


```

NN      NN      000000  TTTTTTTTTT  EEEEEEEEEE
NN      NN      000000  TTTTTTTTTT  EEEEEEEEEE
NN      NN      00      00      TT      FF
NN      NN      00      00      TT      FF
NNNN    NN      00      00      TT      FF
NNNN    NN      00      00      TT      FF
NN      NN      00      00      TT      EEEEEEEE
NN      NN      00      00      TT      EEEEEEEE
NN      NN      00      00      TT      FF
NN      NN      00      00      TT      FF
NN      NN      00      00      TT      FF
NN      NN      00      00      TT      FF
NN      NN      000000  TT      EEEEEEEEEE
NN      NN      000000  TT      EEEEEEEEEE

```

```

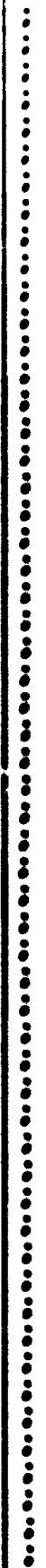
....
....
....
....

```

```

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
LLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLL  IIIIII  SSSSSSSS

```



```

1 0001 0 MODULE NOTE (
2 0002 0 IDENT = 'V04-000'
3 P 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 *
12 0012 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
13 0013 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
14 0014 1 * ALL RIGHTS RESERVED.
15 0015 1 *
16 0016 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
17 0017 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
18 0018 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
19 0019 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
20 0020 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
21 0021 1 * TRANSFERRED.
22 0022 1 *
23 0023 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
24 0024 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
25 0025 1 * CORPORATION.
26 0026 1 *
27 0027 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
28 0028 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
29 0029 1 *
30 0030 1 *
31 0031 1 *****
32 0032 1
33 0033 1
34 0034 1 **
35 0035 1 FACILITY: DSR (Digital Standard RUNOFF) / DSRPLUS
36 0036 1
37 0037 1 ABSTRACT: Processes the .NOTE command
38 0038 1
39 0039 1
40 0040 1 ENVIRONMENT: Transportable
41 0041 1
42 0042 1 AUTHOR: R.W.Friday CREATION DATE: June, 1978
43 0043 1

```

Revision History

J 6
16-Sep-1984 01:18:11
14-Sep-1984 13:07:27

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

:	45	0044	1	%SBTTL 'Revision History'
:	46	0045	1	
:	47	0046	1	MODIFIED BY:
:	48	0047	1	
:	49	0048	1	006 KAD00006 Keith Dawson 9-Jun-1983
:	50	0049	1	Fix .NOTE margins to be more sensible. This is motivated by
:	51	0050	1	the fact that notes come out looking centered on .PS, instead
:	52	0051	1	of on .LM-.RM as desired. Also, changed the number of
:	53	0052	1	lines before a note to 1 from 2.
:	54	0053	1	
:	55	0054	1	005 KAD00005 Keith Dawson 07-Mar-1983
:	56	0055	1	Global edit of all modules. Updated module names, idents,
:	57	0056	1	copyright dates. Changed require files to BLISS library.
:	58	0057	1	
:	59	0058	1	--

Module Level Declarations

```

: 61      0059 1 %SBTTL 'Module Level Declarations'
: 62      0060 1
: 63      0061 1  TABLE OF CONTENTS:
: 64      0062 1
: 65      0063 1
: 66      0064 1  INCLUDE FILES:
: 67      0065 1
: 68      0066 1
: 69      0067 1  LIBRARY 'NXPOR:XPOR';           ! XPORT Library
: 70      0068 1  REQUIRE 'REQ:RNODEF';       ! RUNOFF variant definitions
: 71      0199 1
: 72      U 0200 1  %IF DSRPLUS %THEN
: 73      U 0201 1  LIBRARY 'REQ:DPLLIB';     ! DSRPLUS BLISS Library
: 74      0202 1  %ELSE
: 75      0203 1  LIBRARY 'REQ:DSRLIB';     ! DSR BLISS Library
: 76      0204 1  %FI
: 77      0205 1
: 78      0206 1
: 79      0207 1  MACROS:
: 80      0208 1
: 81      0209 1
: 82      0210 1  EQUATED SYMBOLS:
: 83      0211 1
: 84      0212 1
: 85      0213 1  LITERAL
: 86      0214 1  skip_before_note = 1,      ! Spacing before 'NOTE'
: 87      0215 1  ... changed from 2 by KAD, 6-10-1983.
: 88      0216 1  skip_after_note = 1;      ! Spacing after 'NOTE'
: 89      0217 1
: 90      0218 1  OWN STORAGE:
: 91      0219 1
: 92      0220 1
: 93      0221 1  EXTERNAL REFERENCES:
: 94      0222 1
: 95      0223 1
: 96      0224 1  EXTERNAL
: 97      0225 1  gca : gca_definition,
: 98      0226 1  pdt : ref_pdt_definition,
: 99      0227 1  sca : sca_definition;
100     0228 1
101     0229 1  !
102     0230 1
103     0231 1  EXTERNAL ROUTINE
104     0232 1  centxt,
105     0233 1  gcskip,
106     0234 1  getlin,
107     0235 1  gtpc,
108     0236 1  stkfrm;
109     0237 1

```

```
: 111 0238 1 GLOBAL ROUTINE NOTE (HANDLER_CODE) : NOVALUE =
: 112 0239 1
: 113 0240 1 !++
: 114 0241 1 FUNCTIONAL DESCRIPTION:
: 115 0242 1
: 116 0243 1     See the ABSTRACT, above.
: 117 0244 1
: 118 0245 1 FORMAL PARAMETERS:
: 119 0246 1
: 120 0247 1     HANDLER_CODE indicates which command is to be processed.
: 121 0248 1
: 122 0249 1 IMPLICIT INPUTS:     None
: 123 0250 1
: 124 0251 1 IMPLICIT OUTPUTS:     None
: 125 0252 1
: 126 0253 1 ROUTINE VALUE:
: 127 0254 1 COMPLETION CODES:     None
: 128 0255 1
: 129 0256 1 SIDE EFFECTS: None
: 130 0257 1
: 131 0258 1 --
: 132 0259 2 BEGIN
: 133 0260 2 LOCAL
: 134 0261 2     margin_adjust,           !How much to adjust the margins.
: 135 0262 2     skip;
: 136 0263 2
: 137 0264 2 IF NOT stkfrm (.handler_code)
: 138 0265 2 THEN
: 139 0266 2     RETURN;                 !Quit if stack overflow
: 140 0267 2
: 141 0268 2     skip = skip_before_note * .sca_spacing;
: 142 0269 2     gtpc (.skip + (skip_after_note + .pdt_tp + 1)*.sca_spacing);
: 143 0270 2     gcskip (max (0, .skip - (.sca_spacing - 1)));
: 144 0271 2     !Set up formatting definitions
: 145 0272 2     sca_fill = true;
: 146 0273 2     sca_justify = .gca_autojust or .sca_justify;           !Turn on justification unless user has said .NAJ.
: 147 0274 2     sca_crock = .sca_justify;
: 148 0275 2
: 149 0276 2 ! Margin adjustment.
: 150 0277 2
: 151 0278 2 IF (.sca_rm - .sca_lm) GTR 60
: 152 0279 2 THEN
: 153 0280 2     margin_adjust = .gca_note_prim ! (8)
: 154 0281 2 ELSE
: 155 0282 2     margin_adjust = .gca_note_alt; ! (4)
: 156 0283 2
: 157 0284 2 !Defensive margin computation in case margins are very restrictive
: 158 0285 2
: 159 0286 2 IF (.sca_lm + .margin_adjust) LSS .sca_rm
: 160 0287 2 THEN
: 161 0288 2     sca_lm = .sca_lm + .margin_adjust;
: 162 0289 2
: 163 0290 2 IF (.sca_rm - .margin_adjust) GTR .sca_lm
: 164 0291 2 THEN
: 165 0292 2     sca_rm = .sca_rm - .margin_adjust;
: 166 0293 2
: 167 0294 2 ! End of margin-adjustment computations.
```

```

: 168 0295 2
: 169 0296 2 !Determine if text was given on the .NOTE command.
: 170 0297 2
: 171 0298 2 IF NOT centxt (ch$ptr (uplit ('NOTE')), 4)
: 172 0299 2 THEN
: 173 0300 2 !Text after ".NOTE". Fetch and output text centered.
: 174 0301 2 getlin (true, false, 0, true);
: 175 0302 2
: 176 0303 2 !Prepare for the text that will probably come.
: 177 0304 2 gcskip (.sca_spacing);
: 178 0305 2
: 179 0306 2 END;

```

!End of NOTE

```

.TITLE NOTE
.IDENT \V04-000\

.PSECT $PLITS, NOWRT, NOEXE, 2

.ASCII \NOTE\

.EXTRN GCA, PDT, SCA, CENTXT
.EXTRN GCSKIP, GETLIN, GTPC
.EXTRN STKFRM

.PSECT $CODE$, NOWRT, 2

```

```

003C 00000
55 00000000G EF 9E 00002
54 00000000G EF 9E 00009
04 AC DD 00010
00000000G EF 01 FB 00013
01 50 E8 0001A
04 0001D
52 00 B4 D0 0001E 1$:
50 00000000G EF D0 00022
50 08 A0 02 C1 00029
50 00 B4 C4 0002E
00000000G EF 6042 9F 00032
52 00 01 FB 00035
00 B4 C2 0003C
52 D6 00040
52 DD 00042
02 18 00044
6E D4 00046
2$:
65 01 FB 00048
EC B4 01 D0 0004B
E8 B4 00000000G FF C8 0004F
F4 B4 EB B4 D0 00057
51 FC B4 D0 0C05C
52 FB A4 D0 00060
50 62 3C C1 00064
50 51 D1 00068
09 15 0006B
50 00000000G EF D0 0006D
07 11 00074
50 00000000G EF D0 00076 3$:

```

```

.ENTRY NOTE, Save R2,R3,R4,R5
MOVAB GCSKIP, R5
MOVAB SCA+124, R4
PUSHL HANDLER CODE
CALLS #1, STKFRM
BLBS RC, 1$
RET
MOVL @SCA+124, SKIP
MOVL PDT, R0
ADDL3 #2, 8(R0), R0
MULL2 @SCA+124, R0
PUSHAB (R0)[SKIP]
CALLS #1, GTPC
SUBL2 @SCA+124, R2
INCL R2
PUSHL R2
BGEQ 2$
CLRL (SP)
CALLS #1, GCSKIP
MOVL #1, @SCA+104
BISL2 @GCA+16, @SCA+100
MOVL @SCA+100, @SCA+112
MOVL @SCA+120, R1
MOVL SCA+116, R2
ADDL3 #60, (R2), R0
CML R1, R0
BLEQ 3$
MOVL GCA+148, MARGIN_ADJUST
BRB 4$
MOVL GCA+152, MARGIN_ADJUST

```

0238
0264
0268
0269
0270
0272
0273
0274
0278
0280
0282

NOTE
V04-000

Module Level Declarations

**F

```

53          62          50 C1 0007D 4$:  ADDL3  MARGIN_ADJUST, (R2), R3          ; 0286
          51          53 D1 00081      CMPL   R3, R1          ;
          62          03 18 00084      BGEQ   5$          ;
          51          50 C0 00086      ADDL2  MARGIN_ADJUST, (R2)          ; 0288
          62          50 C2 00089 5$:  SUBL2  MARGIN_ADJUST, R1          ; 0290
          51          51 D1 0008C      CMPL   R1, (R2)          ;
          FC  B4          04 15 0008F      BLEQ   6$          ;
          00000000G EF 00000000' 50 C2 00091      SUBL2  MARGIN_ADJUST, @SCA+120          ; 0292
          OD          04 DD 00095 6$:  PUSHL  #4          ; 0298
          00000000G EF          EF 9F 00097      PUSHAB P,AAA          ;
          0D          02 FB 0009D      CALLS  #2, CENTXT          ;
          00000000G EF          50 EB 000A4      BLBS   R0, 7$          ;
          00          01 DD 000A7      PUSHL  #1          ; 0301
          00          7E 7C 000A9      CLRQ   -(SP)          ;
          00          01 DD 000AB      PUSHL  #1          ;
          65          00          04 FB 000AD      CALLS  #4, GETLIN          ;
          00          84 DD 000B4 7$:  PUSHL  @SCA+124          ; 0304
          00          01 FB 000B7      CALLS  #1, GCSKIP          ;
          00          04 000BA      RET          ; 0306

```

; Routine Size: 187 bytes, Routine Base: \$CODE\$ + 0000

```

: 180          0307 1
: 181          0308 1 END          !End of module
: 182          0309 0 ELUDOM

```

PSECT SUMMARY

Name	Bytes	Attributes
\$PLITS	4	NOVEC,NOWRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
\$CODE\$	187	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

File	Symbols			Pages Mapped	Processing Time
	Total	Loaded	Percent		
\$255\$DUA28:[SYSLIB]XPORT.L32;1	590	0	0	252	00:00.2
_\$255\$DUA28:[RUNOFF.SRC]DSRLIB.L32;1	1248	18	1	86	00:00.3

COMMAND QUALIFIERS

NOTE
V04-000

Module Level Declarations

B 7
16-Sep-1984 01:18:11
14-Sep-1984 13:07:27

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

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

: Size: 187 code + 4 data bytes
: Run Time: 00:05.0
: Elapsed Time: 00:12.7
: Lines/CPU Min: 3685
: Lexemes/CPU-Min: 14922
: Memory Used: 56 pages
: Compilation Complete

NEWSPAG LIS	NODOPX LIS	OFT LIS	OUTXT LIS
NDXURS LIS	NOTE LIS	OUTLN LIS	PACK LIS
NM LIS	OUTXHR LIS	NDXXTN LIS	OUTCHA LIS
OUTHDR LIS			