 lines from the
183 0310 !bottom.
184 0311 IF .FNCT_COLLECTING !This is TRUE if user is inside a footnote.
185 0312 THEN
186 0313 BEGIN !.END xxxx with negative value is illegal in a footnote.
187 0314 ERMA (RNFIFT, FALSE);
188 0315 FINAL_SPACING = .DEFAULT_SPACING
189 0316 END
190 0317 ELSE !
191 0318
192 0319 IF ABS (.FINAL_SPACING) GEQ .PHAN_Lines
193 0320 THEN
194 0321 !User said something like .END xxxx -9999.
195 0322 BEGIN
196 0323 !Complain to the user
197 0324 ERMA (RNFIM, FALSE);
198 0325 !Use default spacing instead of erroneous spacing
199 0326 FINAL_SPACING = .DEFAULT_SPACING
200 0327 END
201 0328
202 0329 ELSE
203 0330 !It's a valid negative skip
204 0331 BEGIN
205 0332 GCPOS(.NUM_VALUE);
206 0333 RETURN
207 0334 END
208 0335 END;
209 0336
210 0337
211 0338 !If we arrive here, it's either .END xxxx with no skip specified,
212 0339 !or else the skip is a positive skip.
213 0340 !Bad skip specifications continue here too.
214 0341
215 0342
216 0343 !Check for .END xxxx 0
217 0344 IF .FINAL_SPACING NEQ 0
218 0345 THEN
219 0346 !SCA SPACING is subtracted off because when ENDCHR gets
220 0347 !called again (eventually) it will add that many blank
221 0348 !lines.
222 0349 GCSKIP ( MAX(1, .FINAL_SPACING - (.SCA_SPACING - 1)) );
223 0350
224 0351 END; !End of ELSEN

```

				.TITLE	ELSEN	
				.IDENT	\V04-000\	
				.EXTRN	FNCT, GCA, LSTCNT	
				.EXTRN	NUMPRM, PHAN, SCA	
				.EXTRN	RNFIFT, RNFINM, ERMA	
				.EXTRN	GCPOS, GCSKIP, STKFRM	
				.PSECT	\$CODE\$,NOWRT,2	
			003C 00000	.ENTRY	ELSEN, Save R2,R3,R4,R5	: 0240
	55	00000000G	EF 9E 00002	MOVAB	NUMPRM+4, R5	
	54	00000000G	EF 9E 00009	MOVAB	SCA+124, R4	
	52	04	AC D0 00010	MOVL	HANDLER_CODE, R2	: 0270
			52 DD 00014	PUSHL	R2	
	00000000G		EF 01 FB 00016	CALLS	#1, STKFRM	
	01		50 E8 0001D	BLBS	R0, 1\$	
			04 00020	RET		
	3D		52 D1 00021 1\$:	CMPL	R2, #61	: 0278
			06 12 00024	BNEQ	2\$	
	53	00	B4 D0 00026	MOVL	@SCA+124, DEFAULT_SPACING	: 0280
			13 11 0002A	BRB	3\$	: 0275
	3B		52 D1 0002C 2\$:	CMPL	R2, #59	: 0283
			0E 12 0002F	BNEQ	3\$	
	53	00	B4 D0 00031	MOVL	@SCA+124, DEFAULT_SPACING	: 0285
	50	00000000G	EF D0 00035	MOVL	LSTCNT, R0	: 0287
			04 A0 D7 0003C	DECL	4(R0)	
	3D	FC	A5 E9 0003F 3\$:	BLBC	NUMPRM, 7\$	: 0295
		08	A5 D5 00043	TSTL	NUMPRM+12	: 0296
			38 13 00046	BEQL	7\$	
	52	65	B4 C5 00048	MULL3	@SCA+124, NUMPRM+4, FINAL_SPACING	: 0304
			40 18 0004D	BGEQ	9\$	: 0306
	0A	00000000G	EF E9 0004F	BLBC	FNCT+20, 4\$	: 0311
			7E D4 00056	CLRL	-(SP)	: 0314
		00000000G	8F DD 00058	PUSHL	#RNFIFT	
			19 11 0005E	BRB	6\$	
	50		52 D0 00060 4\$:	MOVL	FINAL_SPACING, R0	: 0319
			03 18 00063	BGEQ	5\$	
	50		50 CE 00065	MNEGL	R0, R0	
	00000000G	FF	50 D1 00068 5\$:	CMPL	R0, @PHAN+4	
			14 19 0006F	BLSS	8\$	
			7E D4 00071	CLRL	-(SP)	: 0324
		00000000G	8F DD 00073	PUSHL	#RNFINM	
	00000000G	EF	02 FB 00079 6\$:	CALLS	#2, ERMA	
		52	53 D0 00080 7\$:	MOVL	DEFAULT_SPACING, FINAL_SPACING	: 0326
			0A 11 00083	BRB	9\$	
	00000000G	EF	65 DD 00085 8\$:	PUSHL	NUMPRM+4	: 0332
			01 FB 00087	CALLS	#1, GCPOS	
			04 0008E	RET		: 0331
			14 13 0008F 9\$:	BEQL	11\$	: 0344
	52	00	B4 C2 00091	SUBL2	@SCA+124, R2	: 0349
			52 D6 00095	INCL	R2	
			52 DD 00097	PUSHL	R2	
			03 14 00099	BCTR	10\$	
	00000000G	6E	01 D0 0009B	MOVL	#1, (SP)	
		EF	01 FB 0009E 10\$:	CALLS	#1, GCSKIP	
			04 000A5 11\$:	RET		: 0351



ELSEN  
V04-000

Module Level Declarations

K 14  
16-Sep-1984 00:22:32  
14-Sep-1984 13:06:06

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[RUNOFF.SRC]ELSEN.BLI;1 Page 7 (4)

: Routine Size: 166 bytes, Routine Base: \$CODE\$ + 0000

: 225 0352 1  
: 226 0353 1 END !End of module  
: 227 0354 0 ELUDOM

PSECT SUMMARY

: Name Bytes Attributes  
: \$CODE\$ 166 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	19	1	86	00:00.3

COMMAND QUALIFIERS

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

: Size: 166 code + 0 data bytes  
: Run Time: 00:04.7  
: Elapsed Time: 00:15.4  
: Lines/CPU Min: 4548  
: Lexemes/CPU-Min: 11678  
: Memory Used: 55 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 view of a system, likely related to the VAX/VMS operating system. The content within the windows is mostly illegible due to the small size and low resolution, but some key elements are visible:

- Many windows show a header with the text "LSPV00000000" and "LSPV00000000" below it, possibly indicating a specific process or user.
- Some windows display vertical bars or histograms, suggesting graphical data representation.
- Other windows show lists of text, possibly command-line output or system logs.
- Specific labels are visible in some windows, such as "ELSEN LIS" in the second row, eighth column; "ENDCMT LIS" in the fifth row, tenth column; "ENDCHR LIS" in the sixth row, tenth column; "DSRMSG LIS" in the seventh row, eighth column; and "EMSG LIS" in the seventh row, ninth column.