

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

```
0001 0 MODULE list ( IDENT = 'V04-000'  
P 0002 0 %BLISS32[, 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: Processes .LIST and .LIST ELEMENT commands.  
0034 1  
0035 1 ENVIRONMENT: Transportable  
0036 1  
0037 1 AUTHOR: R.W.Friday CREATION DATE: June, 1978  
0038 1  
0039 1
```

B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z
[
\
]
^
_
`
a
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p
q
r
s
t
u
v
w
x
y
z
{|

Revision History

:	41	0040	1	%SBTTL 'Revision History'
:	42	0041	1	
:	43	0042	1	MODIFIED BY:
:	44	0043	1	
:	45	0044	1	004 RER00004 Ron Randall 07-Mar-1983
:	46	0045	1	Global edit of all modules. Updated module names, idents,
:	47	0046	1	copyright dates. Changed require files to BLISS library.
:	48	0047	1	
:	49	0048	1	--
:	50	0049	1	

Module Level Declarations

```

: 52      0050 1 %SBTTL 'Module Level Declarations'
: 53      0051 1
: 54      0052 1
: 55      0053 1 : TABLE OF CONTENTS:
: 56      0054 1
: 57      0055 1 : INCLUDE FILES:
: 58      0056 1
: 59      0057 1 LIBRARY 'NXPORT:XPORT';           ! XPORT Library
: 60      0058 1 REQUIRE 'REQ:RNODEF';           ! RUNOFF variant definitions
: 61      0189 1
: 62      U 0190 1 %IF DSRPLUS %THEN
: 63      U 0191 1 LIBRARY 'REQ:DPLLIB';           ! DSRPLUS BLISS Library
: 64      0192 1 %ELSE
: 65      0193 1 LIBRARY 'REQ:DSRLIB';           ! DSR BLISS Library
: 66      0194 1 %FI
: 67      0195 1
: 68      0196 1
: 69      0197 1 : EXTERNAL REFERENCES:
: 70      0198 1
: 71      0199 1 EXTERNAL
: 72      0200 1     FSO1 : FIXED STRING,
: 73      0201 1     IRA  : FIXED STRING,
: 74      0202 1     LSTCHR : REF VECTOR,           !.LIST ELEMENT character.
: 75      0203 1     LSTCNT : REF COUNTED_LIST,     !.LIST ELEMENT counters and controls.
: 76      0204 1     LSTLCH : REF VECTOR,           !The character to the left of the counter.
: 77      0205 1     LSTRCH : REF VECTOR,           !The character to the right of the counter.
: 78      0206 1     LSTLDD : REF VECTOR,           !.LIST ELEMENT display codes.
: 79      0207 1     LSTSKP : REF VECTOR,           !Inter-.LIST ELEMENT skipping
: 80      0208 1     NUMPRM : NUMPRM DEFINE,
: 81      0209 1     PDT   : REF PDT DEFINITION,
: 82      0210 1     SCA   : SCA_DEFINITION;
: 83      0211 1
: 84      0212 1 EXTERNAL LITERAL                   !Error messages
: 85      0213 1     RNFMS,
: 86      0214 1     RNFSTD;
: 87      0215 1
: 88      0216 1 EXTERNAL ROUTINE
: 89      0217 1     ENDCHR,
: 90      0218 1     ENDWRD,
: 91      0219 1     ERMA,
: 92      0220 1     GCSKIP,
: 93      0221 1     GETQC,
: 94      0222 1     PACXXX,
: 95      0223 1     SKPSEP,
: 96      0224 1     STKFRM;
: 97      0225 1

```

Module Level Declarations

```

99 0226 1 GLOBAL ROUTINE LIST (HANDLER_CODE) : NOVALUE =
100 0227 1
101 0228 1 !+
102 0229 1 FUNCTIONAL DESCRIPTION:
103 0230 1
104 0231 1 See ABSTRACT, above.
105 0232 1
106 0233 1 FORMAL PARAMETERS:
107 0234 1
108 0235 1 HANDLER_CODE indicates which command is to be processed.
109 0236 1
110 0237 1 IMPLICIT INPUTS:
111 0238 1
112 0239 1 NUMPRM contains a number, as processed by GETNUM.
113 0240 1
114 0241 1 IMPLICIT OUTPUTS: None
115 0242 1
116 0243 1 ROUTINE VALUE:
117 0244 1 COMPLETION CODES: None
118 0245 1
119 0246 1 SIDE EFFECTS: None
120 0247 1 --
121 0248 1
122 0249 2 BEGIN
123 0250 2 LOCAL
124 0251 2 BULLET,
125 0252 2 LIST_SKIP;
126 0253 2
127 0254 2 SELECTONE .HANDLER_CODE OF
128 0255 2 SET
129 0256 2
130 0257 2 [H_LIST] :
131 0258 3 BEGIN
132 0259 3 LOCAL
133 0260 3 GETQC RESULT,
134 0261 3 GOT_SEPARATOR;
135 0262 3
136 0263 3 !If the user specified a bullet as the second parameter, pick it up.
137 0264 3 GOT_SEPARATOR=SKPSEP (IRA); !Skip spaces/tabs, and a comma
138 0265 3
139 0266 3 !Attempt to pick up a bullet.
140 0267 3 GETQC_RESULT = GETQC ();
141 0268 3
142 0269 3 !See if user did specify a quoted string.
143 0270 4 IF (.GETQC_RESULT GEQ 0)
144 0271 4 !User did give a quoted string.
145 0272 3 THEN
146 0273 3 BULLET = .GETQC_RESULT
147 0274 3 !User specified no bullet.
148 0275 3 ELSE
149 0276 4 BEGIN
150 0277 4
151 0278 4 IF .GOT_SEPARATOR
152 0279 4 !But user said to expect something!!!
153 0280 4 THEN
154 0281 4 ERMA (RNFMS,TRUE);
155 0282 4

```



```

: 213 0340 3
: 214 0341 3
: 215 0342 3
: 216 0343 3
: 217 0344 3
: 218 0345 3
: 219 0346 3
: 220 0347 3
: 221 0348 3
: 222 0349 3
: 223 0350 3
: 224 0351 3
: 225 0352 3
: 226 0353 3
: 227 0354 3
: 228 0355 3
: 229 0356 3
: 230 0357 3
: 231 0358 4
: 232 0359 4
: 233 0360 4
: 234 0361 4
: 235 0362 4
: 236 0363 4
: 237 0364 4
: 238 0365 4
: 239 0366 5
: 240 0367 5
: 241 0368 5
: 242 0369 4
: 243 0370 4
: 244 0371 4
: 245 0372 4
: 246 0373 4
: 247 0374 4
: 248 0375 4
: 249 0376 4
: 250 0377 5
: 251 0378 5
: 252 0379 5
: 253 0380 4
: 254 0381 4
: 255 0382 4
: 256 0383 4
: 257 0384 4
: 258 0385 4
: 259 0386 4
: 260 0387 3
: 261 0388 3
: 262 0389 3
: 263 0390 3
: 264 0391 3
: 265 0392 3
: 266 0393 3
: 267 0394 3
: 268 0395 3
: 269 0396 3

!Update the running counter for this .LIST ELEMENT level
LSTCNT [.LSTCNT [CL_INDEX]] = .LSTCNT [.LSTCNT [CL_INDEX]] + 1;

!Pick up proper inter-list-element spacing.
IF .LSTCNT [CL_INDEX] NEQ 1
!User said .LIST before .LIST ELEMENT, so pick up his
!specified spacing.
THEN
LIST_SKIP = .LSTSKP [.LSTCNT [CL_INDEX] - 1 ]
!User is starting a "level zero" list. Use current
!paragraph skip instead.
ELSE
LIST_SKIP = .PDT_SKIP;

IF .LSTCHR [.LSTCNT [CL_INDEX] - 1] EQL 0
!The list elements are to be numbered.
THEN
BEGIN
PTR = CH$PTR (DIGITS);
DIGIT_COUNT = 0;

!If user specified a left-character output it right now.
IF .LSTLCH [.LSTCNT [CL_INDEX] - 1 ] NEQ 0
THEN
!User did specify a left-character, so output it.
B$WCHAR (.LSTLCH [.LSTCNT [CL_INDEX] - 1], PTR);
DIGIT_COUNT = 1
END;

!Convert the counter to characters.
DIGIT_COUNT = .DIGIT_COUNT + PACXXX (.LSTCNT [.LSTCNT [CL_INDEX]], PTR, .LSTLDD [.LSTCNT [CL_INDEX]]);

!Add on the right-character (usually a '.') if there is one
IF .LSTRCH [.LSTCNT [CL_INDEX] - 1 ] NEQ 0
THEN
BEGIN
CH$WCHAR (.LSTRCH [.LSTCNT [CL_INDEX] - 1], .PTR);
DIGIT_COUNT = .DIGIT_COUNT + 1;
END;

!Now figure out how much to "back-dent", so that the number and
!the '.' appear just to the left of the left margin.
NEEDED_COLUMNS = .DIGIT_COUNT + 2; !2 for '##'
END

!The user has specified that a bullet is to be output.
ELSE
NEEDED_COLUMNS = 3; !3 for character + '##'.

IF .NEEDED_COLUMNS GTR .SCA_LM
THEN
SCA_INDENT = -.SCA_LM
ELSE
SCA_INDENT = -.NEEDED_COLUMNS;

!Now signal a paragraph is to be started.
```



```

: 270      0397 3      SCA_PARA_PND = TRUE;
: 271      0398 3
: 272      0399 3      !Text is output just as if it were coming from the 'IRA'
: 273      0400 4      BEGIN
: 274      0401 4      LOCAL
: 275      0402 4      HOLD_PDT_SKIP;
: 276      0403 4
: 277      0404 4      !Substitute list element spacing for paragraph skip.
: 278      0405 4      HOLD_PDT_SKIP = .PDT_SKIP;           !Preserve current status.
: 279      0406 4      PDT_SKIP = .LIST_SKIP;           !Insert list element spacing.
: 280      0407 4
: 281      0408 4      !Output the list element marker; either a number or a bullet.
: 282      0409 4      IF .LSTCHR [.LSTCNT [CL_INDEX] - 1] EQL 0
: 283      0410 4      !Output a numbered list element
: 284      0411 4      THEN
: 285      0412 5      BEGIN
: 286      0413 5      PTR = CH$PTR (DIGITS);
: 287      0414 5      INCR I FROM 1 TO .DIGIT_COUNT DO
: 288      0415 5      ENDCHR (CH$RCHAR_A (PTR))
: 289      0416 5      END
: 290      0417 5      !Output a list element marked with a bullet.
: 291      0418 4      ELSE
: 292      0419 4      ENDCHR (.LSTCHR [.LSTCNT [CL_INDEX] - 1]);
: 293      0420 4
: 294      0421 4      !Output two spaces which are to precede the start of the text.
: 295      0422 4      ENDCHR (' ');
: 296      0423 4      ENDCHR (' ');
: 297      0424 4      ENDWRD (FALSE, FALSE, FALSE);
: 298      0425 4
: 299      0426 4      !Restore paragraph skipping.
: 300      0427 4      PDT_SKIP = .HOLD_PDT_SKIP;
: 301      0428 3      END;
: 302      0429 3
: 303      0430 2      END;
: 304      0431 2      TES;
: 305      0432 2
: 306      0433 1      END;

```

!End of LIST

```

.TITLE LIST
.IDENT \V04-000\

.EXTRN FS01, IRA, LSTCHR
.EXTRN LSTCNT, LSTLCH, LSTRCH
.EXTRN LSTLDD, LSTSKP, NUMPRM
.EXTRN PDT, SCA, RNFMQS
.EXTRN RN$STD, ENDCHR, ENDWRD
.EXTRN ERMA, GCSKIP, GETQC
.EXTRN PACXXX, SKPSEP, STKFRM

```

```

.PSECT $CODE$,NOWRT,2

```

```

OFFC 00000
5B 00000000G EF 9E 00002
5A 00000000G EF 9E 00009
59 00000000G EF 9E 00010
58 00000000G EF 9E 00017

```

```

.ENTRY LIST, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11 : 0226
MOVAB ERMA, R11
MOVAB LSTCHR, R10
MOVAB LSTSKP, R9
MOVAB ENDCHR, R8

```

Module Level Declarations

H 16
16-Sep-1984 00:48:07
14-Sep-1984 13:06:52

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

	57	00000000G	EF	9E	0001E	MOVAB	PDT, R7		
	56	00000000G	EF	9E	00025	MOVAB	LSTCNT, R6		
	55	00000000G	EF	9E	0002C	MOVAB	SCA+116, R5		
	5E		18	C2	00033	SUBL2	#24, SP		
	52	04	AC	D0	00036	MOVL	HANDLER CODE, R2	0254	
	8F		52	D1	0003A	CMPL	R2, #110	0257	
			03	13	00041	BEQL	1\$		
			00D0	31	00043	BRW	12\$		
		00000000G	EF	9F	00046	1\$:	PUSHAB	IRA	0264
	00000000G	EF	01	FB	0004C	CALLS	#1, SKPSEP		
	53		50	D0	00053	MOVL	R0, GOT_SEPARATOR		
	00000000G	EF	00	FB	00056	CALLS	#0, GETQC	0267	
			50	D5	0005D	TSTL	GETQC_RESULT	0270	
			05	19	0005F	BLSS	2\$		
	54		50	D0	00061	MOVL	GETQC_RESULT, BULLET	0273	
			10	11	00064	BRB	4\$		
	0B		53	E9	00066	2\$:	BLBC	GOT_SEPARATOR, 3\$	0278
			01	DD	00069	PUSHL	#1	0281	
		00000000G	8F	DD	0006B	PUSHL	#RNFMS		
	6B		02	FB	00071	CALLS	#2, ERMA		
			54	D4	00074	3\$:	CLRL	BULLET	0283
			52	DD	00076	4\$:	PUSHL	R2	0286
	00000000G	EF	01	FB	00078	CALLS	#1, STKFRM		
		01	50	E8	0007F	BLBS	R0, 5\$		
			04	00082	RET				
			66	D0	00083	5\$:	MOVL	LSTCNT, R0	0290
	04	A0	60	D1	00086	CMPL	(R0), 4(R0)		
			0C	12	0008A	BNEQ	6\$		
			7E	D4	0008C	CLRL	-(SP)	0293	
		00000000G	8F	DD	0008E	PUSHL	#RNFSTD		
	6B		02	FB	00094	CALLS	#2, ERMA		
			04	00097	RET			0292	
			66	D0	00098	6\$:	MOVL	LSTCNT, R0	0301
		04	A0	9E	0009B	MOVAB	4(R0), R2		
		00000000G	EF	D5	0009F	TSTL	NUMPRM+12	0298	
			0E	13	000A5	BEQL	7\$		
			62	D0	000A7	MOVL	(R2), R0	0301	
	00 B940	00000000G	EF	D0	000AA	MOVL	NUMPRM+4, @LSTSKP[R0]		
			0C	11	000B3	BRB	8\$		
			62	D0	000B5	7\$:	MOVL	(R2), R0	0304
			67	D0	000B8	MOVL	PDT, R1		
	00 B940	04	A1	D0	000BB	MOVL	4(R1), @LSTSKP[R0]		
			65	D0	000C1	8\$:	MOVL	SCA+116, R0	0311
			60	D5	000C4	TSTL	(R0)		
			05	12	000C6	BNEQ	9\$		
			09	D0	000C8	MOVL	#9, TEMP	0313	
			03	11	000CB	BRB	10\$		
			04	D0	000CD	9\$:	MOVL	#4, TEMP	0315
53			51	C1	000D0	10\$:	ADDL3	TEMP, (R0), R3	0317
	04	B5	53	D1	000D4	CMPL	R3, @SCA+120		
			03	18	000D8	BGEQ	11\$		
			51	C0	000DA	ADDL2	TEMP, (R0)	0319	
			62	D6	000DD	11\$:	INCL	(R2)	0324
			62	D0	000DF	MOVL	(R2), R0	0325	
		00 B640	DE	000E2	MOVAL	@LSTCNT[R0], R1			
		04 A1	D4	000E7	CLRL	4(R1)			
		00 BA40	DE	000EA	MOVAL	@LSTCHR[R0], R1		0326	

Module Level Declarations

FC	A1	54	D0	000EF	MOVL	BULLET, -4(R1)	
	51	00000000GFF40	DE	000F3	MOVAL	@LSTLCH[R0], R1	0327
		FC A1	D4	000FB	CLRL	-4(R1)	
	51	00000000GFF40	DE	000FE	MOVAL	@LSTRCH[R0], R1	0328
FC	A1	2E	D0	00106	MOVL	#46, -4(R1)	
	50	00000000GFF40	DE	0010A	MOVAL	@LSTLDD[R0], R0	0329
		FC A0	D4	00112	CLRL	-4(R0)	
			04	00115	RET		0254
0000006F	8F	52	D1	00116	12\$:	C MPL R2, #111	0332
		01	13	0011D		BEQL 13\$	
			04	0011F		RET	
	51	66	D0	00120	13\$:	MOVL LSTCNT, R1	0342
	50	04 A1	D0	00123		MOVL 4(R1), R0	
		04 A140	D6	00127		INCL 4(R1)[R0]	
	01	50	D1	0012B		C MPL R0, #1	0345
		0B	13	0012E		BEQL 14\$	
	52	00 B940	DE	00130		MOVAL @LSTSKP[R0], R2	0349
	53	FC A2	D0	00135		MOVL -4(R2), LIST_SKIP	
		07	11	00139		BRB 15\$	
	52	67	D0	0013B	14\$:	MOVL PDT, R2	0353
	53	04 A2	D0	0013E		MOVL 4(R2), LIST_SKIP	
	52	00 BA40	DE	00142	15\$:	MOVAL @LSTCHR[R0], R2	0355
		FC A2	D5	00147		TSTL -4(R2)	
		5A	12	0014A		BNEQ 18\$	
	6E	04 AE	9E	0014C		MOVAB DIGITS, PTR	0359
		52	D4	00150		CLRL DIGIT_COUNT	0360
	54	00000000GFF40	DE	00152		MOVAL @LSTLCH[R0], R4	0363
		FC A4	D5	0015A		TSTL -4(R4)	
		0A	13	0015D		BEQL 16\$	
00	BE	FC A4	90	0015F		MOVB -4(R4), @PTR	0367
		6E	D6	00164		INCL PTR	
	52	01	D0	00166		MOVL #1, DIGIT_COUNT	0368
	54	00000000GFF40	DE	00169	16\$:	MOVAL @LSTLDD[R0], R4	0372
		FC A4	DD	00171		PUSHL -4(R4)	
		04 AE	9F	00174		PUSHAB PTR	
		04 A140	DD	00177		PUSHL 4(R1)[R0]	
00000000G	EF	03	FB	0017B		CALLS #3, PACXXX	
	52	50	C0	00182		ADDL2 R0, DIGIT_COUNT	
	50	66	D0	00185		MOVL LSTCNT, R0	0375
	50	04 A0	D0	00188		MOVL 4(R0), R0	
	50	00000000GFF40	DE	0018C		MOVAL @LSTRCH[R0], R0	
		FC A0	D5	00194		TSTL -4(R0)	
		07	13	00197		BEQL 17\$	
00	BE	FC A0	90	00199		MOVB -4(R0), @PTR	0378
		52	D6	0019E		INCL DIGIT_COUNT	0379
	50	02 A2	9E	001A0	17\$:	MOVAB 2(R2), NEEDED_COLUMNS	0384
		03	11	001A4		BRB 19\$	0355
	50	03	D0	001A6	18\$:	MOVL #3, NEEDED_COLUMNS	0388
00	B5	50	D1	001A9	19\$:	C MPL NEEDED_COLUMNS, @SCA+116	0390
		07	15	001AD		BLEQ 20\$	
68	A5	00 B5	CE	001AF		MNEGL @SCA+116, SCA+220	0392
		04	11	001B4		BRB 21\$	
68	A5	50	CE	001B6	20\$:	MNEGL NEEDED_COLUMNS, SCA+220	0394
6C	A5	01	D0	001BA	21\$:	MOVL #1, SCA+224	0397
	50	67	D0	001BE		MOVL PDT, R0	0405
	54	04 A0	D0	001C1		MOVL 4(R0), HOLD_PDT_SKIP	
04	A0	53	D0	001C5		MOVL LIST_SKIP, 4(R0)	0406

	50		66	D0	001C9	MOVL	LSTCNT, R0		: 0409
	50	04	A0	D0	001CC	MOVL	4(R0), R0		:
	50	00	BA40	DE	001D0	MOVAL	@LSTCHR[R0], R0		:
		FC	A0	D5	001D5	TSTL	-4(R0)		:
			18	12	001D8	BNEQ	24\$:
	6E	04	AE	9E	001DA	MOVAB	DIGITS, PTR		: 0413
			53	D4	001DE	CLRL	1		: 0414
			0A	11	001E0	BRB	23\$:
	7E	00	BE	9A	001E2	MOVZBL	@PTR, -(SP)		: 0415
		04	AE	D6	001E6	INCL	PTR		:
	68		01	FB	001E9	CALLS	#1, ENDCHR		:
F2	53		52	F3	001EC	AOBLEQ	DIGIT_COUNT, 1, 22\$:
			06	11	001F0	BRB	25\$: 0414
		FC	A0	DD	001F2	PUSHL	-4(R0)		: 0419
	68		01	FB	001F5	CALLS	#1, ENDCHR		:
			20	DD	001F8	PUSHL	#32		: 0422
	68		01	FB	001FA	CALLS	#1, ENDCHR		:
			20	DD	001FD	PUSHL	#32		: 0423
	68		01	FB	001FF	CALLS	#1, ENDCHR		:
			7E	7C	00202	CLRQ	-(SP)		: 0424
			7E	D4	00204	CLRL	-(SP)		:
00000000G	EF		03	FB	00206	CALLS	#3, ENDWRD		:
	50		67	D0	0020D	MOVL	PDT, R0		:
04	A0		54	D0	00210	MOVL	HOLD_PDT_SKIP, 4(R0)		: 0427
			04	00214		RET			: 0433

: Routine Size: 533 bytes, Routine Base: \$CODE\$ + 0000

: 307 0434 1
: 308 0435 1 END !End of module
: 309 0436 0 ELUDOM

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	533	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	00:00.1
_\$255\$DUA28:[RUNOFF.SRC]DSRLIB.L32:1	1248	21 1	86	00:00.2

LIST
V04-000

Module Level Declarations

K 16
16-Sep-1984 00:48:07
14-Sep-1984 13:06:52

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

COMMAND QUALIFIERS

```
;
;      BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:LIST/OBJ=OBJ$:LIST MSRC$:LIST/UPDATE=(ENH$:LIST)
; Size:          533 code + 0 data bytes
; Run Time:      00:10.1
; Elapsed Time:  00:28.0
; Lines/CPU Min: 2584
; Lexemes/CPU-Min: 15527
; Memory Used:  145 pages
; Compilation Complete
```

