



```

UU      UU  NN      NN  P P P P P P P P  UU      UU  S S S S S S S S
UU      UU  NN      NN  P P P P P P P P  UU      UU  S S S S S S S S
UU      UU  NN      NN  PP          PP  UU      UU  SS
UU      UU  NN      NN  PP          PP  UU      UU  SS
UU      UU  N N N N  NN  PP          PP  UU      UU  SS
UU      UU  N N N N  NN  PP          PP  UU      UU  SS
UU      UU  NN  NN  NN  P P P P P P P P  UU      UU  S S S S S S
UU      UU  NN  NN  NN  P P P P P P P P  UU      UU  S S S S S S
UU      UU  NN      N N N N  PP          SS
UU      UU  NN      N N N N  PP          SS
UU      UU  NN      NN      PP          SS
UU      UU  NN      NN      PP          SS
UU      UU  NN      NN      PP          SS
UUUUUUUUUU  NN      NN  PP          S S S S S S S S
UUUUUUUUUU  NN      NN  PP          S S S S S S S S

```

```

LL      I I I I I I  S S S S S S S S
LL      I I I I I I  S S S S S S S S
LL      I I          SS
LL      I I          SS
LL      I I          SS
LL      I I          SS
LL      I I          S S S S S S
LL      I I          S S S S S S
LL      I I          SS
LL      I I          SS
LL      I I          SS
LL      I I          SS
LLLLLLLLLLLL  I I I I I I  S S S S S S S S
LLLLLLLLLLLL  I I I I I I  S S S S S S S S

```



```

1 0001 0 MODULE UNPUS (
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 FACILITY: DSR (Digital Standard RUNOFF) / DSRPLUS
35 0035 1
36 0036 1 ABSTRACT: Takes the contents of MRA as described by TSF and translates
37 0037 1 it into an equivalent .INDEX or .SEND TOC command.
38 0038 1
39 0039 1 ENVIRONMENT: Transportable
40 0040 1
41 0041 1 AUTHOR: R.W.Friday CREATION DATE: February, 1979
42 0042 1

```

Revision History

```

: 44      0043 1 %SBTTL 'Revision History'
: 45      0044 1
: 46      0045 1   MODIFIED BY:
: 47      0046 1
: 48      0047 1       006   KFA00006   Ken Alden   5-Apr-1983
: 49      0048 1       Added support for SEND CONTENTS.
: 50      0049 1
: 51      0050 1       005   KAD00005   Keith Dawson 22-Mar-1983
: 52      0051 1       Added support for non-STREAM output (for LN01, VT100) --
: 53      0052 1       call to macro OP_DEV WRITE_OUTPUT_line instead of to
: 54      0053 1       clh (clh_write_output).
: 55      0054 1
: 56      0055 1       004   KAD00004   Keith Dawson 07-Mar-1983
: 57      0056 1       Global edit of all modules. Updated module names, idents,
: 58      0057 1       copyright dates. Changed require files to BLISS library.
: 59      0058 1
: 60      0059 1  --

```

Module Level Declarations

```

: 62      0060 1 %SBTTL 'Module Level Declarations'
: 63      0061 1
: 64      0062 1 : TABLE OF CONTENTS:
: 65      0063 1
: 66      0064 1
: 67      0065 1
: 68      0066 1 : INCLUDE FILES:
: 69      0067 1
: 70      0068 1
: 71      0069 1 LIBRARY 'NXPORT:XPORT';           ! XPORT Library
: 72      0070 1 REQUIRE 'REQ:RNODEF';           ! RUNOFF variant definitions
: 73      0201 1
: 74      U 0202 1 %IF DSRPLUS %THEN
: 75      U 0203 1 LIBRARY 'REQ:DPLLIB';           ! DSRPLUS BLISS Library
: 76      0204 1 %ELSE
: 77      0205 1 LIBRARY 'REQ:DSRLIB';           ! DSR BLISS Library
: 78      0206 1 %FI
: 79      0207 1
: 80      0208 1
: 81      0209 1 : MACROS:
: 82      0210 1
: 83      0211 1
: 84      0212 1
: 85      0213 1 : EQUATED SYMBOLS:
: 86      0214 1
: 87      0215 1 EXTERNAL LITERAL
: 88      0216 1     RINTES : UNSIGNED (8);
: 89      0217 1
: 90      0218 1
: 91      0219 1 : OWN STORAGE:
: 92      0220 1
: 93      0221 1
: 94      0222 1
: 95      0223 1 : EXTERNAL REFERENCES:
: 96      0224 1
: 97      0225 1 EXTERNAL
: 98      0226 1     FLGT : FLAG_TABLE,
: 99      0227 1     FRA : FIXED_STRING,
100     0228 1     GCA : GCA_DEFINITION,
101     0229 1     MRA : REF_FIXED_STRING,
102     0230 1     TSF : TSF_DEFINITION;
103     0231 1
104     0232 1 EXTERNAL ROUTINE
105     0233 1     CLH;
```

```
107 0234 1 GLOBAL ROUTINE UNPUS (TOC) : NOVALUE = .
108 0235 1
109 0236 1 |++
110 0237 1 | FUNCTIONAL DESCRIPTION:
111 0238 1 |
112 0239 1 |     UNPUS untranslates the results of a .INDEX command or
113 0240 1 |     INDEX flag, and puts the result into the output file.
114 0241 1 |
115 0242 1 | FORMAL PARAMETERS:
116 0243 1 |
117 0244 1 |     If TOC is true, what is to be regurgitated is a .SEND TOC
118 0245 1 |     or a .SEND CON(tent;) command, depending on the TSF_MAJOR.
119 0246 1 |     Otherwise, its an ndexing command.
120 0247 1 |
121 0248 1 | IMPLICIT INPUTS:      None
122 0249 1 |
123 0250 1 | IMPLICIT OUTPUTS:    None
124 0251 1 |
125 0252 1 | ROUTINE VALUE:
126 0253 1 | COMPLETION CODES:    None
127 0254 1 |
128 0255 1 | SIDE EFFECTS: None
129 0256 1 |
130 0257 1 | --
131 0258 1 |
132 0259 2     BEGIN
133 0260 2     LOCAL
134 0261 2         mark,
135 0262 2         ptr;
136 0263 2
137 0264 2     fs_init (fra);                !Initialize FRA.
138 0265 2
139 0266 2     ptr = .fs_start(mra);
140 0267 2
141 0268 2     !Generate either .SEND TOC, or .INDEX / .ENTRY
142 0269 2     !or .XPLUS / .YPLUS.
143 0270 2     IF .toc
144 0271 2     THEN
145 0272 3         BEGIN
146 0273 3             IF .tsf_major EQL maj_send
147 0274 3             THEN
148 0275 3                 mark = CH$PTR ( UPLIT('.SEND TOC '))
149 0276 3 %IF DSRPLUS %THEN
150 0277 3             ELSE
151 0278 3                 mark = CH$PTR ( UPLIT('.SEND CONT'))
152 0279 3 %FI
153 0280 3             END
154 0281 2     ELSE
155 0282 2         IF .tsf_first_xtn NEQ 0
156 0283 2         THEN
157 0284 2 %IF DSRPLUS %THEN
158 0285 2             IF .tsf_xyplus
159 0286 2             THEN
160 0287 2                 mark = CH$PTR ( UPLIT('.XPLUS  '))
161 0288 2             ELSE
162 0289 2 %FI
163 0290 2                 mark = CH$PTR ( UPLIT('.INDEX  '))
```

```
164      0291      2      !
165      0292      2      ELSE
166      0293      2      %IF DSRPLUS %THEN
167      0294      2      IF .tsf_xyplus
168      0295      2      THEN
169      0296      2      mark = CH$PTR ( UPLIT('YPLUS  '))
170      0297      2      ELSE
171      0298      2      %FI
172      0299      2      mark = CH$PTR ( UPLIT('ENTRY  '));
173      0300      2
174      0301      2      INCR i FROM 1 TO 10 DO
175      0302      2      FS_WCHAR (fra, CH$RCHAR_A (mark) );
176      0303      2      FS_WCHAR (fra, %C' ');
177      0304      2
178      0305      2      !Scan the text, untranslating escape codes, etc. back to RUNOFF flags.
179      0306      2      INCR i FROM 1 TO .tsf_int_hl DO
180      0307      2      BEGIN
181      0308      2      LOCAL
182      0309      2      khar;
183      0310      2
184      0311      2      khar = CH$RCHAR_A (ptr);
185      0312      2
186      0313      2      IF .khar EQL rintes
187      0314      2      THEN
188      0315      2      !Untranslate special function
189      0316      2      BEGIN
190      0317      2      LOCAL
191      0318      2      function_code,
192      0319      2      operand;
193      0320      2
194      0321      2      function_code = CH$RCHAR_A (ptr);
195      0322      2      operand = CH$RCHAR_A (ptr);
196      0323      2      i = .i + 2;
197      0324      2
198      0325      2      SELECTONE .function_code OF
199      0326      2      SET
200      0327      2      [%C'B'] :
201      0328      2      !Bolded character.
202      0329      2      fs_wchar (fra, .flgt [bld_flag, flag_character]);
203      0330      2
204      0331      2      [%C'U'] :
205      0332      2      !Underlined character.
206      0333      2      fs_wchar (fra, .flgt [und_flag, flag_character]);
207      0334      2
208      0335      2      [%C'O'] :
209      0336      2      BEGIN
210      0337      2      !An overstruck character.
211      0338      2      !NOTE: Order is the reverse of what user specified.
212      0339      2      fs_wchar (fra, .operand);
213      0340      2      fs_wchar (fra, .flgt [ovr_flag, flag_character]);
214      0341      2      END;
215      0342      2
216      0343      2      [%C'J'] :
217      0344      2      !A word mark. For indexing commands, this
218      0345      2      !starts a new sub-indexing level.
219      0346      2      fs_wchar (fra, .flgt [sbx_flag, flag_character]);
220      0347      2
```

```

: 221 U 0348 4 %IF DSRPLUS %THEN
: 222 U 0349 4 [%C'P'] :
: 223 U 0350 4 !Non-permuted word.
: 224 U 0351 4 fs_wchar (fra, .flgt [npx_flag, flag_character]);
: 225 U 0352 4 %FI
: 226 U 0353 4 [OTHERWISE] :
: 227 U 0354 4 !Unknown/unsupported special function
: 228 U 0355 4 fs_wchar (fra, %c'?');
: 229 U 0356 4
: 230 U 0357 4 TES;
: 231 U 0358 4 END
: 232 U 0359 3 ELSE
: 233 U 0360 3 !Some normal character
: 234 U 0361 4 BEGIN
: 235 U 0362 4 !Normal characters go out as themselves. Control characters
: 236 U 0363 4 !are translated to something else.
: 237 U 0364 5 IF (.khar GEQ %C' ')
: 238 U 0365 5 AND (.khar LEQ %O'176')
: 239 U 0366 4 THEN
: 240 U 0367 4 !Output a normal character
: 241 U 0368 5 fs_wchar (fra, .khar)
: 242 U 0369 4 ELSE
: 243 U 0370 4 !Translate and output a control character.
: 244 U 0371 4 !NOTE: DEL (Octal 177) and NUL (Octal 0) have the same result.
: 245 U 0372 5 BEGIN
: 246 U 0373 5 fs_wchar (fra, %c'^');
: 247 U 0374 5 fs_wchar (fra, %c'@'+.khar);
: 248 U 0375 4 END;
: 249 U 0376 3 END;
: 250 U 0377 2 END;
: 251 U 0378 2 op_dev_write_output_line; !Write result to document.
: 252 U 0379 2
: 253 U 0380 2
: 254 U 0381 1 END; !End of UNPUS

```

```

.TITLE UNPUS
.IDENT \V04-000\

.PSECT $SPLITS, NOWRT, NCEXE, 2

00 00 20 43 4F 54 20 44 4E 45 53 2E 0000 P.AAA: .ASCII \.SEND TOC \<0><0>
00 00 20 20 20 20 58 45 44 4E 49 2E 0000C P.AAB: .ASCII \.INDEX \<0><0>
00 00 20 20 20 20 59 52 54 4E 45 2E 00018 P.AAC: .ASCII \.ENTRY \<0><0>

.EXTRN RINTES, FLGT, FRA
.EXTRN GCA, MRA, TSF, CLH

.PSECT $CODES, NOWRT, 2

FC 57 00000000' 00FC 00000 .ENTRY UNPUS, Save R2,R3,R4,R5,R6,R7 : 0234
56 00000000G EF 9E 00002 MOVAB P.AAA, R7
55 00000000G EF 9E 00009 MOVAB FLGT+104, R6
08 A5 D4 00017 MOVAB FRA+4, R5
FC A5 OC A5 9E 0001A CLRL FRA+12 : 0264
65 FC A5 D0 0001F MOVAB FRA+16, FRA
MOVL FRA, FRA+4

```



	54	00000000G	FF	D0	00023	MOVL	@MRA, PTR	0266	
	50	00000000G	EF	D0	0002A	MOVL	TSF, R0	0273	
	08	04	AC	E9	00031	BLBC	TOC, 1\$	0270	
	03	38	A0	D1	00035	CMPL	56(R0), #3	0273	
			14	12	00039	BNEQ	3\$		
	51		67	9E	0003B	MOVAB	P.AAA, MARK	0275	
			0F	11	0003E	BRB	3\$	0270	
			38	A0	D5	00040	TSTL	56(R0)	0282
			06	13	00043	BEQL	2\$		
	51	0C	A7	9E	0C045	MOVAB	P.AAB, MARK	0290	
			04	11	00049	BRB	3\$		
	51	18	A7	9E	0004B	MOVAB	P.AAC, MARK	0299	
	50		01	D0	0C04F	MOVL	#1, I	0301	
	00	B5	81	90	00052	MOVB	(MARK)+, @FRA+4	0302	
			65	D6	00056	INCL	FRA+4		
			08	A5	D6	0005B	INCL	FRA+12	
F3	50		0A	F3	0005B	AOBLEQ	#10, I, 4\$	0301	
	00	B5	20	90	0005F	MOVB	#32, @FRA+4	0303	
			65	D6	0C063	INCL	FRA+4		
			08	A5	D6	00065	INCL	FRA+12	
			52	D4	00068	CLRL	I	0306	
			008C	31	0006A	BRW	13\$		
	51		84	9A	0006D	MOVZBL	(PTR)+, KHAR	0311	
00000000G	8F		51	D1	00070	CMPL	KHAR, #RINTES	0313	
			57	12	00077	BNEQ	10\$		
	50		84	9A	00079	MOVZBL	(PTR)+, FUNCTION_CODE	0321	
	53		84	9A	0C07C	MOVZBL	(PTR)+, OPERAND	0322	
	52		02	C0	0007F	ADDL2	#2, I	0323	
00000042	8F		50	D1	00082	CMPL	FUNCTION_CODE, #66	0327	
			06	12	00089	BNEQ	6\$		
	00	B5	66	90	0008B	MOVB	FLGT+104, @FRA+4	0329	
			63	11	0008F	BRB	12\$		
00000055	8F		50	D1	00091	CMPL	FUNCTION_CODE, #85	0331	
			07	12	00098	BNEQ	7\$		
	00	B5	FC	A6	90	0009A	MOVB	FLGT+100, @FRA+4	0333
			53	11	0009F	BRB	12\$		
0000004F	8F		50	D1	000A1	CMPL	FUNCTION_CODE, #79	0335	
			10	12	0C0A8	BNEQ	8\$		
	00	B5	53	90	000AA	MOVB	OPERAND, @FRA+4	0339	
			65	D6	000AE	INCL	FRA+4		
			08	A5	D6	000B0	INCL	FRA+12	
	00	B5	10	A6	90	000B3	MOVB	FLGT+120, @FRA+4	0340
			3A	11	000B3	BRB	12\$		
0000004A	8F		50	D1	000BA	CMPL	FUNCTION_CODE, #74	0343	
			07	12	000C1	BNEQ	9\$		
	00	B5	14	A6	90	000C3	MOVB	FLGT+124, @FRA+4	0346
			2A	11	000C8	BRB	12\$		
	00	B5	3F	90	000CA	MOVB	#63, @FRA+4	0355	
			24	11	000CE	BRB	12\$		
	20		51	D1	000D0	CMPL	KHAR, #32	0364	
			0F	19	000D3	BLSS	11\$		
0000007E	8F		51	D1	000D5	CMPL	KHAR, #126	0365	
			06	14	000DC	BGTR	11\$		
	00	B5	51	90	000DE	MOVB	KHAR, @FRA+4	0368	
			10	11	000E2	BRB	12\$		
	00	B5	5E	8F	90	000E4	MOVB	#94, @FRA+4	0373
			65	D6	000E9	INCL	FRA+4		

```

      00 B5          51      08  A5 D6 000EB      INCL  FRA+12
      40      8F 81 000EE      ADDB3 #64, KHAR, @FRA+4
      65 D6 000F4 12$:      INCL  FRA+4
      08  A5 D6 000F6      INCL  FRA+12
FF6A 01 00000000G 52      FF F1 000F9 13$:      ACBL  @TSF, #1, 1, 5$
  02 00000000G EF      04 04 ED 00103      CMPZV #4, #4, GCA+208, #2
      04 04 14 0010C      BGTR  14$
      06 DD 0010E      PUSHL #6
      02 11 00110      BRB   15$
      0B DD 00112 14$:      PUSHL #11
      01 04 0011B 15$:      CALLS #1, CLH
      04 0011B      RET

```

```

: 0374
: 0368
: 0306
: 0377
:
: 0381

```

: Routine Size: 284 bytes, Routine Base: \$CODE\$ + 0000

```

: 255      0382 1
: 256      0383 1 END
: 257      0384 0 ELUDOM

```

!End of module

PSECT SUMMARY

Name	Bytes	Attributes
\$SPLITS	36	NOVEC,NOWRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
\$CODE\$	284	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NGPIC,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	28	2	86	00:00.3

COMMAND QUALIFIERS

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

```

: Size:      284 code + 36 data bytes
: Run Time:  00:08.1
: Elapsed Time: 00:26.3
: Lines/CPU Min: 2844

```

UNPUS  
V04-000

Module Level Declarations

N 3  
16-Sep-1984 01:56:20

VAX-11 Bliss-32 V4.0-742

Page 9

XTA  
VJ4

: Lexemes/CPU-Min: 21525  
: Memory Used: 105 pages  
: Compilation Complete

.....

